Dialogic

Dialogic® Converged Services Platform Release 8.4.1 Engineering Release 3

Developer's Guide: Internet Protocol

February 2008

Copyright and Legal Disclaimer

All contents of this document are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, due to ongoing product improvements and revisions, Dialogic Corporation and its subsidiaries do not warrant the accuracy of this information and cannot accept responsibility for errors or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS EXPLICITLY SET FORTH BELOW OR AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic Corporation or its subsidiaries may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic Corporation or its subsidiaries do not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic Corporation or its subsidiaries. More detailed information about such intellectual property is available from Dialogic Corporation's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. The software referred to in this document is provided under a Software License Agreement. Refer to the Software License Agreement for complete details governing the use of the software.

Dialogic Corporation encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

Dialogic, Dialogic Pro, Brooktrout, Cantata, SnowShore, Eicon, Eicon Networks, Eiconcard, Diva, SIPcontrol, Diva ISDN, TruFax, Realblocs, Realcomm 100, NetAccess, Instant ISDN, TRXStream, Exnet, Exnet Connect, EXS, ExchangePlus VSE, Switchkit, N20, Powering The Service-Ready

Dialogic Product Line Warranty

Unless otherwise stated in an applicable product purchase agreement between the Customer and Dialogic, Dialogic warrants that during the Warranty Period, products will operate in substantial conformance with Dialogic's standard published documentation accompanying the product. If a product does not operate in accordance therewith during the Warranty Period, the Customer must promptly notify Dialogic. Dialogic, at its option, will either repair or replace the product without charge. The Customer has the right, as their exclusive remedy, to return the product for a refund of purchase price or license fee if Dialogic is unable to repair or replace it.

Warranty Period

In the event that you have no signed agreement setting out a warranty period, the Warranty Period shall be the standard warranty period set out on www.dialogic.com on the date of your purchase of the product.

The Warranty Period begins on the date of shipment of any products or software by Dialogic.

The Warranty Period for repaired, replaced or corrected products and software shall be coterminous to the Warranty Provided for the original products or software purchased.

To report warranty claims, Customer may contact Dialogic via email at techsupport@cantata.com or call (781) 433-9600.

Warranty Provisions

A. During the Warranty Period, Dialogic warrants to Customer only that:

(i) Products manufactured by Dialogic (including those manufactured for Dialogic by an original equipment manufacturer) will be free from defects in material and workmanship and will substantially conform to specifications for such products;

(ii) software developed by Dialogic will be free from defects which materially affect performance in accordance with the specifications for such software. With respect to products or software or partial assembly of products furnished by Dialogic but not manufactured by Dialogic, Dialogic hereby assigns to Customer, to the extent permitted, the warranties given to Dialogic by its vendors of such items.

B. If, under normal and proper use, a defect or non conformity appears in warranted products or software during the applicable Warranty Period and Customer promptly notifies Dialogic in writing during the applicable warranty period of such defect or non conformance, and follows Dialogic's instructions regarding return of such defective or non conforming Product or Software, then Dialogic will, at no charge to Customer, either:

(i) repair, replace or correct the same at its manufacturing or repair facility or

(ii) if Dialogic determines that it is unable or impractical to repair, replace or correct the product or software, provide a refund or credit not to exceed the original purchase price or license fee.

C. No product or software will be accepted for repair or replacement without the written authorization of and in accordance with instructions from Dialogic. Removal and reinstallation expenses as well as transportation expenses associated with returning such product or software to Dialogic shall be borne by Customer. Dialogic shall pay the costs of transportation of the repaired or replaced product or software to the destination designated in the original Order. If Dialogic determines that any returned product or software is not defective, Customer shall pay Dialogic's costs of handling, inspecting, testing and transportation. In repairing or replacing any product, part of product, or software medium under this warranty, Dialogic may use new, remanufactured, reconditioned, refurbished or functionally equivalent products, parts or software media. Replaced products or parts shall become Dialogic's property.

D. Dialogic makes no warranty with respect to defective conditions or non conformities resulting from any of the following: Customer's modifications, misuse, neglect, accident or abuse; improper wiring, repairing, splicing, alteration, installation, storage or maintenance performed in a manner not in accordance with Dialogic's or its vendor's specifications, or operating instructions; failure of Customer to apply Dialogic's previously applicable modifications or corrections; or items not manufactured by Dialogic or purchased by Dialogic pursuant to its procurement specifications. Dialogic makes no warranty with respect to products which have had their serial numbers removed or altered; with respect to expendable items, including, without limitation, fuses, light bulbs, motor brushes and the like; or with respect to defects related to Customer's data base errors. Improper packaging of product for repair will not be covered under this warranty agreement. No warranty is made that software will run uninterrupted or error free.

E. Warranty does not include:

a) Dialogic's assistance in diagnostic efforts;

b) access to Dialogic's Technical Support web sites, databases or tools;

c) product integration testing;

d) on-site assistance; or

e) product documentation updates.

These services are available either during or after warranty at Dialogic's published prices.

F. THE FOREGOING WARRANTIES ARE EXCLUSIVE & ARE GRANTED IN LIEU OF ALL OTHER EXPRESS & IMPLIED WARRANTIES (WHETHER WRITTEN, ORAL, STATUTORY OR OTHERWISE), INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. CUSTOMER'S SOLE AND EXCLUSIVE REMEDY AND DIALOGIC'S SOLE OBLIGATION HEREUNDER, SHALL BE TO REPAIR, REPLACE, CREDIT OR REFUND AS SET FORTH ABOVE.

G. IN NO EVENT SHALL DIALOGIC, ITS DIRECTORS, OFFICERS, EMPLOYEES, AGENTS OR AFFILIATES, BE LIABLE FOR ANY COSTS OR DAMAGES ARISING DIRECTLY OR INDIRECTLY FROM YOUR USE OF ANY PRODUCT INCLUDING ANY INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, MULTIPLE, PUNITIVE OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, WHETHER BASED ON CONTRACT, TORT (INCLUDING NEGLEGENCE), STRICT LIABILITY OR OTHER LEGAL THEORY, EVEN IF DIALOGIC, OR ANY OF ITS DIRECTORS, OFFICERS, EMPLOYEES, AGENTS OR AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY EVENT, DIALOGIC'S CUMULATIVE LIABILITY TO YOU FOR ANY AND ALL CLAIMS RELATING TO THE USE OF ANY PRODUCT SHALL NOT EXCEED THE TOTAL AMOUNT OF THE PURCHASE PRICE OR LICENSE FEES PAID TO DIALOGIC FOR SUCH PRODUCT.

H. CUSTOMER AND DIALOGIC HEREBY WAIVE THEIR RIGHT TO TRIAL BY JURY TO THE FULLEST EXTENT PERMITTED BY LAW IN CONNECTION WITH ALL CLAIMS ARISING OUT OF OR RELATED TO THIS WARRANTY, THE PRODUCTS COVERED HEREBY OR THE PERFORMANCE OF ANY PARTY HEREUNDER.

I. THIS WARRANTY SHALL BE CONSTRUED UNDER AND GOVERNED BY THE LAWS OF THE COMMONWEALTH OF MASSACHUSETTS WITHOUT GIVING EFFECT TO ANY CHOICE OR CONFLICT OF LAW PROVISION OR RULE (WHETHER OF THE COMMONWEALTH OF MASSACHUSETTS OR ANY OTHER JURISDICTION) THAT WOULD CAUSE THE APPLICATION OF THE LAWS OF ANY JURISDICTION OTHER THAN THE COMMONWEALTH OF MASSACHUSETTS. CUSTOMER SPECIFICALLY AND IRREVOCABLY CONSENTS TO THE PERSONAL AND SUBJECT MATTER JURISDICTION AND VENUE OF THE FEDERAL AND STATE COURTS OF THE COMMONWEALTH OF MASSACHUSETTS AND SUCH COURTS SHALL HAVE EXCLUSIVE JURISDICTION WITH RESPECT TO ALL MATTERS CONCERNING THIS WARRANTY OR THE ENFORCEMENT OF ANY OF THE FOREGOING.

J. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION.

About this Publication

Purpose

This documentation provides guidelines for using the Dialogic® CSP.

Safety Labels

The following Safety labels may appear in this information product to alert customers to avoidable hazards. The following are in the order of priority:

DANGER

Danger indicates the presence of a hazard that will cause death or severe personal injury if the hazard is not avoided.

WARNING

Warning indicates the presence of a hazard that can cause death or severe personal injury if the hazard is not avoided.

CAUTION

Caution indicates the presence of a hazard that will or can cause minor personal injury or property damage if the hazard is not avoided. Caution can also indicate the possibility of data loss, loss of service, or that an application will fail.

Conventions used

This information product uses the text conventions explained below. In addition, hexadecimal numbers are preceded by a zero and small "x." For example, the decimal number 15 is represented in hexadecimal as 0x0F.

Convention	Description
. . .	A horizontal ellipsis in an API message indicates fields of variable length.
:	A vertical ellipsis in an API message indicates that a block of information is repeated or is variable.
n	The letter *n* is a generic placeholder for a number.
`Sans serif mono space`	Indicates a command name, option, input, output, non-GUI error, and system messages.
`Sans serif monospace italic`	Indicates a parameter name in an input message. Example: `move *.dot a: c:` *`-s`* The *`-s`* is the parameter.
Serif italic	Indicates the name of a book, chapter, path, file, or API message. Example: *`UserDirectory/Config.exe`*
Boldface	Indicates keyboard keys, key combinations, and command buttons Example: **Ctrl+Alt+Del**
Sans serif boldface	Identifies text that is part of a graphical user interface (GUI). Example: Go to the **Configuration** menu and select **Card**->**Span Configuration**

Contents

Copyright and Legal Disclaimer i-2
Dialogic Product Line Warranty i-4

1 Introduction to Converged Services Platform

The Converged Network Framework 1-2
VDAC-ONE and IP Network Interface Series 2 Cards 1-3
Call Agent 1-4

2 VDAC-ONE Card

VDAC-ONE Card Overview 2-2
Basic Configuration 2-5
IP Resources Attributes 2-11
General IP Attributes 2-14
Voice Attributes 2-16
Jitter Buffer Delay/Adaptation Rate 2-23
Fax Attributes 2-27
IP Connection Management 2-30
Dynamic Connection Management 2-33
Release Mode Configuration 2-37
Loopback for Testing 2-39
Routing Examples 2-40
Sample Route Control / Resource Attribute Configure 2-43
VDAC VoIP - 0x009C 2-44

3 IP Network Interface Series 2 Card

Overview 3-3
Similarities Between IP Network Interface Series 2 and VDAC-ONE Cards 3-15
Differences Between IP Network Interface Series 2 and VDAC-ONE Cards 3-16
Summary of API Messages 3-19

Host/IP Network Interface Series 2 Card Integration Information 3-21
Startup Sequence/Basic Configuration 3-22
TDM and Packet Resource Information 3-38
VoIP Services 3-41
VoIP Resource Profiles 3-44
VoIP Resource Profile Terminal Capabilities 3-51
IP Resources Attributes 3-58
General IP Attributes 3-61
Voice Attributes 3-66
Fax Attributes 3-71
IP Connection Management 3-74
Dynamic Connection Management 3-77
Release Mode Configuration 3-81
Loopback for Testing 3-83
Routing Examples 3-84
Sample Route Control / Resource Attribute Configure 3-86
PPL Information 3-87
Gateway Mode 3-90
Querying the Ethernet Link Status, Duplex Type and Speed 3-93
IP Network Series 2 NLP Control Support 3-96

4 IP Network Interface Series 3 Card

Comparison of IPN-2 and IPN-3 Cards 4-4
IP Network Interface Series 3 Voice Coder Packet Rate Information 4-7
Software Requirements 4-8
Obtaining Additional Software Fault Log Information 4-9

5 Session-Initiation Protocol (SIP) Software

SIP Protocol Overview 5-2
CSP SIP Overview 5-6
Configuration Messages 5-8
Configuration 5-9
Possible Modifications to Existing Applications 5-12
PPL Information: SIP UA 0x00A7 5-14
Call Flow Messages 5-25
Routing with Route Control or Outseize Control messages 5-28
Tandem SIP-to-SIP Call Flow Example 5-31
Host Controlled SIP Registration Failure Response Code 5-50
SIP Registration Duration Check 5-52
Dual Ethernet Port 5-54
SIP User Agent Redundancy 5-55

Host Notification of Selective SIP Header Information 5-62
Via Heading Reporting 5-63
Call-ID Reporting for Outbound SIP Calls 5-70
Subject Header Field Access 5-71
SIP Display Name Parameter in From Header 5-83
Access To Contact Header 5-85
SIP Access To Parameters in To Header 5-87
Report and Control of
P-Asserted and P-Access Network Info Headers 5-91
SIP Remote Party ID and RPID Privacy for Outbound Calls 5-94
Remote Party ID 5-99
Call Progress Notification to Host with PPL Event Indication 5-102
Early Media 5-103
SIP 182 Queued Message 5-106
SIP/VoIP Media Parameters Synchronization 5-109
Authentication 5-128
Host Control Method of SIP 180 Provisional Response Generation 5-166
RFC 2833 (DTMF Digits) Dynamic Payload Negotiation 5-168
Session Timers 5-170
Delayed Media 5-171
RE-INVITE Message 5-172
Codec List in SIP-Initiated Offer 5-173
Send and Receive SIP Signals Using the Same Port 5-177
SIP Notifications of Options 5-191
Disabling the SIP Domain Name System (DNS) Server 5-195
REFER and NOTIFY Methods 5-203
SIP Referred By Mechanism 5-204
Host Generated Refer Message 5-225
Refer-To Header Parameter Access 5-231
SIP Subject Header in REFER 5-234
Report REFER Request URI in PPL Event Indication 5-236
Refer to Phone Number 5-246
SIP Population of Status in Outbound NOTIFY Message 5-248
SUBSCRIBE and NOTIFY Method for DTMF Detection 5-250
SIP Notify Subscription State 5-252
Programmable SIP URI Extensions 5-264
Support for Request URI Parameters in SIP INVITE Messages 5-269
Support SIP Max Forward in INVITE Message 5-272
PRACK Support 5-275
Support for SIP INFO Message 5-283
SIP Tunneling 5-286

SIP Support for MIME 5-300
Outbound SIP Call with Call Agent Mode 5-308

6 Call Agent

Overview 6-2
NPDI Data Model 6-6
Virtual IP Channels 6-15
Deploying Call Agent 6-19
Sample Configuration File for Call Agent 6-22
Call Agent Reconnect 6-27
Outbound SIP Call with Call Agent 6-32
Session Description Protocol (SDP) Pass-Through for Call Agent Mode 6-38
Support REFER Request in Call Agent 6-45
PPL Event Request for RE-INVITE Message 6-55
Call Flows 6-92

7 Interworking

Introduction to Interworking 7-2
Configuring Interworking 7-3
SIP-UPDF Tunneling 7-6
Tunneling UPDF TLVs 7-7
UPDF Example 7-12

8 Routing VoIP Calls

Channel Management PPL Component for VDAC - 0x0061 8-2
Routing SIP and H.323 Calls Using Route Control Message 8-3
Routing SIP and H.323 Calls Using Outseize Control Message 8-14
Routing Clear Channel VoIP Calls 8-18
IP Based Routing 8-20

9 H.323 Software

H.323 Protocol Overview 9-2
Hardware Requirements 9-4
Other Prerequisites 9-5
Software Overview 9-6
Internal Architecture 9-19
Possible Modifications to Existing Applications 9-23
Configuration 9-25
PPL Component for RAS - 0x00A0 9-31
PPL Component for H.225 - 0x00A1 9-38
PPL Component for L3P H.245 - 0x00A2 9-44

Call Flows 9-48

A H.323 Support and Compliance

B SIP Support and Compliance

C H.323 Call Flows and Message Traces

Call Flows C-2

Call Release Call Flows C-95

1 Introduction to Converged Services Platform

Purpose This chapter provides an introduction to the Dialogic Converged Services Platform (CSP).

The Converged Network Framework

Overview

The CSP enables developers to move existing and new applications into the next generation converged network framework. It provides a feature-rich, multi-protocol, multi-functional platform with proven Internet Protocol (IP) and Public Switched Telephone Network (PSTN) capabilities.

Two Types of Protocols

The CSP supports two types of signaling protocols:

- *Session*: Includes Session-Initiation Protocol (SIP) User Agent and H.323
- *Transport*: Includes Real-Time Protocol/Real-Time Control Protocol (RTP/RTCP)

SIP

SIP allows the CSP to act as an IP Service Node, providing application services and media resources to a softswitch or SIP proxy server. A softswitch or SIP proxy server uses SIP to hand off a call requiring call treatment with partner-developed applications resident on the host. The SIP software is embedded in the CSP Matrix Series 3 Card and interacts with host applications the same way that other Layer 3 circuit-based protocols do, such as SS7 ISUP and ISDN.

H.323

The H.323 protocol is used for multimedia conferencing over packet networks. H.323 allows multimedia communication devices to interoperate regardless of the type of connection devices on the network.

The H.323 is embedded in the IP Signaling Series 3 card. The H.323 offering on the CSP provides both signaling and media capability, so application developers do not need to add these features separately.

RTP/RTCP

RTP and RTCP streams are transported to the media gateways using the VDAC-ONE card.

VDAC-ONE and IP Network Interface Series 2 Cards

VDAC-ONE Card

The VDAC-ONE (Voice-Data Access Concentrator) card performs two-way conversion between circuit-switched data and packet-switched Ethernet data. This conversion is required by packetized voice applications such as the Voice over Internet Protocol (VoIP). The card also integrates media resources over IP technology.

The VDAC-ONE card converts circuit-switched voice to IP packets, employing compression algorithms that can increase capacity toward the IP network side. You can have parameters modified for an individual call, often while the call is active, changing the quality of service, as needed. Refer to the *Basic Configuration (2-5)* section for more information.

For more information refer to the VDAC-ONE Card Chapter.

IP Network Interface Series 2 Card

IP Network Interface Series 2 card is the second generation VoIP line card used in the CSP. In the broadest sense, IP Network Interface Series 2 card is conceptually a VDAC-ONE card with higher channel densities. A major requirement for IP Network Interface Series 2 card is to provide an easy migration path for VDAC-ONE users with minimal host impact.

For more information refer to the chapter *IP Network Interface Series 2 Card*.

Similarities and Differences

Refer to the following sections for similarities and differences with the VDAC-ONE and IP Network Interface Series 2 cards.

Differences Between IP Network Interface Series 2 and VDAC-ONE Cards

VoIP Attributes

Call Agent

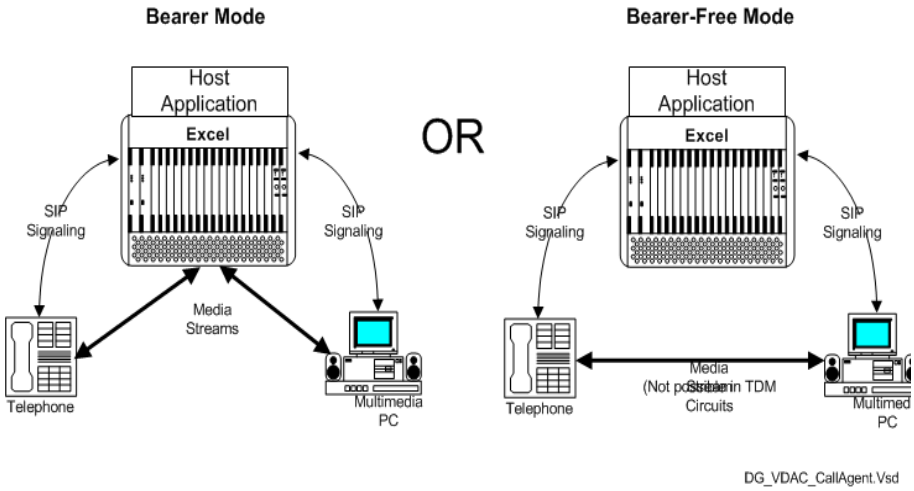
Overview

The Call Agent feature enables the CSP to connect calls with or without bearer paths through the CSP. The CSP can accept requests from SIP clients for a specific host-based voice service and then establish a SIP signaling connection for the requested service.

Call Agent Mode

Call control is resident on the CSP so the RTP stream does not have to pass through the CSP via the VDAC card which avoids “hairpinning” the call. The CSP is now Call Agent mode. You can switch modes and have the RTP stream pass through the VDAC card. The CSP is in VDAC mode.

Figure 1-1 VDAC or Call Agent Mode



Background

Call agent switching technology is driven by the convergence of telecommunication and data networks. Data networks such as IP are typically self-routing and connectionless. Therefore, are more like circuit-switched transport networks.

Refer to the Call Agent chapter for a full explanation of Call Agent in the CSP.

2 VDAC-ONE Card

Purpose This chapter provides information you need to configure the VDAC-ONE (Voice-Data Access Concentrator) card.

VDAC-ONE Card Overview

Two-Way Conversion

The VDAC-ONE (Voice-Data Access Concentrator) card performs two-way conversion between circuit-switched data and packet-switched Ethernet data. This conversion is required by packetized voice applications such as the Voice over Internet Protocol (VoIP). The card also integrates media resources over IP technology.

The VDAC-ONE card converts circuit-switched voice to IP packets, employing compression algorithms that can increase capacity toward the IP network side. You can have parameters modified for an individual call, often while the call is active, changing the quality of service, as needed.

Media Resources

Integrating media resources using IP technology provides many advantages. Typically, media resources are connected by T1, E1, or J1 interfaces that consume one 64 Kbps port per call, limiting the capacity of the system. The VDAC-ONE card integrates media resources over IP using the standards-based Real-Time Protocol (RTP).

Integrating media resources using standards-based technology also allows media resources to be shared between the CSP and other network infrastructure. Packet switching to media resources allows the application to benefit from voice compression, increasing capacity on the application. This flexibility allows the CSP and the applications to scale independently and incrementally, as needed, eliminating excess hardware.

Scalability/Programmability

Each VDAC-ONE card consists of 160 channels divided into five spans each with 32 channels. Each VDAC-ONE card has a main board and four modules 0-3.

Use the *IP Address Configure* message to assign the main board and each module an IP address, subnet mask, and gateway IP address. Assign up to five logical spans on the VDAC-ONE card using the *Assign Logical Span ID* message. Use the *Resource Attribute Configure* message to assign attributes to the VDAC channels or modules.

> **Important!** Span/Channel is bound to an IP Address/Port during call setup. The *Route Control/Outseize Control* messages are used to initiate the call and bind a Span/Channel to the IP Address/Port. This association lasts for the duration of the call.

The VDAC-ONE card has its own resources to provide echo cancellation, silence suppression, and detection of fax tones. Multiple VDAC-ONE cards can populate one node and interoperate with all other CSP cards.

> **Important!** The CSP supports eight VDAC-ONE cards per chassis.

Adding the VDAC-ONE card to existing enhanced-service platforms allows applications to be quickly IP-enabled, providing a significant advantage over other systems that require developers to rewrite the application for packet transport. The ability to support both circuit- and packet-based transport enables developers to create entirely new services, such as text-to-speech and web-based applications.

Redundancy

You can provide reliability for IP switching two ways with the VDAC-ONE card.

- Each VDAC-ONE card has two 100 Base-T Ethernet ports that you can connect to separate Ethernet switches, providing link-level redundancy.
- Using multiple VDAC-ONE cards provides card-level redundancy with load sharing, where sufficient VDAC resources exist to keep the system operational when one card goes out of service. The host must first transfer those resources to the other cards with the *Assign Logical Span* (0x00A8) message. (Note that existing calls on the failed VDAC-ONE card will drop and subsequent calls are directed to available resources on another card.)

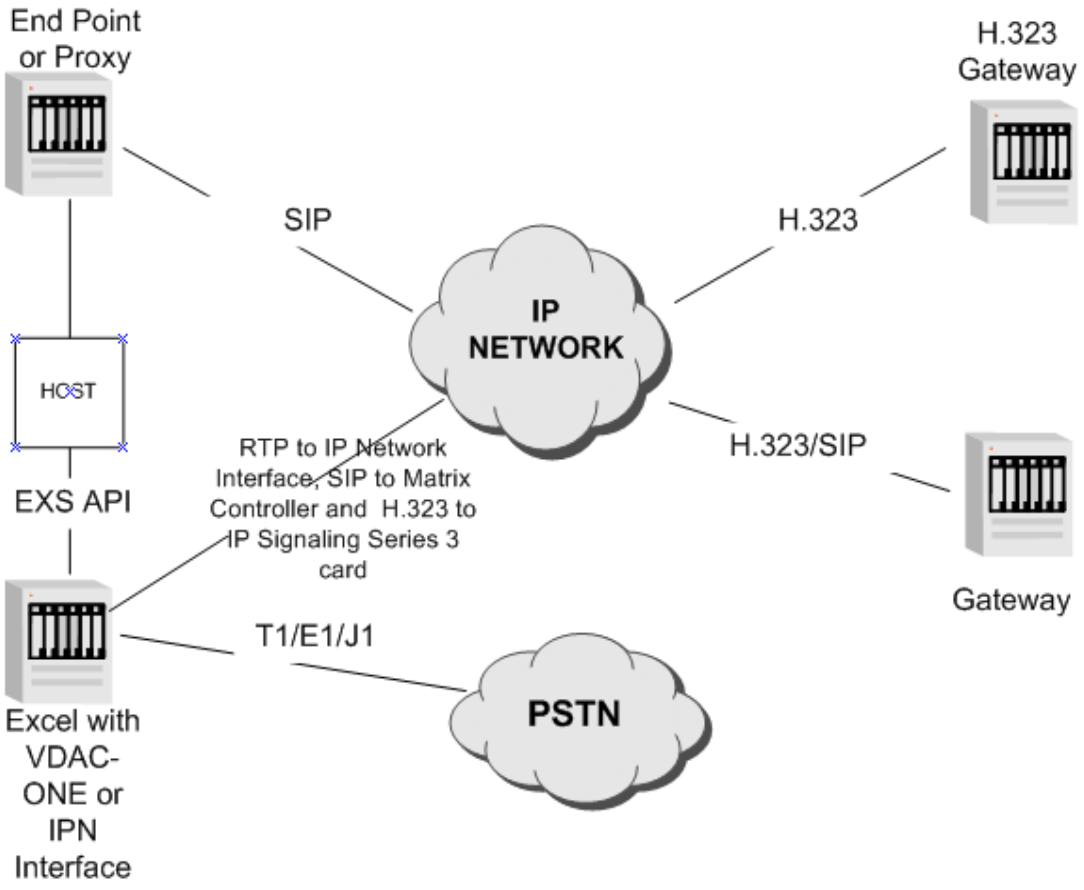
Cost Effective Service Integration

Adding packet switching capabilities to the already flexible CSP provides many additional cost-saving and revenue-generating benefits. The ability to compress voice increases system capacity toward the IP network side, allowing the same amount of bandwidth to transport voice more cost-effectively, and allowing the excess bandwidth to support more customers. Using packet transport for media resources extends the capacity gains of voice channels to the resource, allowing each service to handle greater call volumes.

Integrating applications through IP provides scalability for the applications because they are no longer limited by the number of interface ports available. The standards-based transport technology allows you to easily add third-party applications. This integration

method allows multiple, multi-vendor applications to operate on a single system. Other networks can also share these resources because RTP is a standards-based integration technology.

Figure 2-1 Protocol Interworking through the VDAC-ONE card



Basic Configuration

Overview

The VDAC-ONE card requires basic card configuration of spans and channels similar to the T-ONE card.

Refer to the *API Reference* for additional information on the API messages and TLVs mentioned hereafter.

API Messages

You can use the following API Messages to configure the VDAC-ONE card. Refer to the *API Reference* for details of each:

- *ARP Cache Query* (0x00FC)
- *ARP Cache Report* (0x00FB)
- *IP Address Configure* (0x00E7)
- *IP Address Query* (0x00E6)
- *Resource Attribute Configure* (0x00E3)
- *Resource Attribute Query* (0x00E4)
- *Route Control* (0x00E8)
- *Card Population Query* (0x0007)
- *Assign Span* (0x00A8)
- *Service State Configure* (0x000A)

Configuration

Follow the steps below to configure the VDAC-ONE Card. The API messages involved appear in parenthesis.

1. Configure IP addresses, subnet masks, and gateway IP addresses (*IP Address Configure 0x00E7*).
    - On a fully-populated VDAC-ONE card (with four VoIP modules) you assign five IP Addresses/Subnet Masks (one to the main board and one to each of the VoIP modules). See your IT/network administrator to ensure that you use the correct values for your network configuration.
2. Assign Logical Span IDs (*Assign Span 0x00A8*).
    - Depending on your application and configuration, you can assign five spans of 32 channels each to the four VoIP modules.
3. Configure Call Control parameters with *PPL Configure 0x00D7*.
4. Bring spans in service with *Service State Configure 0x000A*.
5. Bring channels in service with *Service State Configure 0x000A*.

Configuring IP Address and Subnet Mask

The IP address is 32 bits. IP addresses and subnet masks can be assigned using the *IP Address Configure* message.

If the host sends an *IP Address Configure* message with an IP address and subnet mask that differ from the values stored in the Electronic Erasable Programmable Read Only Memory (EEPROM), the new values will overwrite the stored values.

If the host assigns the same IP address and subnet mask, the message is acknowledged and the card does not need to be reset, so the host can send the *IP Address Configure* message whenever it is necessary.

For the new IP address and subnet mask to take effect, you must reboot the card. You can perform this step by including the Engage IP TLV in the *IP Address Configure* message.

The subnet masks default to the values shown in the following table.

Table 2-1 Default Values

Class	IP Address	Subnet Mask
A	1.0.0.0 - 127.0.0.0	0xFF.00.00.00
B	128.0.0.0 - 191.0.0.0	0xFF.FF.00.00
C	192.0.0.0 - 254.0.0.0	0xFF.FF.FF.00

Assigning Dynamic IP Addresses

You can dynamically assign an IP address to a VoIP module without affecting the other modules on the VDAC-ONE card. You configure the IP address on a module by using the *IP Address Configure* message, with the IP Address/Subnet Mask TLV (0x01). You must include one of the following TLVs in the message as well:

- Engage IP TLV (0x02)
- Reset IP TLV (0x03)

Re-Assigning an IP Address

When IP addresses are re-assigned, the TCP/IP stack must be re-initialized. To properly re-initialize the stack, you must reboot the board or module using one of the following TLVs in the *IP Address Configure* message:

- Engage IP TLV (0x02)

- Reset IP TLV (0x03)

Gateway IP Address

You can assign Gateway IP addresses to the VDAC-ONE card, to the VoIP modules or both. When you use the *IP Address Configure* message to assign a Gateway IP address, you must include in the message the corresponding IP address and subnet mask.

Ethernet Link Redundancy

If you configure redundancy for the Ethernet links on the VDAC-ONE card, the standby Ethernet port becomes active automatically if the previously active link fails. Ethernet Link Redundancy is disabled by default.

To configure Ethernet Link Redundancy:

1. Connect the lower Ethernet port on the VDAC-ONE card to an Ethernet switch.
2. Enable Ethernet Link Redundancy on the VDAC-ONE card by using the Ethernet Link Redundancy (0x09) TLV in the *IP Address Configure* message.

To manually activate an Ethernet port, send the *IP Address Configure* message with the Activate Ethernet Port TLV (0x0A).

To query the state of Ethernet Link Redundancy, use the *IP Address Query* message.

Do not enable Ethernet Link Redundancy unless there are IP connections to both the upper and lower Ethernet ports on the VDAC-ONE card.

Flexible RTCP and T.38 Port Selection

You can choose any port number for RTCP and T.38 connections. By using the RTCP or T.38 port TLVs, the host explicitly specifies port values at one of the following three points:

- Before the call is connected (with the *Resource Attribute Configure 0x00E3* message and the IP Address AIB)
- During call setup (with the *Route Control 0x00E8* and *Outseize Control 0x002C* messages)
- During an active call (with the *Resource Attribute Configure 0x00E3* message and the Span/Channel AIB)

This flexible port selection scheme reduces or eliminates interoperability issues with other gateways and IP-enabled switches. It also aligns with the requirements set forth by the ITU-T H.323 Annex D standards, as well as other protocols such as SIP.

Considerations

The VDAC-ONE card does not automatically select the T.38 ports. Instead, the host must select the source and destination T.38 ports. The VDAC-ONE card is fax-aware whenever Fax Relay Mode is enabled. The VDAC-ONE card detects the CED (Called Station ID) tone from the PSTN or T.38 packets from an IP network. The card then alerts the host with a PPL Event of Fax Start. The channel then transitions from voice to fax mode and starts generating only T.38 packets. At this point, the host must select the T.38 port as quickly as possible. Otherwise, the VDAC-ONE card discards any T.38 packets. The fax machines at each endpoint will then give up because they do not receive an appropriate response, and the fax transmission fails.

DTMF over IP Considerations

Refer to *DTMF over IP Considerations (3-34)*.

VDAC-ONE Card over the Exnet® Ring

The VDAC-ONE card over the Exnet® ring is similar to any other resource over the ring. Spans are assigned on the VDAC-ONE card and this span information is shared among all nodes. When the host sends a message to connect to a VDAC-ONE card span/channel on a remote node, the system makes the connection to the remote node.

You must configure the ring for four packets (not three as normal).

> **Important!** Ring Timing is not supported on nodes that contain VDAC-ONE cards. Use Loop Timing from a non-VDAC span or an external clock source.

Address Resolution Protocol (ARP)

The VDAC-ONE card supports the Address Resolution Protocol (ARP) of TCP/IP, which is used to obtain the physical address (Ethernet address) of an endpoint when only the logical address (IP Address) is known.

The CSP broadcasts an *ARP Request* message to the network with the known IP address. The corresponding endpoint responds with its Ethernet address and the connection is made.

The VDAC-ONE card drops all incoming the VDP packets and generates a single ARP request per second for each VDAC-ONE channel until the Destination Hardware Address (MAC address) is received.

ARP Cache Table

Each module on the VDAC-ONE card maintains an ARP Cache Table containing the IP Address and Ethernet address of remote endpoints. These tables allow connections to be made to the remote endpoint without sending the *ARP Request* message each time. An ARP Cache Table can contain a maximum of 40 entries.

You can query the content of a module's ARP Cache Table by using the *ARP Cache Query* message. The ARP Cache Table information is sent to the host in one or more *ARP Cache Report* messages.

ARP Cache Aging

With the ARP Cache Aging feature, entries in the ARP cache are deleted after a preset interval. Each entry in the ARP cache has an associated time-to-live. The ARP Cache Aging timer is set to five minutes. The ARP Cache manager continually increments the time-to-live field, and discards the entry when the value reaches the aging timer value. Entries are removed based solely upon the time-to-live value, and not upon the frequency that the entry was used.

ARP Cache Query

This feature allows the host to use the *ARP Cache Query* message to query the ARP cache of any XoIP module. The CSP responds with one or more *ARP Cache Report* messages that contain all existing ARP cache entries.

ARP Cache Flushing

You can manually remove entries from an ARP Cache Entry Table by using the Flush ARP Cache Entry TLV in the *ARP Cache Query* message.

Internet Control Message Protocol (ICMP) Message

The VDAC-ONE card supports the Internet Control Message Protocol (ICMP), which reports errors related to IP packet processing. If the VDAC-ONE card attempts to connect to a remote endpoint and receives the ICMP message Destination Unreachable, it flushes the entry for that IP address from the ARP Cache Table, and sends an *ARP Request* message to obtain a new Ethernet address for the entry. When the VDAC-ONE card receives the new Ethernet address, the connection is made and the ARP Cache Table is updated. This process occurs without host intervention or notification.

The VDAC-ONE can generate the following ICMP message types:

- Destination Unreachable
- Echo Reply (Ping Reply)

The VDAC-ONE can process the following incoming ICMP messages:

- Destination Unreachable
- Echo Reply (Ping Reply)

IP Resources Attributes

Overview

You can assign attributes to a VoIP module during initial configuration by using the *Resource Attribute Configure* message.

You can assign attributes during call set-up by using the *Route Control* or *Outseize Control* messages with Address Information Blocks (AIBs).

You can dynamically assign some attributes to a channel after call setup by using the *Resource Attribute Configure* message with the span/channel AIB.

Synchronizing VoIP Media Parameters

If you change VoIP resource configuration with the *Resource Attribute Configure* message, the local copy of the configuration maintained by SIP and H.323 software remains unchanged. You synchronize the configurations as follows:

- CSA users select the Resynchronization option.
- Non-CSA users send the Resynchronize VoIP Media Parameters TLV (0x0284) in the *VoIP Protocol Configure* (0x00EE) message. Refer to the *0x0284 Resynchronize VoIP Media Parameter* in the *API Reference* for the details of this TLV.

IP Resource Attributes Summary Table

The following tables show the default values for IP Resource Attributes and indicate if you can configure them dynamically on an active connection.

Table 2-2 General Attributes

TLV Tag and Name	Default Setting	Dynamic Configuration Supported
0x01D0 Gateway Mode	Non Port Consuming	No
0x01D4 Type of Service	See TLV Format	Yes
0x01DF UDP Source Port Validate	Enabled	Yes
0x01EB RTP Timer Timeout	Disabled	No
0x01EC Media Inactivity Detection Timer	Disabled	No

Table 2-3 Fax Attributes

TLV Tag and Name	Default Setting	Dynamic Configuration Supported
0x01C5 Fax Type Enable	Disabled	No
0x01C7 Fax Bypass Coder Type	Disabled	No
0x01D2 Fax Payload Redundancy	Disabled	No
0x01E1 Fax Compatibility Mode	Backward compatible mode	No
0x01E6 Source T.38 Port	No default	Yes
0x01E7 Destination T.38 Port	No default	Yes
0x01E8 Source RTCP Port	One offset from source RTP port	Yes
0x01E9 Destination RTCP Port	One offset from destination RTCP port	Yes

Table 2-4 Voice Attributes

TLV Tag and Name	Default Setting	Dynamic Configuration Supported
0x0100 RTP Payload Type (VDAC-ONE)	G.711µ-Law	Only between G.723 5.3 Kbps and G.723 6.3 Kbps
0x0101 RTP Payload Size (VDAC-ONE)	1X	Yes
0x0102 RTP Silence Suppression	Disabled	Yes
0x0103 RTP Echo Cancellation	Enabled	Yes
0x01C2 Minimum Jitter Buffer Delay	75 ms	No
0x01C3 Maximum Jitter Buffer Delay	150 ms	No
0x01C4 Adaptation Rate	7	No
0x01D1 RTP Payload Redundancy	Disabled	No
0x01DB Connection Mode	See the TLV format.	Yes
0x01E2 RFC 2833 Enable	Disabled	No
0x01E8 Source RTCP Port	One offset from destination RTP port	Yes
0x01E9 Destination RTCP Port	One offset from destination RTP port	Yes

General IP Attributes

Overview

The following are general IP resource attributes that apply to voice and fax connections:

- Type of Service
- UDP Source Port Validation
- RTP Timer Timeout
- Port Consumption

IP Type Of Service

The VDAC-ONE card supports the *IPv4 Type of Service* field in the IP packet header. This field is used to instruct packet switches and routers in an IP-based network how to control packets sent from a VDAC-ONE card. You can set options in this field to indicate that the packets should be given preferential treatment on a Class of Service basis.

> **Important!** Most IP switches and routers do not currently support different levels of service.

You can configure the following IP Type of Service options for outgoing packets:

- Cost
- Reliability
- Throughput
- Delay
- Precedence

Default: 0x00 (No preferential options set)

Use TLV: 0x01D4 Type of Service

UDP Source Port Validation

If the User Datagram Protocol (UDP) Source Port Validation is enabled, the VDAC-ONE card verifies that the UDP source port of the incoming packets matches the destination port of the originating outgoing packets.

Default: Enabled

Use TLV: 0x01DF UDP Source Port Validate

RTP Timer

You can create a timer for monitoring the first incoming network packet (RTP, RTCP, or T.38 packet) on a per channel basis.

When you enable this timer for a particular VDAC-ONE card channel, and if the configured timer value has elapsed and no packet is ever received, then this timer expires. A PPL Event Indication with Event ID 0x07 is generated.

The timer value is in 10 millisecond increments, 0x00 to 0xFFFF. By configuring the timer to a non-zero value, you automatically enable the timer.

Default: Disabled

Use TLV: 0x01EB RTP Timer Timeout (VDAC-ONE)

Media Inactivity Detection Timer

To detect possible media inactivity occurrence, you can create a timer for monitoring the incoming RTCP packets at a per module or channel basis. When this timer is enabled for a particular VDAC channel, if the configured timer value has elapsed and no RTCP packets are received, then this timer will expire and generate a PPL Event Indication with Event ID of 0x08. This alerts the host on a possible media inactivity detection. RTCP must be supported by remote endpoints for this TLV to work.

Default: Disabled

Use TLV: 0x01EC Media Inactivity Detection Timer

CNAME Support for VDAC/ IPN-2

The CNAME field is supported for the IPN-2 and VDAC-ONE cards. The CNAME field will automatically be populated at channel configuration time and included in the RTCP Sender Report. The CNAME field is not configurable.

The data used for the CNAME field is comprised solely of the ASCII character formats for the VDAC/IP module's source IP address (15 bytes) and source UDP (RTP) port number (5 bytes) for a specific RTP voice session, separated by an ASCII "colon" (":") character (1 byte).

You can use an Ethernet trace program to capture and analyze RTCP packets.

Voice Attributes

Introduction

You can configure the following voice attributes with the *Resource Attribute Configure* (0x00E3) message:

- RTP Payload Type
- RTP Payload Size
- Silence Suppression (sometimes referred to as Voice Activity Detection, VAD)
- Echo Cancellation
- Redundancy
- RTCP Port Offset
- Connection Mode
- Jitter Buffer/Adaptation Rate

RTP Payload Type

The RTP payload type indicates the algorithm used for compressing the data or payload portion of the packet. For an active connection, the only configuration change you can make is between G.723 5.3 Kbps and G.723 6.3 Kbps. See the TLV for options.

Default: G.711µ-Law

Use TLV: 0x0100 RTP Payload Type (VDAC-ONE)

RTP Payload Size

RTP payload sizes can be changed in multiples of their individual packetization rates. The default multiplication factor is 1x. The maximum multiplication factor supported is 8x. The higher the multiplication factor, the larger the packets, and the smaller the network throughput.

Example: For G.711, you can compute the payload size by multiplying the packetization rate by the number of bytes of PCM data sampled per second.

160 Bytes = 8000 samples/sec * 20 milliseconds

The RTP payload size factor produces the number of 20 millisecond payloads (30 milliseconds for G.723) to be used for generating one RTP packet. Multiples lower than 20 milliseconds (that is, 5 milliseconds, 10 milliseconds, 15 milliseconds) or 30 milliseconds for G.723 are not supported.

Default: 1x

Use TLV: 0x0101 RTP Payload Size (VDAC-ONE)

Payload Type and Payload Size

The following table describes the relationship between the Payload Type and Payload Size on a VDAC-ONE card.

Table 2-5 VDAC-ONE Voice Coder Packet Rate Information

TLV Payload Type Data Value (decimal)	RTP Payload Type	Basic Packet Rate (ms)	Default Packet Rate (ms)	Max Packet Rate (ms)
0	G.711 A-Law (64Kbps)	20ms	1x - 20ms	8x - 160ms
1	G.711 μ-Law (64Kbps)	20ms	1x - 20ms	8x - 160ms
2	G.726 (16Kbps)	20ms	1x - 20ms	8x - 160ms
3	G.726 (24Kbps)	20ms	1x - 20ms	8x - 160ms
4	G.726 (32Kbps)	20ms	1x - 20ms	8x - 160ms
5	G.726 (40Kbps)	20ms	1x - 20ms	8x - 160ms
6	G.727 (16Kbps)	20ms	1x - 20ms	8x - 160ms
7	G.727 (24, 16Kbps)	20ms	1x - 20ms	8x - 160ms
8	G.727 (24Kbps)	20ms	1x - 20ms	8x - 160ms
9	G.727 (32, 16Kbps)	20ms	1x - 20ms	8x - 160ms
10	G.727 (32, 24Kbps)	20ms	1x - 20ms	8x - 160ms
11	G.727 (32Kbps)	20ms	1x - 20ms	8x - 160ms
12	G.727 (40, 16Kbps)	20ms	1x - 20ms	8x - 160ms
13	G.727 (40, 24Kbps)	20ms	1x - 20ms	8x - 160ms
14	G.727 (40, 32Kbps)	20ms	1x - 20ms	8x - 160ms
15	G.723.1 (5.3Kbps)	30ms	1x - 30ms	8x - 240ms
16	G.723.1 (6.3Kbps)	30ms	1x - 30ms	8x - 240ms
17	G.729 (8Kbps)	20ms	1x - 20ms	8x - 160ms

Silence Suppression

During a normal voice conversation, much of the time is wasted on silence from one or both ends. Ethernet bandwidth can be conserved if, during these periods of silence, RTP packets are sent with silence-encoded, compressed payloads.

You can enable or disable Silence Suppression on an established connection.

Default: Disabled

TLV: 0x0102 RTP Silence Suppression
(Do not send to an active call that is in fax/modem mode.)

The VDAC-ONE card supports silence suppression as follows:

A voice activity detection (VAD) algorithm determines which portions of the input signal contain active speech. At the beginning of silence periods (the end of active periods), a special silence insertion descriptor (SID) packet is generated that describes the background noise. The SID packets are generated at the onset of the silence period, and whenever the characteristics of the background noise change. The speech decoder that receives the SID packet uses a comfort noise generation (CNG) algorithm to reproduce the background noise from the information in the packet and possibly information contained in past active voice packets. The average bit rate required for voice transmission is lowered considerably when using silence suppression.

Echo Cancellation

In compliance with ITU G.168, this feature eliminates echo introduced by impedance mismatched hybrids. You can enable and disable Echo Cancellation on an established connection.

Default: Enabled

TLV: 0x0103 RTP Echo Cancellation

RTP Payload Redundancy

You can set the redundancy level for RTP using the RTP Payload Redundancy TLV. See RFC 2198.

Default: Disabled

TLV: 0x01D1 RTP Payload Redundancy

Connection Mode

During call setup, the Connection Mode is automatically determined according to the presence of destination and source addresses in the *Route Control* or *Outseize Control* message.

Source Address Only - Receive Only

Source and Destination Address - Transmit and Receive

To change the Connection Mode after a call is established using *Resource Attribute Configure 0x00E3*, send the Connection Mode TLV with the appropriate mode set.

In hold mode, no voice is transmitted or received.

> **Example:** After a two-way call has been established to a Call Center, the caller is put on hold and connected to music. The Connection Mode of the Call Center line is changed to Transmit Only. If callers enable the mute feature on their phone, the Connection Mode on their line is changed to Receive Only.

Default

There is no default Connection mode. If you query the VDAC Module Connection Mode, it is reported as 0xFF. This is a generic value and does not indicate a specific mode. You cannot set the Connection Mode to this value.

TLV: 0x01DB Connection Mode

> **Important!** Source Address (IP/RTP Port) is always required in *Route Control* or *Outseize Control* messages.

RFC 2833 Enable

This TLV allows DTMF signals to be relayed within the VoIP media stream, using a special RTP payload format (see RFC 2833). Low bit-rate audio codecs (such as G.729 or G.723.1) can compromise the signal integrity of DTMF digits (and other telephony tones and signals), causing inaccurate detection and recognition of the DTMF digits on the recipient side. When this feature is enabled, the detected incoming DTMF digits (PSTN side) are removed from the audio stream by the VDAC-ONE card. The information for the detected and removed DTMF digits is embedded within the RTP stream, using a special RTP payload format. The VDAC-ONE card on the receiving side decodes this special RTP payload carrying DTMF information, and regenerates the DTMF digits toward the receiving PSTN side. This TLV cannot be set for an active call. It can be configured only before or during the setup of the VDAC-ONE card call.

> **Important!** The CSP does NOT support RFC 2833 via H.323 signaling.

Default: Disabled

TLV: 0x01E2 RFC 2833 Enable

See also the follow section in the *Call Agent* chapter.

- *RFC 2833 Multi-Unicast (6-4)*

RTP Suppression for H.245 Signaling Tones

The EXS API has been updated to allow you to configure how DTMF digits are transmitted to the IP network.

Important! This feature is for Clear Channel calls only.

This feature extends the RFC2833 Enable (0x01E2) TLV as follows:

- The new value 0x02 removes the digit from the voice stream and drops the digit.
- If you are using the VDAC-ONE card, this TLV can be configured only before or during the call establishment. It cannot be set during an active call. If you are using the IP Networking Series 2 card, this TLV can be configure on the fly while the call is active.

Source RTCP Port

This TLV is used to specify the local RTCP port number. If this port is not specified by the host during the VDAC-ONE card call setup, the source RTCP port is set one higher than the source RTP port, per RFC 1889. This TLV can be set during an active call.

Default: No default

TLV: 0x01E8 Source RTCP Port

Destination RTCP Port

This TLV is used to specify the remote RTCP port number. If this port is not specified by the host during the VDAC-ONE card call setup, the destination RTCP port is set to one higher than the destination RTP port, per RFC 1889. This TLV can be set during an active call.

Default: No default

TLV: 0x01E9 Destination RTCP Port

RFC 2198 (RTP Redundancy) Dynamic Payload Negotiation

The VDAC-ONE card and the IP Network Interface Series 2 card allow RFC 2198 dynamic payload negotiation for clear channel calls only.

You first must enable RFC 2198 using the RTP Payload Redundancy TLV (0x01D1) in the *Resource Attribute Configuration* message.

You can configure the default value at configuration time, or change this value per call, as follows:

- To configure the default RFC 2198 Codec Type, use the RFC 2198 Dynamic Payload Type TLV (0x01F2) in the *Resource Attribute Configuration* message (0x00E3).

- To change the default RFC 2198 per call, use the RFC 2198 Dynamic Payload Type TLV (0x01F2) within the Generic PPL ICB (0x1E) in the *Route Control* (0x00E8) or *Outseize Control* (0x002C) messages.

Default: 104

TLV: 0x01F2

Jitter Buffer Delay/Adaptation Rate

Overview

You can set the Jitter Buffer delay for each channel. Through the jitter buffer, you can tailor the delay for various IP network types, based on an application's sensitivity to delay and error.

You set the Jitter Buffer delay by configuring the following attributes with the *Resource Attribute Configure* message.

- Minimum Jitter Buffer Delay
- Maximum Jitter Buffer Delay
- Adaptation Rate

These parameters, especially the Minimum Jitter Buffer Delay, affect the overall packet error rate. In this context, the packet error rate is defined as the rate at which a particular active VDAC channel encounters an under-run condition. An under-run condition exists when the active VDAC channel attempts to obtain an Ethernet packet from the jitter buffer when no packet is available.

Based upon the jitter/network delay and the error packet rate, you can derive a ratio that describes the performance of the jitter buffer. The algorithm that the active VDAC channel uses to obtain packets from the jitter buffer is adaptive, taking into account the changing network latencies.

Adaptation Rate

If you configure the adaptation rate for higher sensitivity, the active VDAC channel algorithm approaches the maximum jitter buffer delay for excessive network delays and maintains that delay. If you configure the adaptation rate for lower sensitivity, the jitter buffer delay increases more gradually.

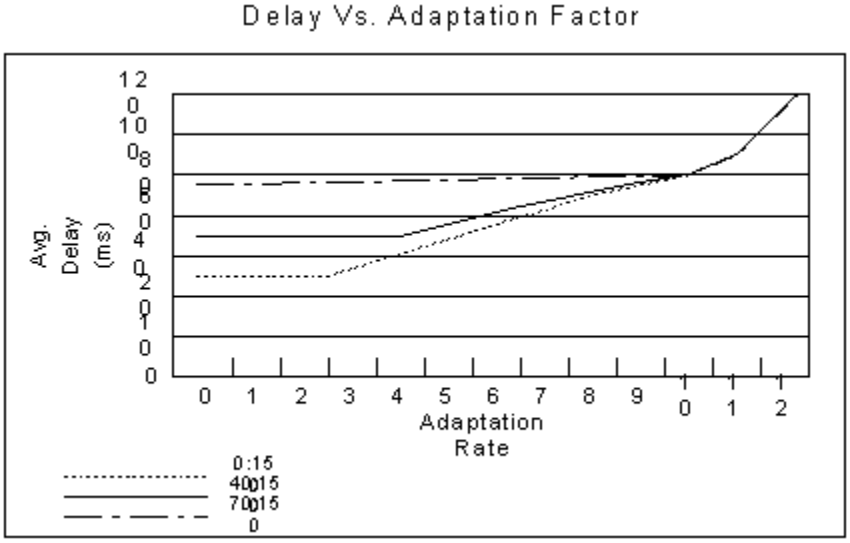
The opposite is also true once the algorithm has adapted to its new jitter buffer delay. Based upon the sensitivity setting, it either maintains that delay for a longer period with a higher adaptation rate, or it corrects more quickly with a lower adaptation rate.

Select the minimal safe delay for the network to which the VDAC-ONE card is connected. If the application can tolerate packet errors, use a lower adaptation rate and minimum jitter buffer delay. If the application cannot tolerate packet errors, use a higher adaptation rate and minimum packet delay.

Effect of Adaptation Rate on Jitter Buffer Delay

Figure 2-2 shows the effect of the adaptation rate on the average jitter buffer delay, for three different settings. The average delay increases with the higher sensitivity of the adaptation rate because, after network problems, the jitter buffer increases the delay and maintains the delay longer. This principle behavior remains the same for all networks, but due to differing latency times, the actual results vary across networks. With the jitter buffer set to 70 milliseconds minimal delay, the network shows a very slow increase in delay because the network has far fewer under-runs.

Figure 2-2 The Effect of Adaptation Rates on Average Jitter Buffer Delay

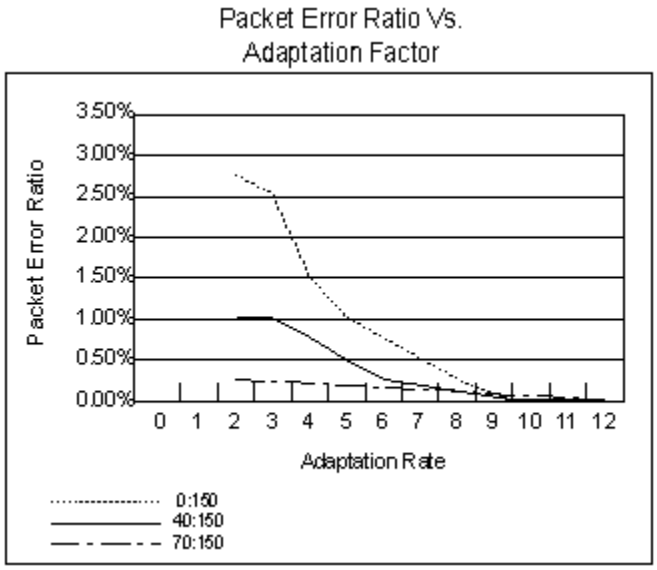


Packet Error Ratio vs. Adaptation Factor

When the jitter buffer delay setting is low, the adaptation factor significantly influences the error ratio, as shown in Figure 2-3. As the minimum jitter buffer delay is increased, fewer packet errors are encountered. As the adaptation factor increases and the average network delay worsens, the number of packet errors decreases.

As the sensitivity of the adaptation factor increases and network delays are encountered, the adaptive algorithm combats packet errors by acquiring and maintaining a higher jitter buffer delay, approaching the maximum delay value.

Figure 2-3 Packet Error Ratio vs. Adaptation Factor



Jitter Buffer Delay Configuration

Through the jitter buffer, you can tailor the delay experienced for each channel for various Ethernet network types, based on an application's sensitivity to delay and error.

To configure the Jitter Buffer Delay, use the *Resource Attribute Configure* message to configure the following attributes:

- Minimum Jitter Buffer Delay
- Maximum Jitter Buffer Delay
- Adaptation Rate

Minimum Jitter Buffer Delay

You configure the Minimum Jitter Buffer Delay by using the Minimum Jitter Buffer Delay TLV in the *Resource Attribute Configure* message. The range of valid values is 0 - 150 milliseconds. The default is 75 milliseconds.

You cannot configure the Minimum Jitter Buffer Delay on an established connection.

Maximum Jitter Buffer Delay

You configure the Maximum Jitter Buffer Delay by using the Maximum Jitter Buffer Delay TLV in the *Resource Attribute Configure* message. The range of valid values is 150-300 milliseconds. The default is 150 milliseconds.

You cannot configure the Maximum Jitter Buffer Delay on an established connection.

Adaptation Rate

You configure the Adaptation Rate by using the Adaptation Rate TLV in the *Resource Attribute Configure* message. The range of valid values is 0 - 12. The default rate is 7.

You cannot configure the Adaptation Rate on an established connection.

Fax Attributes

Overview

The VDAC-ONE card provides real-time fax relay, compliant with ITU T.38, over IP networks. This fax capability fully supports the ITU T.30 for Group 3 facsimile transmission protocol, with speeds of up to 14.4 Kbps. To remedy network packet loss and latency issues, the VDAC-ONE card employs fax spoofing techniques and packet redundancy schemes.

With the VDAC-ONE card fax relay mode, faxes can be sent over IP using as much as 40 to 50 percent less bandwidth than the full 64 Kbps used on a traditional circuit-switched network.

The VDAC-ONE card also provides a fax bypass mode for real-time fax transmission using high-speed coders. This capability is based on a proprietary protocol that uses RTP packets to transport payloads of compressed fax transmissions, with G.726 or G.727 waveform coders operating at a rate of 32 Kbps or higher.

Important! If a matrix switchover (or switchback) occurs during fax transmission, the transmission fails.

Configurable Fax Attributes

You can configure the following fax attributes:

- Fax Type: Fax Relay or Fax Bypass
- Fax Bypass Coder Type
- Fax Payload Redundancy

Fax Type Enable

Fax Relay

The VDAC-ONE card's behavior depends upon how it is configured for detecting fax transmissions. When the host has configured a channel for Fax Relay modes and the VDAC-ONE card detects that a call is not a voice call, the codec switches from the Voice Coder mode to Answer Tone mode and then to Fax Relay mode. The packets are sent to the network as Fax Relay packets that comply with T.38. When the fax transmission ends, the channel reverts to Voice Coder mode.

Fax Bypass

The Fax Bypass feature is proprietary to the VDAC-ONE card and is supported only between VDAC-ONE cards.

When the host has configured a channel for bypass operation and a fax transmission is detected, the codec automatically switches from the current voice coder to the rate of the configured Coder Type. When the fax transmission ends, the channel reverts to Voice Coder mode.

Default: Disabled

TLV: 0x01C5 Fax Type Enable

0x00 Disable

0x01 Fax Relay

0x02 Fax Bypass

If the fax type is set to disabled (RTP not changed) then the fax goes through as a voice call. Therefore, the fax transmission might not be successful depending on the codecs used.

A PPL Event Indication of Fax Start is sent to the host regardless of the value above.

Fax Bypass Coder Type

This attribute selects the codec to be used in a fax bypass scenario. Dialogic strongly advises selecting a data rate greater than 32 kbps.

Default: G.711 µLaw

TLV: 0x01C7 Fax Bypass Coder Type

Fax Payload Redundancy

Payload redundancy is a type of protection against network packet loss. By enabling redundancy, you can configure the CSP to send redundant packets to the network. For example, when you set the redundancy level to 2, the original payload is followed by two duplicate payload. Fax redundancy is supported in relay mode only.

Default: Disabled

TLV: 0x01D2 Fax Payload Redundancy

Fax Compatibility Mode

This TLV determines the VDAC-ONE card T.38 Fax Relay Compatibility mode: Backward Compatible mode or Interoperability mode. You must choose one mode or the other, not both. The Backward Compatible mode interoperates with the fax relay mode on the previously released VDAC-ONE card software. The Interoperability mode is used primarily to facilitate interoperation with third party T.38 fax relay devices. This TLV can be configured only before or during the setup of a VDAC-ONE card call, not during an active call.

Default: Backward Compatible Mode

TLV: 0x01E1 Fax Compatibility Mode

Source T.38 Port

This TLV is used to specify the local T.38 port number for fax relay. There is no default T.38 port value for this TLV. The host must select this port number as early as possible to ensure proper fax relay operation. If this port number is not specified in time, the fax relay fails. This TLV can be set during an active call.

Default: No default

TLV: 0x01E6 Source T.38 Port

Destination T.38 Port

This TLV is used to specify the remote T.38 port number for fax relay. There is no default T.38 port value for this TLV. The host must select this port number as early as possible to ensure proper fax relay operation. If this port number is not specified in time, the fax relay will fail. This TLV can be set during an active call.

Default: No default

TLV: 0x01E7 Destination T.38 Port

IP Connection Management

Overview

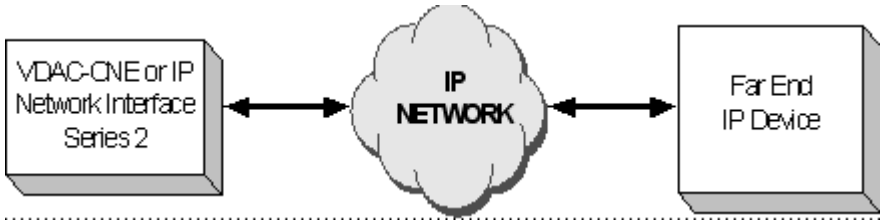
To establish an IP connection, you use the *Route Control* or *Outseize Control* message. For information on modifying an existing connection, see the *Dynamic Connection Management (2-33)* section.

Establishing a Two-way Connection

To establish a two-way connection, use the *Route Control/Outseize Control* message with the following TLVs:

- Source IP Address
- Source RTP Port Number
- Destination IP Address
- Destination RTP Port Number
- Connection Mode: two-way

Figure 2-4 VDAC-ONE Card Connection - Two-Way



One-Way Connection

There may be situations where a one-way IP connection is needed. For example, if Gateway A wants to reserve an IP address and an RTP port, it may establish one side of an IP connection with a one-way listen only. To establish the other side of the IP connection, VDAC A then sends a request, with the IP address and RTP port just reserved, to Gateway B.

Upon receiving the request, Gateway B establishes a two-way IP connection and sends Gateway A the IP address and RTP port that it reserved for the connection. In turn, VDAC A changes the current one-way listen only connection to a full two-way connection.

To establish a one-way connection, use the *Route Control/Outseize Control* message and include the Universal ICB with the Source IP Address and Source Port Number TLVs. If the source IP address/port number TLVs are missing, an error is returned.

The Destination IP Address/Port Number and Connection Mode TLVs are optional.

Receive-Only Connection

To establish a Receive-only connection, use the *Route Control/Outseize Control* message with the following TLVs:

- Source IP Address
- Source RTP Port Number
- Connection Mode: Receive-only
- Destination IP Address (Optional)
- Destination RTP Port Number (Optional)

Example: Message

```
 00 43 00 e8 00 00 ff 00 01 29 02 ff fe 03 03 00 33 00
12 00 02 27 92 00 04 0a 0a 24 15 27 93 00 04 00 00 00
01 02 1e 13 00 03 00 13 00 02 00 06 00 06 00 02 00 01
00 0f 00 01 0b 02 1e 07 00 01 01 db 00 01 01
```

Figure 2-5 VDAC-ONE Card Receive-only Connection

Transmit-Only Connection

To establish a Transmit Only connection, use the *Route Control/ Outseize Control* message with the following TLVs:

- NPDI Universal ICB
- Source IP Address TLV
- Source RTP Port Number TLV
- Destination IP Address TLV
- Destination RTP Port Number TLV
- Connection Mode (Transmit Only)

Example: Message

```
00 67 00 e8 00 00 ff \
00 01 29 02 ff fe (Router AIB) 03 (ICB count)
03 00 33 00 22 00 04 (Universal ICB)
27 92 00 04 0a 0a 24 15 (Source IP Address TLV)
27 93 00 04 00 00 00 01 (Source RTP Port TLVs)
27 94 00 04 0a 0a 24 16 (Destination IP Address TLV)
27 95 00 04 00 00 00 01 (Destination RTP Port TLV)
02 1e 13 00 03 00 13 00 02 00 06 00 06 00 02 00 01 00 0f
   00 01 0b (Router ICB) 02 1e 07 00 01
01 db 00 01 02 (Connection Mode TLV)
```

Figure 2-6 VDAC-ONE Card Connection - Transmit-only Connection

Dynamic Connection Management

Introduction To modify an existing connection, use the *Resource Attribute Configure* message with various combinations of TLVs.

Change Destination Only

To change the destination of the connection without changing the Connection Mode, send a *Resource Attribute Configure* message with the following TLVs:

- Address Element TLV
- Destination IP Address
- Destination RTP Port Number

Change Connection Mode

To change the Connection Mode, include the Connection Mode TLV. To also change the destination, enter the new Destination IP Address and RTP Port Numbers in the corresponding TLVs. To maintain the existing destination, enter the original destination information.

- Address Element TLV
- Destination IP Address
- Destination RTP Port Number
- Connection Mode

Use the Connection Mode TLV only when you want to modify a connection without changing the destination address. To modify a connection to a new destination, include the Destination IP Address and Destination RTP Port Number TLVs.

In hold mode, no voice path is transmitted or received.

> **Important!** You cannot modify the source IP address or RTP port of an existing connection. If you use the Source IP Address/RTP Port Number TLVs, the CSP returns an error and the connection remains unchanged.

Change Destination Only

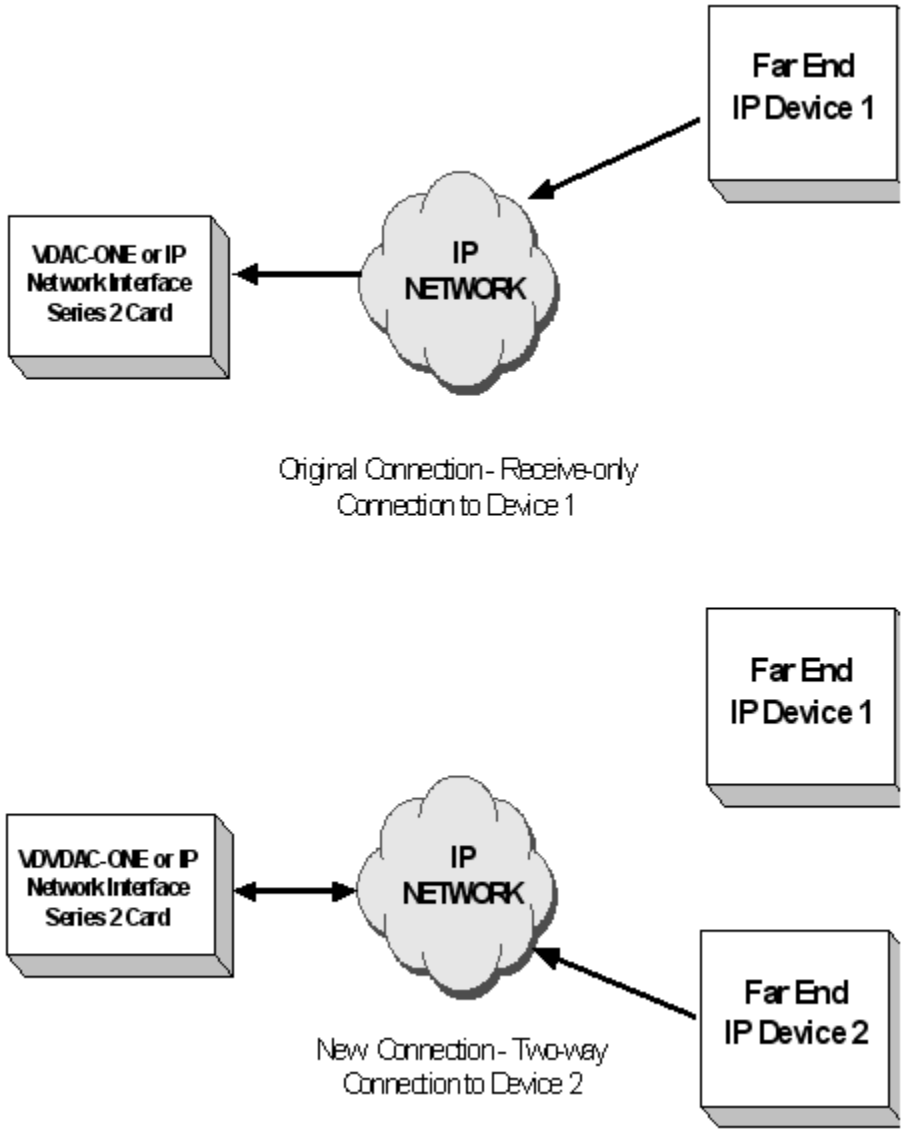
To modify the destination of an existing connection, use the *Resource Attribute Configure* message with the following TLVs:

- Address Element TLV
- Destination IP Address TLV
- Destination RTP Port Number TLV

Example Message

```
00 2a 00 e3 00 00 ff 00 01 01 01 04 00 04
00 09 00 05 0d 03 00 01 00 (Address Element TLV)
27 94 00 04 0a 0a 24 16 (Destination IP Address TLV)
27 95 00 04 00 00 00 01 (Destination RTP Port TLV)
```

Figure 2-7 Change Destination Only



Change Connection Mode Only

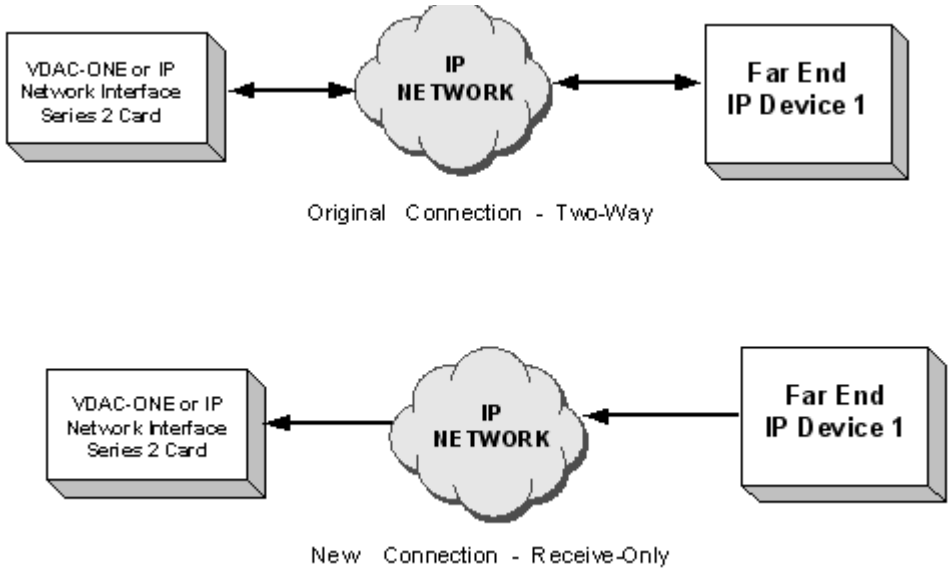
To modify the Connection Mode of an existing two-way connection to listen-only, use the *Resource Attribute Configure* message with the following TLVs:

- Address Element TLV
- Connection Mode TLV; Receive-only

Example Message

```
00 1a 00 e3 00 00 ff 00 01 01 01 04 00 02
00 09 00 05 0d 03 00 01 00 (Address Element TLV)
01 db 00 01 01 (Connection Mode TLV)
```

Figure 2-8 Change Connection Mode Only



Change Destination and Connection Mode

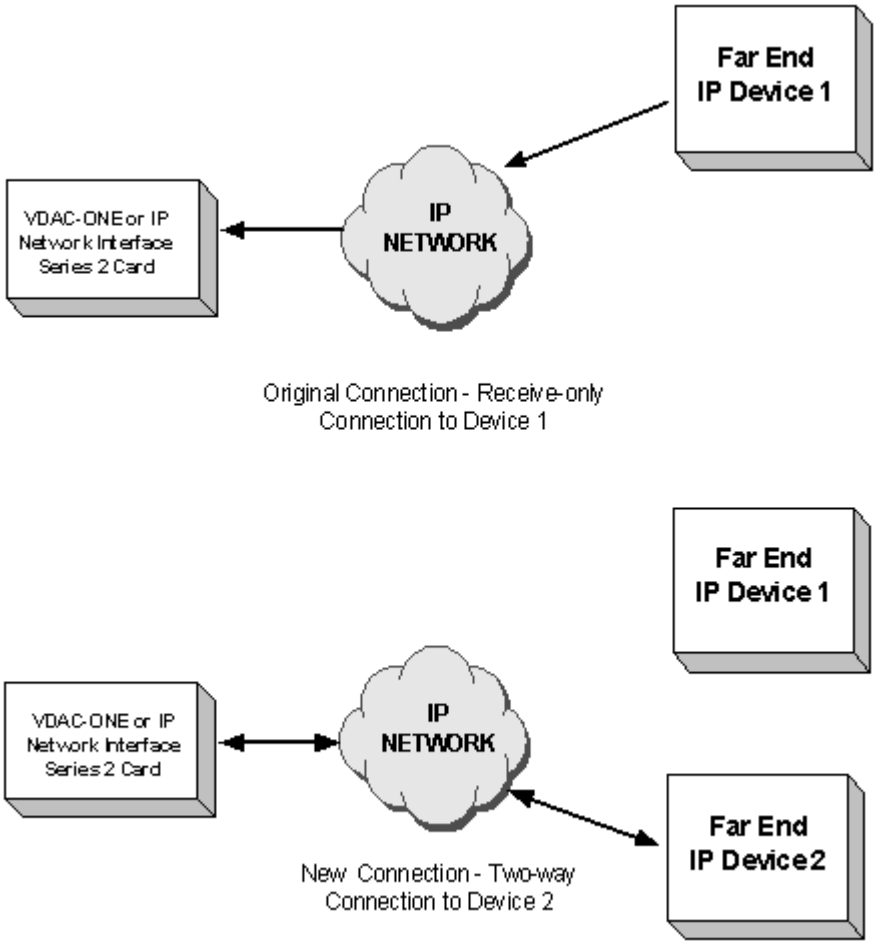
To change an existing listen-only connection to a two-way connection with a new destination, use the *Resource Attribute Configure* message with the following TLVs:

- Address Element TLV
- Destination IP Address TLV
- Destination RTP Port Number TLV
- Connection Mode TLV (Two-way)

Example Message

```
00 2a 00 e3 00 00 ff 00 01 01 01 04 00 04
00 09 00 05 0d 03 00 01 00 (Address Element TLV)
27 94 00 04 0a 0a 24 16 (Destination IP Address TLV)
27 95 00 04 00 00 00 01 (Destination RTP Port TLV)
   01 db 00 01 02 (Connection Mode TLV)
```

Figure 2-9 Change Destination and Connection Mode



Release Mode Configuration

Overview

Release modes are managed by the Channel Management (CH) PPL component of CSP Call Control. When a connection is terminated, the local and distant-end release modes are used to determine what action to take on the released channels.

The terms *local* and *distant* are relative to the channel being configured. To specify whether a channel is released or parked when the other channel in a connection releases, you configure the channel's local-end release mode. To specify whether the other channel in the connection is released or parked when that channel releases, you configure the channel's distant-end release mode.

When a channel terminates a connection, the CSP refers first to the local-end release mode of the other end of the connection, and then to the distant-end release mode of the channel that initiated the release. If either end is set to park, the channel parks, otherwise, it is released. The host is informed of the state of a channel with either a *Channel Released* or a *DS0 Status Change* message.

Distant-end Release Mode

When you configure a channel, use the distant-end release mode so that the distant end (the other channel in the connection) is not released when the connection is terminated. For example, you would not want to release an inbound channel that is connected to a Voice Response Unit's (VRU) *Please Wait* message before it is queued up for an available agent.

When the announcement is finished, the VRU channel is released for other calls and the inbound channel is parked. The distant-end release mode of the VRU channel is set to park so that when the VRU releases at the end of the message, all inbound channels connected to the VRU park.

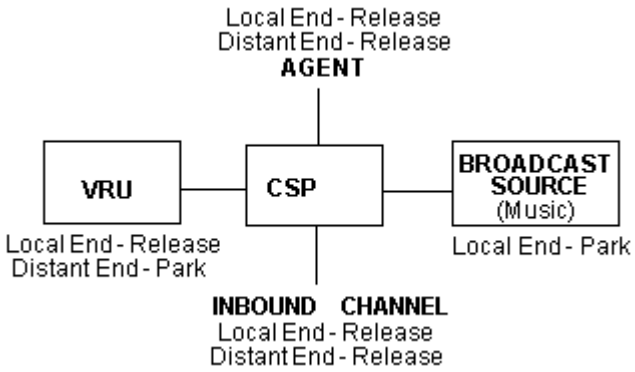
Local-end Release Mode

When you are configuring a channel, use local-end release mode so that the channel is not released when a connection is terminated. Consider the previous example for distant-end release mode. To configure a source to broadcast music to the channels waiting for an agent, the local-end release mode of the broadcast channel is set to park. When no channels are connected to the broadcast channel, it parks until another channel is connected. In contrast, if you allow the channel to release, it must be outseized after each connection is broken down.

Figure 2-10 illustrates the preceding examples. Both release modes for the inbound channel are set to release. Therefore, when the connection is torn down, it will be parked only if the distant-end release mode of the other channel is set to park (as is the case for the VRU and the Broadcast Source).

If the distant-end release mode of the other channel is set to release (as is the case with the agent channel) the inbound channel is released when the connection is torn down.

Figure 2-10 Release Mode Diagram



Configuration PPL Config Bytes

To configure Release Mode to either Park or Release, use the *PPL Configure* message to modify the PPL Config Bytes of the CH PPL component.

- Local-End Release Mode - Config Byte 3
- Distant-End Release Mode - Config Byte 4

Release Mode API Messages

You can also change release modes by using either the *Local End Release Mode Configure* or the *Distant End Release Mode Configure* message.

Loopback for Testing

Overview

To perform loopback on a VDAC-ONE card in a lab environment, specify the same source and destination IP addresses in a *Route Control* (or *Outseize Control*) message. The source and destination RTP port numbers must be different.

To set up a VoIP call using the same IP address, send *Route Control/ Outseize Control* messages to the same IP address with different port numbers. There are 40 channels per module, so you can establish up to 20 pairs of end-to-end calls per IP address. If you try to re-use a port number for a channel that is already being used for a call (RTP, RTCP, T.38) you receive the following error message:

```
0x4B: RTP Port in Use.
```

Example:

Route Control/Outseize Control Message 1:

- Source IP Address: A
- Source Port Number: 1000
- Destination IP Address: A
- Destination Port Number: 1004

Route Control/Outseize Control Message 2:

- Source IP Address: A
- Source Port Number: 1004
- Destination IP Address: A
- Destination Port Number: 1000

The call is now established.

Routing Examples

Routing Based on Criteria Type Source IP Address

For routing based on the Criteria Type, Source IP Address, use the Generic PPL TLV in the following format:

ICB Type	0x03
ICB Subtype	Generic PPL TLV (0x1E)
ICB Data Length	0x13
Number of TLVs	0x0003
Tag	Routing Method (0x0013)
Length	0x0002
Value	Data[0] 0x00 Criteria Type (MSB)
	Data[1] 0x08 Criteria Type (LSB)
Tag	Criteria Type (0x0008)
Length	0x0002
Value	Data[0] 0x27 (MSB) Source IP Address
	Data[1] 0x92 (LSB) Source IP Address
Tag	Router Protocol ID (0x000F)
Length	0x0001
Value	Data[0] 0x0B

Routing Based on Criteria Type Any IP Channel

For routing based on the Criteria Type, Any IP Channel, use the Generic PPL TLV in the following format:

ICB Type	0x03
ICB Subtype	Generic PPL TLV (0x1E)
ICB Data Length	0x13
Number of TLVs	0x0003
Tag	Routing Method (0x0013)
Length	0x0002

Value	Data[0] 0x00 Criteria Type (MSB)
	Data[1] 0x08 Criteria Type (LSB)
Tag	Criteria Type (0x0008)
Length	0x0002
Value	Data[0] 0x00 (MSB) Any IP Channel
	Data[1] 0x65 (LSB) Any IP Channel
Tag	Router Protocol ID (0x000F)
Length	0x0001
Value	Data[0] 0x0B

Routing Based on Group ID

For routing based on the Route Group ID, use the Generic PPL TLV in the following format:

ICB Type	0x02
ICB Subtype	Generic PPL TLV (0x1E)
ICB Data Length	0x13
Number of TLVs	0x0003
Tag	Routing Method (0x0013)
Length	0x0002
Value	Data[0] 0x00 Route Group ID (MSB)
	Data[1] 0x06 Route Group ID (LSB)
Tag	Route Group ID (0x0006)
Length	0x0002
Value	Data[0] (MSB) Route Group ID
	Data[1] (LSB) Route Group ID
Tag	Router Protocol ID (0x000F)
Length	0x0001
Value	Data[0] 0x0B

Sample Route Control / Resource Attribute Configure

The *Route Control* message below establishes a one-way listen only connection:

```
00 43 00 e8 00 00 ff \' Header
00 01 29 02 ff fe \' Router AIB
03 \                    ' ICB count
03 00 33 00 12 00 02 27 92 00 04 0a 0a 24 15 27 93 00 04
   00 00 00 01 \' Universal ICB
02 1e 13 00 03 00 13 00 02 00 06 00 06 00 02 00 01 00 0f
   00 01 0b \' Router ICB
02 1e 07 00 01 01 db 00 01 01 \' Connection Mode TLV
```

The *Route Control* message below establishes a one-way talk only connection:

```
00 67 00 e8 00 00 ff \' Header
00 01 29 02 ff fe \' Router AIB
03 \                    ' ICB count
03 00 33 00 22 00 04 27 92 00 04 0a 0a 24 15 27 93 00 04
   00 00 00 01 \
' Universal ICB (SRC IP/RTP)

27 94 00 04 0a 0a 24 16 27 95 00 04 00 00 00 01 \'
   Universal ICB  (DST IP/RTP)
02 1e 13 00 03 00 13 00 02 00 06 00 06 00 02 00 01 00 0f
   00 01 0b \' Router ICB
02 1e 07 00 01 01 db 00 01 02 \' Connection Mode TLV
```

The *Resource Attribute Configure* message below modifies the current connection to a one-way listen only connection:

```
00 1a 00 e3 00 00 ff \' Header
00 01 01 01 04 00 02 \'Router AIB
00 09 00 05 0d 03 00 01 00 \' Address Element TLV
01 db 00 01 01      \' Connection Mode TLV
```

The *Resource Attribute Configure* message below establishes a new one-way talk only connection:

```
00 2a 00 e3 00 00 ff \' Header
00 01 01 01 04 00 04 \'Router AIB
00 09 00 05 0d 03 00 01 00 \' Address Element TLV
27 94 00 04 0a 0a 24 16 \' Destination IP address TLV
27 95 00 04 00 00 00 01 \' Destination RTP port TLV
01 db 00 01 02      ' Connection Mode TLV
```

VDAC VoIP - 0x009C

This section contains information on PPL Component IDs and component-specific information for PPL Timers and PPL Events.

PPL Events Requests

Event ID	Event Description
0x01	Timer 1
0x02	Timer 2
0x03	Timer 3
0x04	Timer 4
0x0A	Layer 4/L3P Outseize Control
0x0B	Layer 4/L3P Clear Request
0x14	DSP Resource Available
0x15	DSP Resource Unavailable
0x16	DSP Resource Released
0x17	DSP Resource Inconsistency
0x19	RTP Port In Use
0x1A	Invalid Attributes
0x1B	Null Source in NPDI ICB
0x1E	Fax Start
0x1F	Fax End
0x23	Modem Start (not supported)
0x24	Modem End (not supported)
0x25	RTP Timer Expired
0x26	Modify Connection Attributes
0x27	Modify Connection Attributes Response
0x28	Media Inactivity Detection Timer Expired
0x403	Dummy Event
0x404	Channel In Service

Event ID	Event Description
0x405	Channel Out of Service

PPL Event Indications	
Event ID	Event Description
0x01	Received Fax Start Event
0x02	Received Fax End Event
0x05	Received Modem Start Event (not supported)
0x06	Received Modem End Event (not supported)
0x07	Received RTP Timer Expired Event
0x08	Received Media Inactivity Timer Expired Event
0x0A	Incoming Registration Request
0x0B	Registration Timer Expired
0x0C	Incoming Deregistration Request
0x0D	Registration Query Response

Important! When signaling is controlled by the H.323 software, these indications are not used by the host. Instead they are sent to the H.323 software which is controlling the call.

Purge Reasons

Reason	Description
203	Layer 4/L3P Clear Wait Timeout
204	DSP Resource Inconsistency (S13)
205	DSP Resource Inconsistency (S4)
210	DSP Resource Inconsistency (S11)
211	DSP Resource Released (S11)
212	DSP Resource Inconsistency (S21)
213	DSP Resource Released (S21)

PPL Timers The following table defines the new PPL Timers.

Timer Name and Hex ID	Default Value
DSP Response Wait Timer (0x01)	500 ms
Guard Wait Timer (0x02)	700 ms
Layer 4/L3P Clear Wait Timer (0x03)	4000 ms
Fax End Wait Timer (0x04)	5000 ms
Modem End Wait Timer (0x05) (not supported)	5000 ms
Fax Start Validation Timer (0x06)	100 ms
Modem Start Validation Timer (0x07) (not supported)	100 ms

3 IP Network Interface Series 2 Card

Purpose

This chapter provides the developer with software information for the IP Network Interface Series 2 card. (The IP Network Interface Series 2 card is sometimes referred to as the IPN-2 card.)

Important Notes Regarding IP Network Interface Series 2 Cards

Do Not Mix Profiles

You cannot mix VoIP resource profiles on the same IP Network Interface Series 2 card or in the same chassis.

255 Active Channels

For VoIP Resource Profile 2 on IP Network Interface Series 2, one channel on each VoIP module is reserved for ICMP Support. The full 256 channels per module can still be provisioned. The restriction is that only 255 channels can be active at one time. Attempting to use the 256th channel results in a negative acknowledgement of 0x28 (DSP Resources Not Available) to the *Outseize Control* (0x002C) or *Route Control* (0x00E8) API message. VoIP Resource Profile 1 does not have this limitation as it has an extra channel for ICMP support.

510 Simultaneous Calls

Even though 512 channels (16 spans, 32 channels per span) can be configured on a single IP Network Interface Series 2 card, only 510 simultaneous calls are allowed when using the default VoIP module resource Profile 2.

Upgraded ROMs

For Software Release 8.2.1, the ROMs were upgraded on the IP Network Interface Series 2 cards. Do not use Software Release 8.2.0 or earlier on the upgraded IP Network Interface Series 2 cards. The ROM information is listed below:

- Rev 05.01:000 (Release 8.2.1)
- Rev 05.00:014 (Release 8.2.0 or earlier)

Overview

Introduction

The IP Network Interface Series 2 card is the second generation VoIP line card used in the CSP. In the broadest sense, IP Network Interface Series 2 card is conceptually a VDAC-ONE card with higher channel densities. The IP Network Interface Series 2 card provides an easy migration path for VDAC-ONE users with minimal host impact.

Two Way Conversion

The IP Network Interface Series 2 card performs two-way conversion between circuit-switched data and packet-switched data. This conversion is required by packetized voice applications, such as the Voice over Internet Protocol (VoIP). The card also integrates media resources over IP technology.

Circuit-switched voice is converted to IP packets, using compression algorithms that can increase capacity toward the IP network side. You can have parameters modified for an individual call, often while the call is active, changing the quality of service, as needed.

Media Resources

Integrating media resources using IP technology provides many advantages. Typically, media resources are connected by T1, E1, or J1 interfaces that consume one 64 Kbps port per call, limiting the capacity of the system. The IP Network Interface Series 2 card integrates media resources over IP using the standards-based Real-Time Protocol (RTP).

Integrating media resources using standards-based technology also allows media resources to be shared between the CSP and other network infrastructure. Packet switching to media resources allows the application to benefit from voice compression, increasing capacity on the application. This flexibility allows the CSP and the applications to scale independently and incrementally, as needed, eliminating excess hardware.

Scalability - Single or Dual Modules

Dialogic offers the IP Network Interface Series 2 (IPN-2) card in Single and Dual Module options. These options make the card more cost effective at lower densities, and allow customers flexibility and cost effective redundancy at higher densities, as you pay for only the number of resources you require. You can purchase the IPN-2 card in the following options:

- IP Network Interface Series 2 with One Module – 96 ports that you can upgrade to 256/512 ports (depending on codecs)
- IP Network Interface Series 2 with Two Modules – 192 ports that you can upgrade to 512/1,024 ports (depending on codecs)
- Additional IP ports license are in 96 port increments

Important! If an IPN-2 card fails, its resources are available to the other IPN-2 cards but the host must first transfer those resources to the other cards with the *Assign Logical Span* (0x00A8) message. See Redundancy for more information on this process.

Licensing Channels

You download the license to the CSP Matrix Series 3 Card using the *Product License Download* (0x0079) API message. You query resources with the *Product License Query* (0x007A) message. The license key type is 0x3131 for both download and query messages. Refer to the License Key ICB 0x24 in the Information Control Blocks chapter.

The ability to license channels frees up more resources for use by the remaining line cards. When the CSP powers up, or when you insert an IP Network Interface Series 2 card, the CSP Matrix Series 3 Card is informed of the module count. This count determines the number of resources added to the resource pool and this count is saved on the CSP Matrix Series 3 Card. This pool of resources can be used by any IPN Series 2 card in the CSP. Therefore, a single IPN Series 2 card may get to use all resources or the resources may be shared among all IPN Series 2 cards.

If you need more resources and download a license to the matrix, the licensed resources are added to the resource pool and the license information is saved in the Software Locked Module table. As you assign logical spans, the resource count is decremented.

As logical spans are de-assigned from IPN Series 2 cards, the resource count is incremented by 32 and the space assigned to that span is freed up. When you remove an IPN Series 2 card, the resource pool is

decremented by the unlicensed resources not currently used by that card. For example if the card supplies 96 unlicensed channels and has a single span (32 channels) assigned to it, the resource pool is decremented by 64 when the card is removed. The timeslot occupied by the card is freed up. When a license downloaded to the matrix, its details are sent to the Standby CSP Matrix Series 3 Card.

Programmability

The IP Network Interface Series 2 card seamlessly integrates into a CSP system, appearing as a traditional circuit-based line card. The VoIP channels are divided into multiple 32 channel spans, allowing the host application to assign logical span IDs to the IP Network Interface Series 2 card just as it would an E-ONE or T-ONE line card.

> **Important!** Span/Channel is bound to an IP Address/Port during call setup. The *Route Control/Outseize Control* messages are used to initiate the call and bind a Span/Channel to the IP Address/Port. This association lasts for the duration of the call.

The IP Network Interface Series 2 card has its own resources to provide echo cancellation, silence suppression, and detection of fax tones. Multiple IP Network Interface Series 2 cards can populate one node, scale to the full capacity of the CSP, and interoperate with all other CSP cards.

> **Important!** The CSP supports a maximum of four IP Network Interface Series 2 cards per chassis with Profile 2 only. Profile 1 supports a maximum of two cards.

Adding the IP Network Interface Series 2 card to existing enhanced-service platforms allows applications to be quickly IP-enabled, providing a significant advantage over other systems that require developers to rewrite the application for packet transport. The ability to support both circuit- and packet-based transport enables developers to create entirely new services, such as text-to-speech and web-based applications.

Flexibility

The IP Network Interface Series 2 card has the flexibility to configure its VoIP modules to support multiple VoIP resource profiles. A resource profile defines the terminal capabilities of a VoIP endpoint (for example, a VoIP module). A VoIP resource profile's terminal capabilities consist of:

- The maximum number of VoIP channels supported.

- The list of supported VoIP codecs along with each codec's base, default and maximum packet rate.
- The maximum size of a channel's Jitter Buffer.
- The maximum size of a channel's Echo Canceller Tail Length.
- A list of supported VoIP channel features:

  Fax Relay

  Digit Relay (RFC 2833)

  RTP Redundancy (RFC 2198)
- A list of supported VoIP channel attributes along with their:

  Default Value

  Range of Valid Values

  Whether a module wide default can be set. If not, then the attributes can only be set during call establishment using *Route/Outseize Contro*l messages.

  Whether its value can be changed for an active call.

VoIP Resource Profiles

The VoIP capabilities of the VDAC-ONE and IP Network Interface Series 2 cards are defined by the resource profiles assigned to each IP endpoint (module). Based on the terminal capabilities of the profile, the host decides how each VoIP channel can best be provisioned. Listed below are the basic profile components. Refer to *VoIP Resource Profile Terminal Capabilities (3-51)* for detailed resource profile information.

Important! You cannot mix VoIP resource profiles on the same IP Network Interface Series 2 card or in the same chassis.

Profile 0 (VDAC-ONE only)

- codecs - G.711 + G.729 A/B + G.723.1 + G.726 + G.727 + T.38 + V.32
- Channels Per Module - 40
- Jitter Buffer Size - 300 milliseconds
- Echo Tail Length - 25 milliseconds

The IP Network Interface Series 2 card supports Resource Profiles 1 and 2.

Profile 1

- codec - G.711 only
- Channels Per Module - 512

- Jitter Buffer Size - 300 milliseconds
- Echo Tail Length - 64 milliseconds

Profile 2

- codecs - G.711 + G.729A/B + G.723.1 + G.726 + T.38
- Channels Per Module - 256

Important! In Profile 2 there are a total of 256 channels. However, one channel is used internally for ICMP support leaving 255 channels.

- Jitter Buffer Size - 300 milliseconds
- Echo Tail Length - 64 milliseconds

For the IP Network Interface Series 2 card, Resource Profile 2 is the default profile. This profile is compatible with the VDAC-ONE default profile (Profile 0). Using the default profile, the IP Network Interface Series 2 card provides 512 vs.160 VDAC-ONE channels. The *Resource Attribute Configure* (0x00E3) message allows the host to assign a resource profile to a VoIP module.

Redundancy

You can ensure reliability for IP switching at two levels:

- Link Level
- Card Level

Link Level

Each IP Network Interface Series 2 card has three external 100 Mbps Ethernet ports that are configured as a Link Aggregate Group (LAG), extending the effective bandwidth to 300 Mbps. In addition to the increased bandwidth, automatic link failover is provided transparently to the host.

> **Important!** In order to fully utilize the IP Network Interface Series 2 network interface, a Ethernet switch that supports Link Aggregate Groups is required.

Card Level

IP Network Interface Series 2 cards are not hardwired to spans and do not take timeslots from the CSP until they are configured. Card-level redundancy is accomplished with load sharing, where enough capacity is added to keep the system running when one card goes out of service.

Example

In the following example, the IPN-2 card in slot 5 can be considered the "primary" card and the IPN-2 card in slot 6 is the "standby" card. Each card has two modules and use Profile 2.

1. Use the *Resource Attribute Configure* message to configure slot 5 for Profile 2 and normal (non-gateway) mode.
2. The card in slot 5 takes 16 spans of the 32-channel timeslots from the CSP.
3. Use the *Assign Logical Spans* message to map the logical spans to the physical spans in slot 5.
4. If slot 5 fails, you take it out of service or physically remove the card from slot 5. The 16 spans of 32 channels are returned to the CSP.
5. Use the *Resource Attribute Configure* message to configure slot 6 for Profile 2 and normal (non-gateway) mode.
6. The card in slot 6 takes 16 spans of the 32-channel timeslots from the CSP.
7. Use the *Assign Logical Spans* message to map the logical spans to the physical spans in slot 6.

8. Service resumes using slot 6.

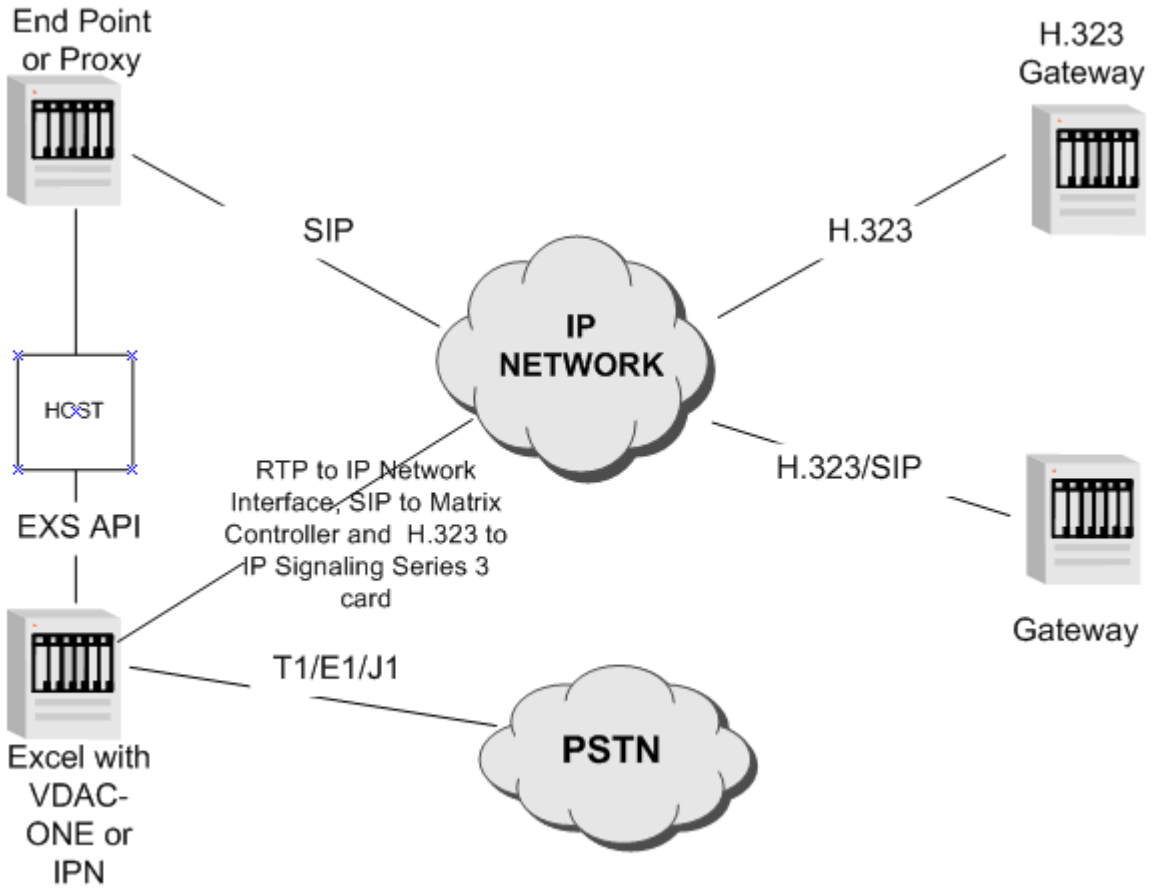
The process is the same for profile 1 or cards with one module. Just the number of spans taken from the CSP differ.

Cost-Effective Service Integration

Adding packet switching capabilities to the already flexible CSP provides many additional cost-saving and revenue-generating benefits. The ability to compress voice increases system capacity toward the IP network side, allowing the same amount of bandwidth to transport voice more cost-effectively, and allowing the excess bandwidth to support more customers. Using packet transport for media resources extends the capacity gains of voice channels to the resource, allowing each service to handle greater call volumes.

Integrating applications through IP provides scalability for the applications because they are no longer limited by the number of interface ports available. The standards-based transport technology allows you to easily add third-party applications. This integration method allows multiple, multi-vendor applications to operate on a single system. Other networks can also share these resources because RTP is a standards-based integration technology.

Figure 3-1 Protocol Interworking through the IP Network Interface Series 2 Card



VoIP Attributes

The IP Network Interface Series 2 card supports the following VoIP attributes which can be changed on active (dynamic) codec connections:

- Changing codec voice signals
- Receiving Real Time Protocol (RTP) packets that change codecs dynamically
- Enabling and disabling RTP redundancy
- Changing a codec payload type and payload size during mid call (see below)

Dynamic Change of Vocoders

The IP Network Interface Series 2 card allows you to dynamically change vocoders during a call.

The host or the signaling protocol sends the following TLVs in the *Resource Attribute Configure message* (0x00E3) to address the span/channel of the active call.

- RTP Payload Size (0x0101)
- RTP Payload Type (0x0100)

If the payload type or size is out of range, the CSP negatively acknowledges it with a status of 0x5F, Invalid Resource Attribute.

Be sure to specify both the payload type and payload size. If you omit the payload size, the default size for that payload type is used. If you just specify the payload size, leaving the payload type the same, there are potential problems. For example, the far end's jitter buffer could be almost empty and the change in the packet rate may cause a buffer under run condition.

Transmitted and Received Voice Signals and Packet Rates

The IP Network Interface Series 2 card transmit and receive paths are asymmetrical. This means that the received codec voice signals can be different from the transmitted codec voice signals. This also means that the packet rates can be different. For example, a channel can transmit G.711 at 25 milliseconds and receive G.723.1 at 60 milliseconds.

Packet Rates

The maximum packet rates and the base packet rate of each codec is lower for a IP Network Interface Series 2 card than for a VDAC-ONE card. For example, a IP Network Interface Series 2 card supports packet rates from 5 milliseconds to 30 milliseconds in increments of 5 milliseconds for G.711. The VDAC-ONE card supports packet rates from 20 milliseconds to 160 milliseconds in increments of 20 milliseconds for G.711.

Terminal Capabilities of a VoIP Module

The terminal capabilities of a VoIP module can be queried using the *Resource Attribute Query* (0xE4) message. The VoIP Terminal Capabilities TLV (0x01EA) returns information for a device server to populate its negotiation tables. The IP Network Interface Series 2 and VDAC-ONE cards support this TLV to provide a transparent interface to an external signaling agent.

Real Time Clock

IP Network Interface Series 2 card Real Time Clock is synchronized with the CSP Matrix Series 3 Card.

Gateway Mode

The IP Network Interface Series 2 card can be configured to either consume physical TDM timeslots or consume extended timeslots. Extended timeslots do not consume any of the chassis physical timeslots, but allow the IP Network Interface Series 2 to come in

service. This mode is called Gateway Mode. The IP Network Interface Series 2 card defaults to Gateway Mode. Refer to *Gateway Mode (3-90)*.

Licensing Channels

The default number of IP Network Interface Series 2 cards resources is 96 channels per installed module.You can purchase a license for extra resources in increments of 96 channels as needed.

Media Inactivity Detection (MID) Timer

The Media Inactivity Detection (MID) Timer feature used on the IP Network Interface Series 2 card has been enhanced from the VDAC-ONE card.

VDAC-ONE Card

The VDAC-ONE card uses two independent activity timers: the Real Time Protocol (RTP) Timer TLV (0x01EB) and the Media Inactivity Detection (MID) Timer TLV (0x1EC). The RTP Timer monitors a channel's media stream for the first valid User Datagram Protocol (UDP) packet. The MID Timer monitors a channel's media stream for any valid Real Time Control Protocol (RTCP) packet. If either of these timers expire, a *PPL Event Indication* (0x0043) message is generated with the following PPL Events, 0x07 and 0x08 respectively. If, for example, both the RTP Timer and the MID Timer are configured and no valid packets arrive, both PPL events are generated.

IP Network Interface Series 2 Card

The IP Network Interface Series 2 combines the RTP Timer TLV (0x01EB) and the MID Timer TLV (0x1EC) timers into a single Media Inactivity Detection (MID) Timer.

> **Important!** For consistency, the RTP Timer Timeout TLV (0x01EB), for the IP Network Interface Series 2, has been renamed the Initial Media Inactivity Detection (IMID) Timeout Value. The functionality of the IMID remains the same as the RTP Timer.

The feature provides for one MID Timer per channel that monitors activity on the incoming media stream. The host has the ability to specify different timeout values for monitoring the first valid packet versus subsequent packets.

The IP Network Interface Series 2 card MID Timer capabilities are as follows:

- The MID timer is refreshed upon the receipt of any valid UDP packet (for example, RTP and RTCP). The VDAC-ONE card only refreshes its MID timer on valid RTCP packets.
- The MID timer is initialized with the IMID Timeout Value (TLV 0x01EB). If the timer expires before a valid packet is received, a *PPL Event Indication* (0x0043) message is generated with PPL Event 0x07. Subsequently, when a valid packet is received, the MID timer is reloaded with the MID Timeout Value (TLV

0x01EC). If the timer expired while the MID Timeout Value is loaded, a *PPL Event Indication* (0x0043) message is generated with PPL Event 0x08.

- If the IMID Timeout Value is set to zero, the MID timer is initialized with the MID Timeout Value.

Important! In order to maintain backwards compatibility with the VDAC-ONE card under this condition, if the MID Timer expires before the first packet is received, a *PPL Event Indication* (0x0043) message is generated with PPL Event 0x08 instead of PPL Event 0x07.

- The MID Timeout Value can be dynamically updated using the *Resource Attribute Configure* (0x00E3) message. Updating the MID Timeout Value causes the MID Timer to be re-enabled using this new timeout value.

Important! If the MID Timeout Value is updated while the MID Timer is enabled, the new timeout value is loaded upon receipt of the next valid UDP packet.

- The MID Timer automatically disables itself when the Connection Mode TLV (0x01DB) is changed or not configured to receive packets. The MID Timer is automatically re-enabled when the Connection Mode TLV (0x01DB) is updated to receive packets.
- The MID Timer automatically disables itself when the channel switches to FAX Relay mode. The MID Timer is automatically re-enabled when the channel switches back to Voice Packet mode.

Important! When the MID Timer is automatically re-enabled, the IMID Timeout Value is used, if configured.

Similarities Between IP Network Interface Series 2 and VDAC-ONE Cards

Overview

A major requirement and goal for IP Network Interface Series 2 is to provide an easy migration path for VDAC-ONE customers. While 100 percent backwards compatibility is not possible, the following similarities exist between the IP Network Interface Series 2 and VDAC-ONE cards.

Host API Messages

There are no new host API messages created for the IP Network Interface Series 2 card.

Network Configuration

The network configuration using *IP Address Configure* (0x00E7) API message remains the same.

TDM Resource Provisioning

TDM resource provisioning remains the same.

VoIP Channels

All VoIP channels are translated into multiples of 32-channel spans.

Call Processing

Call processing using the *Route Control* (0x00E8) API message and the *Outseize Control* (0x002C) API message remains the same.

VoIP Resource Configuration

VoIP Resource Configuration using *Resource Attribute Configure (*0x00E3) API message remains the same.

Generated Alarms

All VDAC-ONE generated alarms are supported.

IP Network Interface Series 2/VDAC-ONE Interoperability

The IP Network Interface Series 2 card is interoperable with the VDAC-ONE card using the default IP Network Interface Series 2, Profile 2 (G.711, G.723.1, G.729A/B and T.38.)

Differences Between IP Network Interface Series 2 and VDAC-ONE Cards

Overview

The following differences exist between the IP Network Interface Series 2 and VDAC-ONE cards.

VoIP Modules

An IP Network Interface Series 2 card uses two VoIP modules, whereas a VDAC-ONE card uses four VoIP modules. Fewer modules mean fewer IP endpoints, making it easier to configure and maintain.

> **Important!** All host messages need to be updated to address VoIP modules 1 and 2 instead of VoIP modules 1, 2, 3, and 4 on the VDAC-ONE card.

Card Types

Card Type 101 (0x65) identifies a IP Network Interface Series 2 card.

Card Type 96 (0x60) identifies a VDAC-ONE card.

Card Type 104 (0x6D) identifies a IP Network Interface Series 2 card VoIP module.

Card Type 97 (0x61) identifies a VDAC-ONE card VoIP module.

Card Type 202 (0xCA) identifies a Multi-Function Media I/O card.

Card Type 207 (0xCF) identifies a VDAC I/O card.

VoIP Channel Attributes

The VoIP Resource Profile 2 on IP Network Interface Series 2 is the most similar to VDAC-ONE. The following are some of the major differences:

- The G.727 codec is not supported on the IP Network Interface Series 2 card.
- The base packet rates and maximum payload size for the voice codecs are different for the IP Network Interface Series 2 card. This difference has probably the biggest impact to the host if the default packet rates are not used. Since the base rate is different, the payload size, which is defined in multiples of the base rate, means something different for different codecs. For example, G.711 @ 20 milliseconds on a VDAC-ONE card uses a packet size of 1x (base rate of 20 milliseconds), whereas @ 20 milliseconds on a IP Network Interface Series 2 card uses a packet size of 4x

(base rate of 5 milliseconds). Please refer to *Table 3-6, IP Network Interface Series 2 Voice Coder Packet Rate Information* for specifics.

- The IP Network Interface Series 2 card is not interoperable with VDAC-ONE card in FAX Bypass Mode. The VDAC-ONE card uses a proprietary implementation.
- The IP Network Interface Series 2 card is not interoperable with VDAC-ONE card using the G.726 Voice codec.
- The Adaptation Rate TLV VoIP Attribute is not supported on the IP Network Interface Series 2 card. The rate at which the Jitter Buffer adapts is not configurable.
- The Fax Compatibility Mode TLV (0x01E1) is not supported.

Redundancy

The IP Network Interface Series 2 card has three external 100 Mbps Ethernet ports that are trunked together to form a single 300 Mbps Ethernet pipe. Automatic link fail-over is supported transparently to the host. This requires a trunking-compatible Ethernet switch in order to take advantage of the increased bandwidth.

Important! The IP Network Interface Series 2 card does not support Ethernet Link Redundancy.

Alarms

Alarms have been created that notify the host when an Ethernet link failure/recovery occurs, and when a Multi-Function Media I/O card removal/insertion is detected.

***Cross Connect Channel Message* (0x001A)**

The *Cross Connect Channel* message (0x001A) is supported only on the VDAC-ONE card. The IP Network Interface Series 2 card does not support this message.

Address Resolution Protocol (ARP)

The sending of ARP messages to resolve a channel's Destination Hardware Address (MAC address) is different. The VDAC-ONE card drops all incoming the UDP packets and generates a single ARP request per second for each VDAC-ONE channel until the Destination Hardware Address (MAC address) is received.

If a destination address is provided in the resource allocation request, the IP Network Interface Series 2 card does not acknowledge the resource allocation request until after the Destination Hardware Address is resolved. Note that the VDAC-ONE card acknowledges the resource allocation request immediately, before the Destination Hardware Address is resolved.

If the Destination Hardware Address can not be resolved, a negative acknowledgement of 0x24 (Outseize Failed Due To No Answer) is returned.

Important! If a destination address change is requested on an existing connection, the request will not be acknowledged until after the Destination Hardware Address is resolved.

Important! The IP Network Interface Series 2 card caches only the VoIP module's gateway Destination Hardware Address.

Summary of API Messages

API Messages

The following API messages support the IP Network Interface Series 2 card.

Card Population Query (0x0007)

IP Network Interface Series 2 card and Multi-Function Media I/O card are supported card types.

Outseize Control (0x002C)/*Route Control* (0x00E8)

The IP Network Interface Series 2 does not respond to the *Outseize Control* message until after it receives an answer from the Destination IP Address. If the destination's hardware address cannot be resolved, a negative acknowledgement (NACK) message of 0x24 (Outseize Failed Due To No Answer) is returned.

The host can optionally override the default VoIP Resource Attributes when sending down a *Route/Outseize Control* message. Please be aware that the list of VoIP Resource Attributes contained in the Generic PPL for VoIP (0x1E) Data ICB is dependent on the resource profile assigned to the Source IP Address. Refer to Chapter 3, VoIP Resource Profiles for more detailed information.

Channel Parameter Query (0x0080)

When querying the channel parameters of a IP Network Interface Series 2 channel, the PPL Protocol Name (Data[18-37]) returned is “VDAC2”

Card Status Query/ Report (0x0083/0x00A6)

The *Card Status Query/Report* messages support the IP Network Interface Series 2 card, Multi-Function Media I/O card, and IP Network Interface Series 2 VoIP module.

You can address the *Card Status Query* message to the I/O slot behind the IP Network Interface Series 2 card. The following are possible results:

- If the Multi-Function I/O card is present, the Card Type contains 0xCA.
- If an illegal I/O card is present the Card Type is 0xAA.

- If no I/O card is present the CSP nacked the message with status 0x61-Invalid Slot.

Alarm (0x00B9)

The text includes reference to the IP Network Interface Series 2 card.

Resource Attribute Configure (0x00E3)

Resource Attribute TLVs:

VoIP Resource Profile Assign (0x01E5)

The resource profile assigned to each VoIP module defines the terminal capabilities of the IP endpoint. This includes which VoIP Attributes are configurable, when VoIP Attributes are configurable, and the range of valid values for each VoIP Attribute. Please refer to Chapter 3, VoIP Resource Profiles for more detailed information.

Resource Attribute Query (0x00E4)

VoIP Resource Attribute TLVs:

VoIP Resource Profile Assign (0x01E5)

Querying this TLV results in the requested VoIP Module's VoIP resource profile information being returned.

VoIP Terminal Capabilities (0x01EA)

A module's terminal capabilities can be queried in one of two ways, by Module IP Address or Span/Channel of an active connection. Please refer to Chapter 3, VoIP Resource Profiles for more detailed information.

IP Address Query/Configure (0x00E6/0x00E7)

Ethernet Redundancy is not configurable for IP Network Interface Series 2 card and, therefore, does not support the following TLVs:

Ethernet Link Redundancy (0x09)

Activate Ethernet Port (0x0A)

Host/IP Network Interface Series 2 Card Integration Information

Overview The following table defines the basic information a host needs to integrate the IP Network Interface Series 2 card into a CSP.

Table 3-1 Host/IP Network Interface Series 2 Card Integration Information

Product Name	IP Network Interface Series 2 Card
IP Network Interface Series 2 card ID	101 (0x65)
IP Network Interface Series 2 card VoIP Module Board ID	104 (0x6D)
Multi-Function Media I/O card ID	202 (0xCA)
L3 PPL Component ID	159 (0x9F)
L3 PPL Protocol ID	BOOT Protocol = 11 (0x0B) VOCC VoIP Protocol = 12 (0x0C)
Host L3 PPL Download Support	Host Downloadable PPL Tables are not supported
TFTP Additions	VOCCVOIP_LOAD = vocc_ip.bin* = SAVE_LOAD_FALSE VOCCVM_H_LOAD = vocc_vmh.bin* = SAVE_LOAD_FALSE VOCCVM_M_LOAD = vocc_vmm.bin* = SAVE_LOAD_FALSE **Important!** *The actual .bin file names change off of the build release.
Default Resource Profile	G.711+G.726+ G.723.1+G.729A/B+T.38 (Profile 2)
Default Gateway Mode	Gateway Mode
PCM Companding Law	μ-Law
Control Interface	The IP Network Interface Series 2 card is controlled by the CSP Matrix Series 3 Card using the API message set.
Data Interface	VoIP data is transported using the three external 100Base-T (100 Mbps) Ethernet ports on the Multi-Function Media I/O card. All three ports are configured as a single Link Aggregate Group (LAG), extending the total bandwidth to 300 Mbps and providing automatic link redundancy. If interfaced with a LAG compatible Ethernet Switch, all three ports can be utilized. If not, only ONE of the three ports can be connected to the Ethernet network, if LAG is not used.

Startup Sequence/Basic Configuration

Overview

This section describes the startup sequence and basic configuration of the IP Network Interface Series 2 card. This card requires basic card configuration virtually identical to that of a VDAC-ONE card with two VoIP modules. The only additional steps are to first put the IP Network Interface Series 2 into the Normal Mode and adjust for the increase in channel density.

Startup Sequence

When a IP Network Interface Series 2 card is powered up, the card performs a “cold start” which requires the IP Network Interface Series 2 card to be completely reconfigured.

Important! The IP Network Interface Series 2 card system software running on the main circuit board and VoIP modules are stored in volatile memory and not maintained through power loss.

The exceptions are the IP network settings assigned to the main circuit board and VoIP modules. The network settings are stored in non-volatile flash memory and maintained through power loss.

When first applying power to the IP Network Interface Series 2 card, the following sequence occurs:

- The resident ROM code performs Power On Self Tests (POST) and requests a main circuit board image from the CSP Matrix Series 3 Card.
- The CSP Matrix Series 3 Card validates the board ID and retrieves the main circuit board image from either its local or remote store.
- Upon completing the image transfer, the ROM code validates the image and starts executing the system software.
- The system software proceeds through its startup logic which does the following:
  - Clears the configuration data and initializes software services
  - Confirms connection to I/O
  - Establishes Ethernet connections
  - Detects VoIP modules
  - Downloads the software image that will run on the detected VoIP modules from the CSP Matrix Series 3 Card based on the default IP resource profile assigned.

- Validates the VoIP module image and transfers it to each VoIP module. Each DSP processor on the VoIP module runs its internal diagnostics and notifies the main circuit board when the initialization is complete.
- After the main circuit board and VoIP modules complete their initialization sequences, a minimum set of diagnostics is run to verify that the IP Network Interface Series 2 card is operating correctly.
- The IP Network Interface Series 2 card signals to the CSP Matrix Series 3 Card that it is ready and provides its card status information.
- The IP Network Interface Series 2 card starts up in the Gateway Mode, which indicates the Time Division Multiplexing (TDM) resources on the chassis midplane are not allocated.
- The CSP Matrix Series 3 Card, upon receipt of the card status information, generates a *Card Status Report* message to the host, signaling that the IP Network Interface Series 2 card is in service and ready to be configured.
- Until a *Card Status Report* message is generated, communication between the host and IP Network Interface Series 2 card is not allowed.

Important Card Status Report Information

The *Card Status Report* (0x00A6) message provides the following information specific to the IP Network Interface Series 2 card.

Table 3-2 Card Status Report Information

Fields	Description
Card Type	Identified with Card ID 0x65
Card Status	It is important that the IP Network Interface Series 2 card contains the following card status: Card is in service (Bits 2 and 3) No faults have been logged (Bit 5) Confidence test has been passed (Bit 6) No hardware failures (Bit 7)
Card Information	The card information contains current span assignments just like all line cards. The maximum number of span changes are based off of the VoIP resource profiles assigned to each VoIP module. From a cold start, the default VoIP resource profile is Profile 2 which supports a maximum of 16 spans.
Hardware Configuration	Similar to VDAC-ONE card. The only difference is that the resource profile assigned to each module is provided.

Example of Card Status Report Message

```
00 71 00 a6 00 0c ff            ' Message Header
00 00                           ' Status
00 01 01 01 03                  ' Slot AIB
65                              ' Card Type (IP Network Interface Series 2)
11                              ' Card Status (Cold Start)
00                              ' Confidence Test Results
58 05                           ' PC Artwork and Function Revision (X05)
05 00                           ' ROM Major.Minor Revision (5.0)
08 02                           ' RAM Major.Minor Revision (8.2)
19 48                           ' Serial Number (6472)
00 40                           ' Ram Size (64 Megs)

                                ' CARD INFORMATION
00 10                   ' Number of spans (16)
0c 02 ff ff   0c 02 ff ff   0c 02 ff ff   0c 02 ff ff
0c 02 ff ff   0c 02 ff ff   0c 02 ff ff   0c 02 ff ff
0c 02 ff ff   0c 02 ff ff   0c 02 ff ff   0c 02 ff ff
0c 02 ff ff             0c 02 ff ff   0c 02 ff ff   0c 02 ff ff

                                ' HARDWARE CONFIGURATION

02                              ' Number of Modules
02 00                           ' Number of Channels
03                              ' Installed Module Bitmap
05 05 02                        ' Module State
02 02 0f                        ' Assigned Resource Profile
01                              ' Normal Mode
00 00 00 00 00                  ' reserved
65                              ' Board ID
0a                              ' Build #10
00 00                           ' tag
08                              ' cpu speed (200Mhz)
00                              ' reset reason
```

IP Network Interface Series 2 Card Configuration Data

Configuration data on the IP Network Interface Series 2 card is categorized by the following three types:

Type 1 – Non-volatile Configuration Data

Type 1 configuration data is stored in non-volatile flash memory. Once configured, Type 1 data is maintained through power loss.

> **Important!** It is recommended that all Type 1 data be configured before configuring Type 2 and Type 3 configuration data.

The following configuration data is defined as Type 1 on the IP Network Interface Series 2 card:

IP Network Settings

The IP Network Interface Series 2 card's main circuit board and VoIP modules all interface with the Ethernet network using Internet Protocol (IP). The IP Network Interface Series 2 card will not be operational until these network settings are configured. Note the following:

- Changing the main circuit board's IP network setting causes the IP Network Interface Series 2 card to reset.
- Changing the VoIP module's IP network settings cause only the impacted VoIP module(s) to reset.

Important! Type 1 configuration data is maintained through a cold start. The host reset of configuration data does NOT reset Type 1 configuration data.

- Changing Type 1 configuration data does not affect Type 2 and Type 3 configuration data.
- Changing Type 1 configuration data causes the Configuration Tag to be cleared.

Type 2 – Configuration Data Impacting TDM Resource Allocation

Typically, a line card requires a fixed number of TDM resources. For example, there are 8-span and 16-span T-ONE and E-ONE line cards. A single T-ONE line card cannot be configured both as an 8-span and 16-span line card.

The IP Network Interface Series 2 card is different in that, based on the host configuration, the required number of TDM resources changes. For example, assigning VoIP Resource Profile 2 to the VoIP modules configures the IP Network Interface Series 2 card as a 16 span line card. Assigning VoIP Resource Profile 1 to the VoIP modules configures the IP Network Interface Series 2 card as a 32-span line card. Because of this and the ability to support more profiles in the future, the IP Network Interface Series 2 card cold starts in Gateway Mode. This allows the host to assign the appropriate VoIP Resource Profiles to the VoIP modules before taking up TDM timeslots allocated for other line cards.

The following configuration data is defined as Type 2 on the IP Network Interface Series 2 card:

Gateway Mode

The IP Network Interface Series 2 card can be configured to either consume physical TDM timeslots or consume extended timeslots. Extended timeslots do not consume any of the chassis physical timeslots, but allow the IP Network Interface Series 2 to come in service. This mode is called Gateway Mode. The IP Network Interface Series 2 card defaults to Gateway Mode. Refer to *Gateway Mode (3-90)*.

VoIP Resource Profiles

Both VoIP modules on the IP Network Interface Series 2 card are assigned the same VoIP Resource Profile, which defines the terminal capabilities of the module and the number of VoIP channels supported. The IP Network Interface Series 2 VoIP module defaults to VoIP Resource Profile 2, requiring 512 TDM timeslot resources.

Note the following:

- Changing any Type 2 configuration data causes the IP Network Interface Series 2 card to reset and clear all Type 3 configuration data, including the configuration tag.
- Type 2 configuration data is maintained through a push button reset (warm boot), but not a power up reset (cold boot).

The following conditions cause Type 2 configuration data to be reset:

- Host reset of configuration data using *Reset Configuration* (0x000B) message.
- Taking the card OOS using *Service State Configure* (0x000A) message.
- Pressing the Stop button on the IP Network Interface Series 2 card front panel.

Type 3 – Configuration Data Not Impacting TDM Resource Allocation

Type 3 configuration data is what is traditionally considered “battery backed” configuration data. The reset of Type 3 configuration data results in the “invalid battery-backed configuration data” bit to be set in the Card Status field of a Card Status Report/Query message.

The following configuration data is defined as Type 3 on IP Network Interface Series 2 card:

VoIP Resource Attribute Data

A VoIP module's terminal capability consists of a set of VoIP Resource Attributes, of which most of them the host can configure default values. See detailed Resource Profile Terminal Capabilities in this chapter for default values.

Logical Span ID Assignments

All VoIP channels are divided into 32-channel spans. The host can assign Logical Span IDs to each VoIP span.

Spans Service State

Once Logical Span IDs are assigned, the host can configure each span's service state. The service state of each span can be changed between Out of Service (OOS) and In Service (INS).

Channel Service State

Once spans are brought in service, the host can configure each channel's service state. The service state of each channel can be changed between Out of Service (OOS) and In Service (INS).

Note the following:

- Changing Type 3 configuration data causes the Configuration Tag to be cleared.
- Type 3 configuration data is maintained through a push button reset (warm boot), but not a power up reset (cold boot).

The following conditions causes Type 3 configuration data to be reset:

- Host reset of configuration data using *Reset Configuration* (0x000B) message.
- Taking the card OOS using *Service State Configure* (0x000A) message.
- Pressing the Stop button on the IP Network Interface Series 2 front panel.
- Change in Type 2 configuration.

Important! Configuration of Type 2 data MUST be completed prior to configuration of Type 3 data.

Basic Configuration

Once the IP Network Interface Series 2 card is in service, it is ready to be configured. Follow the steps below to configure the card. The messages involved appear in parenthesis. Refer to the *API Reference* for additional information on the messages and TLVs mentioned.

1. Configure IP addresses, subnet masks, and gateway IP addresses (*IP Address Configure* (0x00E7).
   - On a fully-populated IP Network Interface Series 2 card (with two VoIP modules) you assign three IP addresses/subnet masks (one to the main circuit board and one to each of the VoIP modules). See your IT/network administrator to ensure that you use the correct values for your network configuration.
2. Assign VoIP Resource Profiles and configure Gateway Mode (*Resource Attribute Configure* (0x00E3)

   Important! You must configure Gateway Mode before you perform subsequent configurations.

3. Configure Default VoIP - *Resource Attribute Configure* (0x00E3)
4. Assign Logical Span IDs - *Assign Logical Span ID* (0x00A8)
5. Bring spans in service - *Service State Configure* (0x000A)
6. Bring channels in service - *Service State Configure* (0x000A)

Configuring IP Address and Subnet Mask

The IP address is 32 bits. IP addresses and subnet masks can be assigned using the *IP Address Configure* (0x00E7) message.

If the host sends an *IP Address Configure* (0x00E7) message with an IP address and subnet mask that differ from the values stored in the Electronic Erasable Programmable Read Only Memory (EEPROM), the new values will overwrite the stored values.

If the host assigns the same IP address and subnet mask, the message is acknowledged and the card does not need to be reset, so the host can send the *IP Address Configure* message whenever it is necessary.

For the new IP address and subnet mask to take effect, you must reboot the card. You can perform this step by including the Engage IP TLV in the *IP Address Configure* message.

The subnet masks default to the values shown in the following table.

Table 3-3 Default Values

Class	IP Address	Subnet Mask
A	1.0.0.0 - 127.0.0.0	0xFF000000
B	128.0.0.0 - 191.0.0.0	0xFFFF0000
C	192.0.0.0 - 254.0.0.0	0xFFFFFF00

Assigning Dynamic IP Addresses

You can dynamically assign an IP address to a VoIP module without affecting the other modules on the IP Network Interface Series 2 card. You configure the IP address on a module by using the *IP Address Configure* message, with the IP Address/Subnet Mask TLV (0x0001).

To implement the new IP address at a later time, omit the Engage IP or Reset IP TLV and then send them separately when you want to reconfigure.

Re-Assigning an IP Address

When IP addresses are re-assigned, the TCP/IP stack must be re-initialized. To properly re-initialize the stack, you must reboot the board or module using one of the following TLVs in the *IP Address Configure* message:

- Engage IP TLV (0x0002)
- Reset IP TLV (0x0003)

Gateway IP Address

You can assign Gateway IP addresses to the IP Network Interface Series 2 card, the VoIP modules, or both. Use the *IP Address Configure* message to assign a Gateway IP address by using the Gateway IP (0x0005) TLV in the *IP Address Configure* message.

Example *IP Address Configure* (0x0005) message:

```
00 36 00 e7 00 00 ff                    ' Message Header
00 01 01 01 03                          ' Slot AIB
00 05                                   ' Number of TLVs
01 09 FF 0a 0a 19 1E ff ff ff 00        ' Main Board IP/Subnet
01 09 00 0a 0a 19 1f ff ff ff 00        ' VoIP Module 0 IP/Subnet
```

```
01 09 01 0a 0a 19 20 ff ff ff 00     ' VoIP Module 1 IP/Subnet
05 05 05 0a 0a 19 01                ' Gateway Address - All
02 00                               ' Engage
```

Assign IP Resource Profile and Configure Gateway Mode

Each VoIP module on the IP Network Interface Series 2 card defaults with the VoIP Resource Profile 2 assigned. Profile 2 is the most similar to the VDAC-ONE card in terms of terminal capabilities. Since each VoIP module supports multiple VoIP Resource Profiles, the IP Network Interface Series 2 card defaults to Gateway Mode. This allows the host to configure each VoIP module's resource profile before requesting physical TDM timeslot resource from the CSP Matrix Series 3 Card.

Once the host decides which VoIP Resource Profile to assign to both VoIP modules, a single *Resource Attribute Configure* (0x00E3) message can be sent to assign the VoIP Resource Profile and the Gateway Mode. The IP Network Interface Series 2 card resets and the CSP Matrix Series 3 Card allocates physical TDM timeslots.

> **Important!** When sending down the *Resource Attribute Configure* message, the Address Element TLV (0x0009) must NOT be included in the TLV list. Leaving the Address Element TLV out of the TLV list indicates that the Resource Attributes be addressed to the IP Network Interface Series 2 card itself rather than to a VoIP module or
> span/channel.

Example *Resource Attribute Configure* (0x00E3) message

```
00 18 00 e3 00 00 ff          ' Message Header
00 01 01 01 03                ' Slot AIB
00 02                         ' Number of TLVs
01 d0 00 01 02                ' Gateway Mode TLV - 0x01 Gateway, 0x02 Normal
01 e5 00 03 ff 02 00          ' VoIP Resource Profile TLV - Assign Profile 2 to all modules
```

Configure Default VoIP Resource Attributes

Once the configuration of the Type 2 data is complete, Type 3 data can start being configured. There are a number of VoIP Resource Attributes associated with each VoIP module in which the host can change their default values. Refer to each resource profile's detailed terminal capabilities later in this chapter. When VoIP connections are established using the *Route Contro*l (0x00E8) or *Outseize Control* (0x002C) messages, the host can specify values for each VoIP Attribute. For all attributes not specified in the *Route Control* message, the configured default values are used.

The default VoIP Resource Attributes on a VoIP module can be configured using the *Resource Attribute Configure* (0x00E3) message. Addressing the VoIP Resource Attributes to a specific module is done

by including the Address Element TLV in the list of VoIP Attribute TLVs. The Address Element TLV is actually an AIB embedded into a TLV. The IP Address (0x3E) AIB is used to specify which VoIP module default attributes are to be updated.

Example *Resource Attribute Configure* (0x00E3) message

```
00 25 00 e3 00 00 ff                    ' Message Header
00 01 01 01 03                          ' Slot AIB
00 04                                   ' Number of TLVs
00 09 00 06 3e 04 0a 0a 19 1F' IP Address Element Block
01 03 00 01 00                          ' Disable Echo Cancellation
01 c5 00 01 02                          ' Enable FAX Bypass
01 c7 00 01 05                          ' Bypass Coder - G.726@40kbps
```

Assign Logical Span IDs

There are no physical spans on a IP Network Interface Series 2 card. Instead, each module has a pooled resource of VoIP channels (the number of resources available is dependent on the resource profile assigned). A VoIP channel can dynamically bind to any TDM timeslot on the chassis and any remote RTP destination on the network. In order for the IP Network Interface Series 2 card to connect to a TDM timeslot, Logical Span IDs have to be assigned. The host can assign Logical Span IDs to cover all of the VoIP channels. To ease resource management, the channel densities of all VoIP modules, regardless of the resource profiles, are multiples of 32 channels.

For the default VoIP Resource Profile 2, a maximum of 16 logical span IDs can be assigned to the IP Network Interface Series 2 card. This is achieved using the *Assign Logical Span* (0x00A8) message.

Example *Assign Logical Span* (0x00A8) message

```
00 0d 00 a8 00 00 ff        ' Message Header
00 01 11 04                 ' Logical/Physical Span
00 A8                       ' Logical Span ID
03                          ' IP Network Interface Series 2 Slot
00                          ' Physical Span 0
```

Bring Spans in Service

Once the logical span IDs have been configured, the spans need to be brought in service before they are integrated into the system to be used by call control. Bringing a IP Network Interface Series 2 card span in service is no different than bringing a span on any other line card in service. This is done using the *Service State Configure* (0x000A) message.

Example *Service State Configure* (0x000A) message.

```
00 0c 00 0a 00 00 ff        ' Message Header
00 01 0c 02                 ' Logical Span AIB (0x0c)
00 a8                       ' Logical Span ID
f0                          ' Action - Bring In Service
```

Bring Channels in Service

Each IP Network Interface Series 2 card logical span consists of 32 VoIP channels. Once the logical spans are in service, the channels can be brought in service. Bringing the channels in service completes the basic configuration of the IP Network Interface Series 2 card. The host can now use these VoIP channels using call control.

Bringing the channels in service is very similar to bringing spans in service. The *Service State Configure* (0x000A) message is used. The one difference is that all 32 channels can be brought in service at one time.

Example *Service State Configure* (0x000A) message.

```
00 13 00 0a 00 00 ff        ' Message Header
01 02                       ' Range of Channels AIB (0x0d)
0d 03 00 a8 00              ' Originating Channel (span 0xa8, channel 0)
0d 03 00 a8 1f              ' Terminating Channel (span 0xa8, channel 31)
f0                          ' Action - Bring In Service
```

Flexible RTCP and T.38 Port Selection

You can choose any port number for RTCP and T.38 connections. By using the RTCP or T.38 port TLVs, the host explicitly specifies port values at one of the following three points:

- Before the call is connected (with the *Resource Attribute Configure* (0x00E3) message and the IP Address AIB)
- During call setup (with the *Route Control* (0x00E8) and *Outseize Control* (0x002C) messages)
- During an active call (with the *Resource Attribute Configure* (0x00E3) message and the Span/Channel AIB)

This flexible port selection scheme reduces or eliminates interoperability issues with other gateways and IP-enabled switches. It also aligns with the requirements set forth by the ITU-T H.323 Annex D standards, as well as other protocols such as SIP.

Considerations

The IP Network Interface Series 2 card does not automatically select the T.38 ports. Instead, the host must select the source and destination T.38 ports. The IP Network Interface Series 2 card is aware of a fax whenever the Fax Relay mode is enabled. The IP Network Interface Series 2 card detects the CED tone from the PSTN or T.38 packets from an IP network. The card then alerts the host with a PPL Event of a Fax Start. The channel then transitions from voice to fax mode and starts generating only T.38 packets. At this point, the host must select the T.38 port as quickly as possible. Otherwise, the IP Network Interface Series 2 card discards any T.38 packets. The fax machines at each endpoint will then stop because an appropriate response has not been received, and the fax transmission fails.

DTMF over IP Considerations

In Time Division Multiplex (TDM) networks, Plain Old Telephone Service (POTS), and cellular phones generate DTMF digits that are transmitted in-band by 64 kbps, dedicated-circuit switched ports. You can input DTMF digit responses, which are used for authentication and to branch through application prompts (for example, voice mail or prepaid applications). These DTMF digit responses are transmitted over the TDM network and presented to the CSP and host application in a prescribed format that is supported by the current applications.

The IP endpoints, whether SIP or H.323, must provide the same information through the IP network to the CSP and to the partner application. This way, the DTMF responses remain transparent to the host application, whether the user is using a POTS phone, cellular phone, or IP endpoint to interact with the voice mail system.

There are several methods of providing this DTMF capability in an IP network:

- In-Band
- Low bit-rate codecs with RFC 2833 (special RTP packets) Refer to *RFC 2833 Multi-Unicast (6-4)*.
- Out-of-Band using IP protocol signaling (SIP).

In-Band IP-to-IP Digit Transmission

Regardless of the audio codec scheme, you can pass in-band digits from IP-to-IP (from the IP Network Interface Series 2 card to another IP Network Interface Series 2 card or another IP endpoint that supports RFC 2833).

IP Network Interface Series 2 Card over the Exnet® Ring

The IP Network Interface Series 2 card over the Exnet® ring is similar to any other resource over the ring. Spans are assigned on the IP Network Interface Series 2 card and this span information is shared among all nodes. When the host sends a message to connect to a IP Network Interface Series 2 card span/channel on a remote node, the system makes the connection to the remote node.

Address Resolution Protocol (ARP)

The IP Network Interface Series 2 card supports the Address Resolution Protocol (ARP) of TCP/IP, which is used to obtain the physical address (Ethernet address) of an endpoint when only the logical address (IP Address) is known.

The CSP broadcasts an *ARP Request* message to the network with the known IP address. The corresponding endpoint responds with its Ethernet address and the connection is made.

ARP Cache Table

The main circuit board on the IP Network Interface Series 2 card maintains an ARP Cache Table for each module. An ARP Cache entry contains the IP Address and Ethernet address of remote endpoints. These tables allow connections to be made to the remote endpoint

without sending the *ARP Request* message each time. An IP Network Interface Series 2 card ARP Cache Table contains only a cache of Gateway Addresses.

You can query the content of a module's ARP Cache Table by using the *ARP Cache Query* (0x00FC) message. The ARP Cache Table information is sent to the host in one or more *ARP Cache Report* (0x00FB) messages.

ARP Cache Aging

With the ARP Cache Aging feature, entries in the ARP cache are deleted after a preset interval. Each entry in the ARP cache has an associated time-to-live. The ARP Cache Aging timer is set to five minutes. The ARP Cache manager continually increments the time-to-live field, and discards the entry when the value reaches the aging timer value. Entries are removed based solely upon the time-to-live value, and not upon the frequency that the entry was used.

ARP Cache Query

This feature allows the host to use the *ARP Cache Query* (0x00FC) message to query the ARP cache of any VoIP module. The CSP responds with one or more *ARP Cache Report* (0x00FB) messages that contain all existing ARP cache entries.

ARP Cache Flushing

You can manually remove entries from an ARP Cache Entry Table by using the Flush ARP Cache Entry TLV in the *ARP Cache Query* (0x00FC) message.

Internet Control Message Protocol (ICMP) Message

The IP Network Interface Series 2 card supports the Internet Control Message Protocol (ICMP), which reports errors related to IP packet processing. If the IP Network Interface Series 2 card attempts to connect to a remote endpoint and receives the ICMP message Destination Unreachable, it flushes the entry for that IP address from the ARP Cache Table, and sends an *ARP Request* message to obtain a new Ethernet address for the entry. When the IP Network Interface Series 2 card receives the new Ethernet address, the connection is made and the ARP Cache Table is updated. This process occurs without host intervention or notification.

The IP Network Interface Series 2 can generate the following ICMP message types:

- Destination Unreachable
- Echo Request (Ping Request)
- Echo Reply (Ping Reply)

The IP Network Interface Series 2 can process the following incoming ICMP messages:

- Destination Unreachable
- Echo Request (Ping Request)
- Echo Reply (Ping Reply)

TDM and Packet Resource Information

Overview

The IP Network Interface Series 2 card converts between packet and circuit switched data that consumes a finite set of resources. The host needs to be aware of how the IP Network Interface Series 2 card interfaces with the Time Division Multiplexing (TDM) and packet resources in order to efficiently provision them.

TDM Resources

The IP Network Interface Series 2 card has a one-to-one relationship between its VoIP channels and its TDM timeslots. Packet data is defined in terms of channels and circuit data is defined in terms of timeslots. If the IP Network Interface Series 2 card supports 1024 VoIP channels, it also needs 1024 TDM timeslots.

In the CSP, the IP Network Interface Series 2 card is like a T-ONE/E-ONE line card, they both need TDM resources that are managed by the CSP Matrix Series 3 Card. The IP Network Interface Series 2 card, like the T-ONE/E-ONE line cards, uses spans and channels. The IP Network Interface Series 2 card VoIP channels are divided into multiples of 32-channel spans. Each VoIP module as well as the entire board divides evenly into 32-channel spans.

The IP Network Interface Series 2 card Pulse Coded Modulation (PCM) Companding law is configured for μ-Law, the same as the T-ONE line card. While in Normal Mode, the CSP Matrix Series 3 Card converts between μ-Law and A-Law for each channel.

Span/Channel Relationship

Spans and channels are assigned to the IP Network Interface Series 2 card, not to a specific VoIP endpoint (module). The VoIP channels are a pooled resource that can be associated with any span/channel. For example, span 1, channel 1 can be mapped to a channel on module 1; span 1, channel 2 can be mapped to a channel on module 2. This allows the host to allocate a span/channel according to system requirements.

For the case where all of the physical timeslots are not available, the host can decide how best to utilize the ones that are. The disadvantage is that the host may exhaust a module's resource but still have available

spans and channels. The other disadvantage is during a module failure. If a module or DSP on the module fails, active connections purge, but spans and channels cannot be taken out of service.

Packet Resources

The IP Network Interface Series 2 card has three external 100Base-T (100 Mbps) Ethernet ports located on its Multi-Function Media I/O card. The intent is for the host to trunk these ports together to achieve an effective 300 Mbps pipe. If trunking is not used, only one of the three ports can be used at a time (any port can be used). Trunking (link aggregation) the ports also provides a transparent link failure protection. If one of the links fails, the traffic is automatically diverted to another link.

The IP Network Interface Series 2 card does not support Ethernet Link Redundancy. Instead, it automatically trunks all active Ethernet ports.

It is recommended that the IP Network Interface Series 2 card packet interface be on a different subnet from the one used to manage the CSP. Its main traffic is RTP, RTCP, and T.38 packets.

Important! Ethernet link failures do not cause RTP connections to be dropped. An alarm is generated to the host and the host can purge the connections. The same occurs for the removal of the Multi-Function Media I/O card. An alarm is generated to the host and the host can purge the connections.

CAUTION

Bandwidth on the External Links is not monitored. If you have one link available and you are generating over 100 Mbps of RTP data, no alarms are generated and data is not throttled. The only indication is the degradation of voice quality.

Auto-Negotiate Mode vs. Full-Duplex Mode

You can choose to either have the Ethernet link negotiate the link rate and duplex mode or you can specify 100 Mbps and full duplex mode.

You control the mode with dip switch 4 on the IP Network Interface main board as follows:

Setting	Function
ON (default)	Ethernet Link Auto-Negotiate Mode
OFF	Ethernet Link Force 100 Mbps/Full Duplex Mode

Follow the steps below to configure Full-Duplex Mode:

1. Remove the IPN-2 card and toggle S1002 dipswitch 4 to the OFF position.
2. Re-insert card and allow the software to load.
3. Allow the card to reconfigure. During configuration, the CSP sends the host the *Card Status Report* (0x00A6) with the updated dip switch settings. Refer to this message in the *API Reference*.

VoIP Services

Overview

There are six VoIP services provided by a CSP equipped with IP Network Interface Series 2 cards.

1. VoIP_Report
2. VoIP_Configure
3. VoIP_Query
4. VoIP_Allocate
5. VoIP_Update
6. VoIP_Deallocate

From the IP Network Interface Series 2 card perspective, the external host is transparent. The host can be a SwitchKit, Softswitch, third party, or other. The six VoIP services provide configuration and call control using API messages. The following is a list of messages used by the host to use the above VoIP services.

> **Important!** The API messages in BOLD have been impacted by the integration of IP Network Interface Series 2 card.

VoIP_Report

VoIP_ Reports are asynchronous events generated by the CSP. These reports include the following API messages:

- *Card Status Report* (0x00A6)
- *Alarm* (0x00B9)
- *Channel Released* (0x0049)
- *Channel Released w/ Data* (0x0069)
- *DS0 Status Change* (0x0042)
- *PPL Event Indication* (0x0043)

VoIP_Configure

VoIP_Configure service allows the host to configure OAM&P related parameters. These configure requests include the following API messages:

- *Resource Attribute Configure* (0x00E3)
- *IP Address Configure* (0x00E7)
- *Assign Logical Span ID* (0x00A8)
- *Reset Configuration* (0x000B)
- *Service State Configure* (0x000A)
- *Tag Configuration* (0x00D0)

- *Time Set* (0x00B5)
- *Answer Supervision Mode Configure* (0x00BB)
- *Distant End Release Mode Configure* (0x00B8)
- *Local End Release Mode Configure* (0x0021)
- *PPL Configure* (0x00D7)
- *PPL Audit Configure* (0x00DC)

VoIP_Query

VoIP_Query service allows the host to query various statistics and information about the IP Network Interface Series 2 card. These query requests include the following API messages:

- *Card Status Query* (0x0083)
- *Resource Attribute Query* (0x00E4)
- *Fault Log Query* (0x0086)
- *IP Address Query* (0x00E6)
- *ARP Cache Report* (0x00FB)
- *ARP Cache Query* (0x00FC)
- *PPL Audit Query* (0x00DD)
- *PPL Data Query* (0x00DE)
- *PPL Protocol Query* (0x00DF)

VoIP_Allocate

VoIP_Allocate service is indirectly initiated by the host by using the following two API messages:

- *Outseize Control* (0x002C)
- *Route Control* (0x00E8)

Once a VoIP channel has been established, it becomes like any other channel in the system (T1/E1/DS3),the host can connect it to any other channel in the CSP. All of the connect messages are supported.

- *Connect* (0x0000)

VoIP_Update

Once a VoIP connection is established, the host has the ability to change some of the call attributes on the fly. This is done using the following API message:

- *Resource Attribute Configure* (0x00E3)

VoIP_Deallocate

De-allocating a VoIP channel is the same as for any other channel in the CSP. The host has the ability to break down the connection using the following API messages:

- *Release Channel* (0x0008)
- *Release Channel with Data* (0x0036)

VoIP Resource Profiles

Overview

The IP Network Interface Series 2 card VoIP capabilities are defined by the resource profiles assigned to each IP endpoint (module). Based on the terminal capabilities of the profile, the host decides how each VoIP channel can best be provisioned.

The IP Network Interface Series 2 card has the ability to convert voice data between a circuit-and packet-switched network. Each endpoint can be configured and managed independently from one another except, they share from a common span/channel pool.

Assigning a VoIP Resource Profile

The IP Network Interface Series 2 card supports multiple profiles and the ability to configure different modules with different profiles. While this provides flexibility, it also adds complexity. The host can configure a VoIP module's Resource Profile using the *Resource Attribute Configure* (0x00E3) message. Currently, this API supports the configuration of three entities:

1. Updating the default VoIP attributes of a particular VoIP module.
2. Updating the current VoIP attributes of an active VoIP channel.
3. Updating the IP Network Interface Series 2 card Gateway Mode.

Updating Default Attributes of a VoIP Module/Updating Current Attributes of a VoIP Channel

The *Resource Attribute Configure* (0x00E3) message is addressed to the slot of the IP Network Interface Series 2 card using the Slot Address Type (0x01) AIB. The first TLV after the AIB defines which of the entities are getting configured.

For updating entities 1 and 2 above, an Address Element (0x0009) TLV is used to specify which attributes are to be updated, the default (1) or an active channel (2). The IP Address (0x3E) AIB is used to specify which VoIP module default attributes are to be updated. The Expanded Span/Channel (0x0D) AIB is used to specify which active channel attributes are to be updated. If the Address Element (0x0009) TLV is not provided, the resource attribute will be applied to the IP Network Interface Series 2 card, rather than to a VoIP module or VoIP channel.

Updating IP Network Interface Series 2 Card Gateway Mode

For updating entity 3 above, the IP Network Interface Series 2 card Gateway Mode, the Gateway Mode TLV (0x01D0) must be present without an Address Element TLV (0x0009). The absence of the Address Element TLV indicates that the resource attribute will be applied to the IP Network Interface Series 2 card itself. Since this causes the TDM resources provisioned on the CSP to be changed, a IP

Network Interface Series 2 card reset is necessary and performed automatically. Once the IP Network Interface Series 2 card finishes resetting and comes into service, the *Resource Attribute Configure* (0x00E3) message is acknowledged.

> **Important!** Changing the Gateway Mode causes the entire IP Network Interface Series 2 card host configuration to be reset.

Updating a VoIP Module Resource Profile

The Resource Profile Assign (0x01E5) TLV has been created to configure a module's resource profile. Similar to changing the Gateway Mode, changing a VoIP module resource profile causes its TDM resources to change, causing the IP Network Interface Series 2 card to reset, so that new resources can be allocated.

> **Important!** Changing the resource profile on any module causes the IP Network Interface Series 2 card host configuration to be reset.

The Resource Profile Assign (0x01E5) TLV addresses the VoIP module by its module number. Since updating a VoIP module's profile causes the IP Network Interface Series 2 card to reset, all of the modules can be reset in a single message.

The host has the flexibility to send one VoIP Resource Profile Assign TLV to assign the same resource profile to all modules or send multiple TLVs to individual modules. If any TLV is invalid, the message is negatively acknowledged (NACKED) and no action is taken. The Resource Profile Assign (0x01E5) TLV format is as follows:

Table 3-4 VoIP Resource Profile Configure TLV

Byte	Description
0, 1	Tag: VoIP Resource Profile Assign (0x01E5)
2, 3	Length: 0x0003
4	Value: Data[0] Module Number 0x00 - Module 1 0x01 - Module 2 0x02 - Module 3 (VDAC-ONE) 0x03 – Module 4 (VDAC-ONE) 0xFF - All Populated VoIP Modules
5	Data[1] Profile Number 0x00 - Profile 0 (VDAC-ONE) 0x01 - Profile 1 (IP Network Interface Series 2 – G.711 Only) 0x02 - Profile 2 (IP Network Interface Series 2 – VDAC-ONE Comp.)
6	Data[2] Force Flag 0x00 - Update only if Spans have not been configured. 0xFF - Update Unconditionally.

Terminal Capability Query

The VoIP Terminal Capabilities (0x01EA) TLV has been created to query an IP endpoint's terminal capabilities. The report returned is based on the resource profile assigned to the module.

The following information is provided:

- VoIP Resource Profile ID
- Maximum EC Tail Length
- Maximum Jitter Buffer Size
- Silence Suppression Support Silence Suppression (sometimes referred to as Voice Activity Detection, VAD)
- Fax Relay Support
- RFC 2833 Digit Relay Support
- RFC 2198 RTP Redundancy Support
- List of codecs supported along with their base and maximum packet rate

Below is the format of the VoIP Terminal Capabilities (0x01EA) TLV, which is queried from the *Resource Attribute Query* (0x00E4) message:

Important! The VoIP Terminal Capabilities can be queried by addressing either the IP Address AIB or Extended Span/Channel AIB embedded within the Address Element TLV.

Table 3-5 VoIP Terminal Capabilities TLV (0x01EA)

Byte	Description
0,1	Tag: VoIP Terminal Capabilities (0x01EA)
2, 3	Length: Variable
4	Data[0] VoIP Resource Profile ID 0x00 - VDAC-ONE 0x01 - IP Network Interface Series 2, G.711 Only 0x02 - IP Network Interface Series 2, LBR + T.38
5	Data[1] Echo Canceler Tail Length (in ms)
6, 7	Data[2] MSB Jitter Buffer Length (in ms) Data[3] LSBB Jitter Buffer Length (in ms)
8	Data[4] Silence Suppression Support (bitmap) 0x00 - No Support 0x01 -Generic VDAC Support 0x02 - G.723.1 Annex A Support 0x04 - G.729 Annex B Support
9	Data[5] Fax Relay Supported (bitmap) 0x00 – No Support 0x01 – Support using T.38
11	Data[6] Modem Relay Supported (bitmap) (Not Supported) 0x00 – No Support 0x01 – Support for V.32 0x02 – Support for V.34
12	Data[7] Digit Relay Support (bitmap) 0x00 – No Support 0x01 – Support using RFC 2833
13	Data[8] RTP Redundancy Support (bitmap) 0x00 – No Support 0x01 – Support using RFC 2198
14	Data[9] Number of Coders Supported (n)

15	Vocoder[1] - Payload Type
16	Vocoder[1] - Basic Packet Rate (in ms)
17	Vocoder[1] - Max Packet Rate (multiples of Basic Rate)
…	…
…	Vocoder[n] - Payload Type
…	Vocoder[n] - Basic Packet Rate (in ms)
14 + n*3	Vocoder[n] - Max Packet Rate (multiples of Basic Rate)

Packet Voice Service

On the VDAC-ONE card, the codecs, G.711, G.726, and G.729 have a basic packet rate of 20 milliseconds. On the IP Network Interface Series 2 card, G.711 and G.726 are 5 milliseconds and G.729 is 10 milliseconds.

IP Network Interface Series 2 Voice Coder Packet Rate Information

For the IP Network Interface Series 2 card there is a coupling between the Payload Type (codec) and Payload Size (packet rate). Unlike VDAC-ONE card, the base packet rate (sampling) of a codec is different for different codecs. This implies that a Payload Size, which is defined in multiples of the base packet rate are specific to the Payload Type. The table below shows the relationship between the Payload Type and Payload Size on a IP Network Interface Series 2 card. It also shows a list of all Voice Coders supported by the IP Network Interface Series 2 card.

Table 3-6 IP Network Interface Series 2 Voice Coder Packet Rate Information

TLV Payload Type Data Value (decimal)	RTP Payload Type	Basic Packet Rate (ms)	Default Packet Rate (sampling) (ms)	Max Packet Rate (ms)
0	G.711 A-Law (64 Kbps)	5 ms	4x - 20 ms	6x - 30 ms
1	G.711 μ-Law (64 Kbps)	5 ms	4x - 20 ms	6x - 30 ms
2	G.726 (16 Kbps)	5 ms	4x - 20 ms	9x - 45 ms
3	G.726 (24 Kbps)	5 ms	4x - 20 ms	9x - 45 ms
4	G.726 (32 Kbps)	5 ms	4x - 20 ms	9x - 45 ms
5	G.726 (40 Kbps)	5 ms	4x - 20 ms	9x - 45 ms
15	G.723.1 (5.3 Kbps)	30 ms	1x - 30 ms	4x - 120 ms
16	G.723.1 (6.3 Kbps)	30 ms	1x - 30 ms	4x - 120 ms
17	G.729 (8 Kbps)	10 ms	2x - 20 ms	15x - 150 ms

IP Network Interface Series 2 and VDAC-ONE Basic Packet and Maximum Packet Rate Differences

Note that the Basic Packet Rate and the Maximum Packet Rate are different between the IP Network Interface Series 2 and VDAC-ONE cards.

These differences will cause problems with Call Control. The host needs to know which IP Network Interface Series 2 card the *Outseize Control* and *Route Control* messages are going to and adjust the packet size accordingly. To compensate for this, the host or upper layer protocol needs to query an IP endpoint's terminal capabilities. The information returned in the query will provide enough information for a user to build negotiation tables and correctly interface with the IP endpoint.

VoIP Resource Profile Terminal Capabilities

Overview VoIP Resource Profiles are categorized by their terminal capabilities. The following tables describe the VoIP terminal capabilities for each IP Network Interface Series 2 card resource profile, including the VDAC-ONE card.

Table 3-7 VDAC-ONE Resource Profile Terminal Capabilities

Profile Number	Profile Name	Max Spans	Max Channels	Max EC Tail	Max JB Delay
0	**VDAC-ONE**	**1.25**	**40**	**25ms**	**300ms**
Codecs Supported				Fax Relay Supported	Modem Relay Supported
(0) G.711 A-Law, (1) G.711 μ-Law, (2-5)G.726 16, 24, 32, 40Kpbs , (6-14) G.727 16, 16-24, 24, 16-32, 24-32, 32, 16-40, 24-40, 32-40, 40Kpbs, (15) G.723.1 5.3Kbps, (16) G.723.1 6.3Kbps, (17) G.729				Yes	No

Attribute Name (TLV Tag)	Length (bytes)	Default Value	Valid Values	Defaults Configurable	Run-time Configurable
(0x100) RTP Payload Type	1	(1) G.711 μ-Law	See Codecs Supported	Yes	No
(0x101) RTP Payload Size	1	(1) 1x	(1) 1x - (8) 8x	Yes	No
(0x102) RTP Silence Suppression	1	(0) Disabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x103) Echo Cancellation	1	(1) Enabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x1C2) Minimum Jitter Buffer Delay	4	75ms	0 - 150ms, Must be smaller than Maximum JB Delay	Yes	No
(0x1C3) Maximum Jitter Buffer Delay	4	150ms	0 - 300ms, Must be larger than Minimum JB Delay	Yes	No
(0x1C4) Adaptation Rate	1	0 - 0xc	7	Yes	No
(0x1C5) Fax Type	1	(0) Transparent	(0) Transparent, (1) Relay, (2) Bypass	Yes	Before Fax has started
(0x1C7) Bypass Coder Type	1	(1) G.711 μ-Law	See Codecs Supported	Yes	Before Fax has started
(0x1D1) RTP Redundancy	1	(0) No Redundancy	(0) No Redundancy, (1) Level 1 using RFC 2198	Yes	No
(0x1D2) Fax Redundancy	1	(0) No Redundancy	(0) No Redundancy, (1 - 2) Level 1 to Level 2	Yes	Before Fax has started

Attribute Name (TLV Tag)	Length (bytes)	Default Value	Valid Values	Defaults Configurable	Run-time Configurable
(0x1D4) Type Of Service	1	0	0 - 0xFE [Precedence (000 - 111), Delay (0-1), Reliability (0-1), Cost (0-1)]	Yes	Yes
(0x1DB) Connection Mode	1	(0) Tx/Rx	(0) Tx/Rx, (1) Rx Only (2) Tx Only	No	Yes
(0x1DF) UDP Source Port Validate	1	(1) Enabled	0) Disabled, (1) Enabled	Yes	Yes
(0x1E2) RFC 2833 Enable	1	(0) Disabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x1E6) T.38 SRC UDP Port	4	No Default	0 – 0xFFFF	No	Before Fax has started
(0x1E7) T.38 DST UDP Port	4	No Default	0 – 0xFFFF	No	Before Fax has started
(0x1E8) RTCP SRC UDP Port	4	RTP Port + 1	0 – 0xFFFF	No	Yes
(0x1E9) RTCP DST UDP Port	4	RTP Port + 1	0 – 0xFFFF	No	Yes
(0x1EB) RTP Timer Timeout	4	(0) Disabled	0 – 0xFFFF (x 10 ms)	Yes	No
(0x1EC) Media Inactivity-Detection Timeout	4	(0) Disabled	0 – 0xFFFF (x 10 ms)	Yes	No
(0x2794) DST IP ADDR	4	No Default	IP Address – any valid IP Address	No	Yes
(0x2795) DST RTP ADDR	4	No Default	UDP Port - Any valid IP Port	No	Yes

Table 3-8 IP Network Interface Series 2 Resource Profile 1

Profile Number	Profile Name	Max Spans	Max Channels	Max EC Tail	Max JB Delay
1	**IP Network Interface Series 2: G.711 Only**	**16**	**512**	**64ms**	**300ms**
Codecs Supported				Fax Relay Supported	Modem Relay Supported
(0) G.711 A-Law, (1) G.711 μ-Law				No	No

(TLV Tag) Attribute Name	Length (bytes)	Default Value	Valid Values	Defaults Configurable	Run-time Configurable
(0x100) RTP Payload Type	1	(1) G.711 μ-Law	See Codecs Supported	Yes	Yes
(0x101) RTP Payload Size	1	Coder Dependent	Coder Dependent. See Table 3-7	Yes	Yes
(0x102) RTP Silence Suppression	1	(0) Disabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x103) Echo Cancellation	1	(1) Enabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x1C2) Minimum Jitter Buffer Delay	4	75ms	0 - 150ms, Must be smaller than Maximum JB Delay	Yes	No
(0x1C3) Maximum Jitter Buffer Delay	4	150ms	0 - 300ms, Must be larger than Minimum JB Delay	Yes	No
(0x1C5) Fax Type	1	(0) Transparent	(0) Transparent, (2) Bypass	Yes	No
(0x1C7) Bypass Coder Type	1	(1) G.711 μ-Law	See Codecs Supported	Yes	No
(0x1D1) RTP Redundancy	1	(0) No Redundancy	(0) No Redundancy	Yes	Yes
(0x1D4) Type Of Service	1	0	0 - 0xFE [Precedence (000 - 111), Delay (0-1), Reliability (0-1), Cost (0-1)]	Yes	Yes
(0x1DB) Connection Mode	1	(0) Tx/Rx	(0) Tx/Rx, (1) Rx Only, (2) Tx Only	No	Yes
(0x1DF) UDP Source Port Validate	1	(1) Enabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x1E2) RFC 2833 Enable	1	(0) Disabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x1E6) T.38 SRC UDP Port	4	No Default	0 – 0xFFFF	No	Before Fax has started.

(TLV Tag) Attribute Name	Length (bytes)	Default Value	Valid Values	Defaults Configurable	Run-time Configurable
(0x1E7) T.38 DST UDP Port	4	No Default	0 – 0xFFFF	No	Before Fax has started.
(0x1E8) RTCP SRC UDP Port	4	RTP Port + 1	0 – 0xFFFF	No	Yes
(0x1E9) RTCP DST UDP Port	4	RTP Port + 1	0 – 0xFFFF	No	Yes
(0x1EB) Initial Media Inactivity Detection Timeout	4	(0) Disabled	0 – 0xFFFF (x 10 ms)	Yes	No
(0x1EC) Media InactivityDetection Timeout	4	(0) Disabled	0 – 0xFFFF (x 10 ms)	Yes	Yes
(0x2794) DST IP ADDR	4	No Default	IP Address – any valid IP Address	No	Yes
(0x2795) DST RTP ADDR	4	No Default	UDP Port - Any valid IP Port	No	Yes

Table 3-9 IP Network Interface Series 2 Resource Profile 2 Terminal Capabilities

Profile Number	Profile Name	Max Spans	Max Channels	Max EC Tail	Max JB Delay
2	**IP Network Interface Series 2: VDAC-ONE Compatible**	**8**	**256***	**64ms**	**300ms**
Codecs Supported				Fax Relay Supported	Modem Relay Supported
(0) G.711 A-Law, (1) G.711 μ-Law, (2-5) G.726, 16Kbps, 24Kbps, 32Kbps, 40Kbps, (15-16) G.723.1 5.3Kbps, 6.3Kbps, (17) G.729 A/B				**Yes**	**No**

(TLV Tag) Attribute Name	Length (bytes)	Default Value	Valid Values	Defaults Configurable	Run-time Configurable
(0x100) RTP Payload Type	1	(1) G.711 μ-Law	See Codecs Supported	Yes	Yes
(0x101) RTP Payload Size	1	Coder Dependent	Coder Dependent. See Table 3-7.	Yes	Yes
(0x102) RTP Silence Suppression	1	(0) Disabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x103) Echo Cancellation	1	(1) Enabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x1C2) Minimum Jitter Buffer Delay	4	75ms	0 - 150ms, Must be smaller than Maximum JB Delay	Yes	No
(0x1C3) Maximum Jitter Buffer Delay	4	150ms	0 - 300ms, Must be larger than Minimum JB Delay	Yes	No

(0x1C5) Fax Type	1	(0) Transparent	(0) Transparent, (1) Relay, (2) Bypass	Yes	No
(0x1C7) Bypass Coder Type	1	(1) G.711 μ-Law	See Codecs Supported	Yes	No
(0x1D1) RTP Redundancy	1	(0) No Redundancy	(0) No Redundancy, (1) Level 1 using RFC 2198	Yes	Yes
(0x1D2) Fax Redundancy	1	(0) No Redundancy	(0) No Redundancy, (1 - 3) Level 1 to Level 3	Yes	No
(0x1D4) Type Of Service	1	0	0 - 0xFE [Precedence (000 - 111), Delay (0-1), Reliability (0-1), Cost (0-1)]	Yes	Yes
(0x1DB) Connection Mode	1	(0) Tx/Rx	(0) Tx/Rx, (1) Rx Only, (2) Tx Only	No	Yes
(0x1DF) UDP Source Port Validate	1	(1) Enabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x1E2) RFC 2833 Enable	1	(0) Disabled	(0) Disabled, (1) Enabled	Yes	Yes
(0x1E6) T.38 SRC UDP Port	4	No Default	0 – 0xFFFF	No	Before Fax has started.
(TLV Tag) Attribute Name	**Length (bytes)**	**Default Value**	**Valid Values**	**Defaults Config-urable**	**Run-time Config-urable**
(0x1E7) T.38 DST UDP Port	4	No Default	0 – 0xFFFF	No	Before Fax has started.
(0x1E8) RTCP SRC UDP Port	4	RTP Port + 1	0 – 0xFFFF	No	Yes
(0x1E9) RTCP DST UDP Port	4	RTP Port + 1	0 – 0xFFFF	No	Yes
(0x1EB) Initial Media Inactivity Detection Timeout	4	(0) Disabled	0 – 0xFFFF (x 10 ms)	Yes	No
(0x1EC) Media Inactivity Detection Timeout	4	(0) Disabled	0 – 0xFFFF (x 10 ms)	Yes	Yes
(0x2794) DST IP ADDR	4	No Default	IP Address – any valid IP Address	No	Yes
(0x2795) DST RTP ADDR	4	No Default	UDP Port - Any valid IP Port	No	Yes

* There are a total of 256 channels. However, one channel is used internally for ICMP support leaving 255 channels.

Definitions of Terminal Capabilities

The definitions of the above terminal capabilities are as follows:

Profile Number (ID)

The Profile Number is the system-wide identification of the Resource Profile. ID 0 is for VDAC-ONE, IDs 1-3 are for IP Network Interface Series 2.

Profile Name

The Profile Name identifies the profile.

Max Spans

The number of 32-channel spans the profile is capable of supporting.

Max Channels

The number of VoIP Channels the profile is capable of supporting.

Codecs

The list of Voice codecs supported by this profile. The number prefixed to the codec is the value of the codec used in the PAYLOAD TYPE TLV Attribute.

Max EC Tail

The Echo Canceller tail length supported by the profile.

Max JB Delay

The maximum amount of voice data buffered up to reduce jitter.

FAX Relay Support

Indicates whether T. 38 FAX Relay is supported.

Attribute Name

Indicates the name and TLV tag of configurable VoIP Channel Attributes.

Length

Indicates the number of bytes for the value in the Attribute TLV.

Default Value

Indicates the default value assigned to the Attribute TLV upon. initiating a IP Network Interface Series 2 card cold start.

Valid Values

Indicates the range of valid values associated with the Attribute TLV.

Default Configurable

Indicates whether or not the host can configure the module-wide default for this Attribute TLV. If so, all subsequent channels will use this value as a default.

Run-time Configurable

Indicates whether or not the host can update the Attribute TLV Value on an active channel.

IP Resources Attributes

Overview

You can assign attributes to a VoIP module during initial configuration by using the *Resource Attribute Configure* message.

You can assign attributes during call set-up by using the *Route Control* or *Outseize Control* messages with Address Information Blocks (AIBs).

You can dynamically assign some attributes to a channel after call setup by using the *Resource Attribute Configure* message with the span/channel AIB.

Synchronizing VoIP Media Parameters

If you change VoIP resource configuration with the *Resource Attribute Configure* message, the local copy of the configuration maintained by SIP and H.323 software remains unchanged. You must resynchronize the configurations as follows:

- CSA users select the Resynchronization option.
- Non-CSA users send the Resynchronize VoIP Media Parameters TLV (0x0284) in the *VoIP Protocol Configure* (0x00EE) message. Refer to the *0x0284 Resynchronize VoIP Media Parameter* in the *API Reference* for the details of this TLV.

IP Resource Attributes Summary Table

The following tables show the default values for IP Resource Attributes and indicate if you can configure them dynamically on an active connection.

Table 3-10 General Attributes

TLV Tag and Name	Default Setting	Dynamic Configuration Supported
0x01D0 Gateway Mode	Non Port Consuming	No
0x01D4 Type of Service	See TLV Format	Yes
0x01DF UDP Source Port Validate	Enabled	Yes
0x01EC Media Inactivity Detection Timer	Disabled	No

Table 3-11 Fax Attributes

TLV Tag and Name	Default Setting	Dynamic Configuration Supported
0x01C5 Fax Type Enable	Disabled	No
0x01C7 Fax Bypass Coder Type	Disabled	No
0x01D2 Fax Payload Redundancy	Disabled	No
0x01E1 Fax Compatibility Mode	Backward compatible mode	No
0x01E2 RFC 2833 Enable	Disabled	Yes
0x01E6 Source T.38 Port	No default	Yes
0x01E7 Destination T.38 Port	No default	Yes
0x01E8 Source RTCP Port	One offset from source RTP port	Yes
0x01E9 Destination RTCP Port	One offset from destination RTCP port	Yes

Table 3-12 Voice Attributes

TLV Tag and Name	Default Setting	Dynamic Configuration Supported
0x0100 RTP Payload Type	G.711µ-Law	Yes
0x0101 RTP Payload Size	1X	Yes
0x0102 RTP Silence Suppression	Disabled	Yes
0x0103 RTP Echo Cancellation	Enabled	Yes
0x01C2 Minimum Jitter Buffer Delay	75 ms	No
0x01C3 Maximum Jitter Buffer Delay	150 ms	No
0x01C4 Adaptation Rate	7	No
0x01D1 RTP Payload Redundancy	Disabled	No
0x01DB Connection Mode	See the TLV format.	Yes
0x01E2 RFC 2833 Enable	Disabled	Yes
0x01E8 Source RTCP Port	One offset from destination RTP port	Yes
0x01E9 Destination RTCP Port	One offset from destination RTP port	Yes

General IP Attributes

Overview

The following are general IP resource attributes that apply to voice and fax connections:

IP Type Of Service

The IP Network Interface Series 2 card supports the *IPv4 Type of Service* field in the IP packet header. This field is used to instruct packet switches and routers in an IP-based network to control packets sent from a IP Network Interface Series 2 card. You can set options in this field to indicate that the packets should be given preferential treatment on a Class of Service basis.

> **Important!** Most IP switches and routers do not currently support different levels of service.

You can configure the following IP Type of Service options for outgoing packets:

- Cost
- Reliability
- Throughput
- Delay
- Precedence

Default: 0x00 (No preferential options set)

Use TLV: 0x01D4 Type of Service

UDP Source Port Validation

If the User Datagram Protocol (UDP) Source Port Validation is enabled, the IP Network Interface Series 2 card verifies that the UDP source port of the incoming packets matches the destination port of the originating outgoing packets. If not, the packets will be transparently dropped.

Default: Enabled

Use TLV: 0x01DF Source Port Validate

Media Inactivity Detection (MID) Timer

The host can enable a Media Inactivity Detection (MID) Timer on a per channel basis to monitor the activity of the incoming media stream (for example, valid UDP packets).

When the MID Timer is enabled on a VoIP Channel, every valid UDP packet received on this channel refreshes the timer. If the timer expires and no valid packets have been received, a *PPL Event Indication* (0x0043) message is generated.

The host has the ability to configure two timeout values for the MID Timer, an Initial Media Inactivity (IMID) Timeout Value TLV (0x01EB) and a MID Timeout Value TLV (0x01EC). Both of these values are configured in units of 10 milliseconds.

When the channel is enabled, the MID Timer is loaded with the IMID Timeout Value. If the timer expires before the first valid packet is received, a *PPL Event Indication* (0x0043) message is generated with PPL Event 0x07, Received Initial Media Inactivity Timeout.

The receipt of a valid UDP packet causes the MID Timer to be reloaded with the MID Timeout Value. If a valid UDP packet is not received before the timer expires, a *PPL Event Indication* (0x0043) message is generated with PPL Event 0x08, Received Media Inactivity Timeout.

> **Important!** In order to maintain backwards compatibility with the VDAC-ONE card's implementation of the MID Timer, the timer is loaded initially with the MID Timeout Value when the IMID Timeout Value is disabled (0x0000). Under this circumstance, the MID Timer expiring results in a *PPL Event Indication* being generated with a PPL Event of 0x08, Received Media Inactivity Timeout.

Once the MID Timer has expired, the receipt of a valid UDP packet does NOT automatically re-enable the timer. The host must manually re-enable the timer by updating the active channel's MID Timeout Value TLV (0x01EC). This is done using the *Resource Attribute Configure* (0x00E3) message. Upon receipt of this message, the IP Network Interface Series 2 card will re-enable the timer and load the supplied MID Timeout Value.

> **Important!** If the host dynamically updates the MID Timeout Value while the MID Timer is enabled, the new value will not be loaded until after the next valid UDP packet is received.

Both the IMID and MID Timeout Values default to 0x0000, disabling the MID Timer. Enabling the IMID and disabling the MID cause the timer to automatically be disabled after the first packet is received. Enabling the MID and disabling the IMID causes the MID to be loaded initially.

Initial Media Inactivity Timeout (0x01EB) TLV

Used in:
Resource Attribute Configure message
Resource Attribute Query message
Route Control message
Outseize message

Byte	Description	
0, 1	Tag	0x01EB
2, 3	Length	0x0004
4, 5	Value[0, 1]	0x0000
6, 7	Media Inactivity Detection Timeout (2 Bytes) 0x0000 - 0xFFFF (in 10 ms increments) 0x0000 is the default and indicates that the IMID detection is disabled. The Timer is loaded instead with the Media Inactivity Detection Timeout Value. Any other value causes the MID Timer to be initialized with this value.	

Important! The IMID Timeout TLV cannot be dynamically updated on an active channel. The default IMID Timeout TLV can be configured on a per module basis.

Media Inactivity Detection Timer (0x01EC) TLV

Used in:
Resource Attribute Configure message
Resource Attribute Query message
Route Control message
Outseize message

Byte	Description	
0, 1	Tag	0x01EC
2, 3	Length	0x0004
4, 5	Value[0,1]	0x0000
6, 7	RTP Initial Media Inactivity Timeout Value (2 Byte) 0x0000 - 0xFFFF (in 10 ms increments)) 0x0000 is the default and indicates that the MID Timer is disabled after the first packet is received.Setting both the IMID and MID to 0x0000 disables the MID Timer.	

Important! The MID Timeout TLV can be dynamically updated on an active channel. The default MID Timeout TLV can be configured on a per module basis.

Examples of MID Timer Events

The examples below describe different MID Timer scenarios.

1. A call is started with an IMID Timeout of 0 and a MID Timeout of 5 seconds. The Timer is loaded with 5 seconds. The receipt of a valid packet reloads the timer with 5 seconds. If the timer ever expires, a MID Expired PPL Event of 0x08 is generated. (This example implies backwards compatibility with the VDAC-ONE card.)
2. A call is started with an IMID Timeout of 20 seconds and a MID Timeout of 5 seconds. The Timer is loaded with 20 seconds. If after 20 seconds, a valid packet is not received, an IMID Expired PPL Event of 0x07 is generated and the timer is disabled (for example a packet arriving at 21 seconds will not reload the timer). If a valid packet is received before 20 seconds, the timer is reloaded with 5 seconds. If 5 seconds expire, a MID Expired PPL Event of 0x08 is generated.

   Important! Even though the MID Timeout is set for 5 seconds, the IMID Timeout overrides it, until the first packet is received.

3. A call is started with an IMID Timeout of 5 seconds and a MID Timeout of 20 seconds. The Timer is loaded with 5 seconds. If after 5 seconds, a valid packet is not received, an INIT MID Expired PPL Event of 0x07 is generated and the timer is disabled. If after 5 seconds, a valid packet is received, the timer is NOT reloaded. If a valid packet is received before 5 seconds, the timer is reloaded with 20 seconds. If 20 seconds expire, a MID Expired PPL Event of 0x08 is generated.

   Important! Even though the MID Timeout is set for 20 seconds, the IMID Timeout overrides it until the first packet is received.

4. A call is started with an IMID Timeout of 0 seconds and a MID Timeout of 0 seconds. The Timer is disabled.

5. Call is starting with both timers set to 0 (disabled). During the call, the MID Timeout value is updated to 20 seconds. The timer is enabled and loaded with 20 seconds.

CNAME Support for VDAC/ IPN-2

The CNAME field is supported for the IPN-2 and VDAC-ONE cards. The CNAME field will automatically be populated at channel configuration time and included in the RTCP Sender Report. The CNAME field is not configurable.

The data used for the CNAME field is comprised solely of the ASCII character formats for the VDAC/IP module's source IP address (15 bytes) and source UDP (RTP) port number (5 bytes) for a specific RTP voice session, separated by an ASCII "colon" (":") character (1 byte).

You can use an Ethernet trace program to capture and analyze RTCP packets.

Voice Attributes

Introduction

You can configure the following voice attributes with the *Resource Attribute Configure* (0x00E3) message:

RTP Payload Type

The RTP payload type indicates the algorithm used for compressing the data or payload portion of the packet.

Default: G.711µ-Law

Use TLV: 0x0100 RTP Payload Type

RTP Payload Size

RTP payload sizes can be changed in multiples of the specified RTP Payload Type's base packetization rate. On the IP Network Interface Series 2 card, the RTP payload size is tightly coupled with RTP payload type. Changing the default RTP payload type also changes the default RTP payload size to its applicable value (unless a new default RTP payload size is also specified). Using the non-default RTP Payload Type during call establishment results in the default RTP Payload Size not being used. If the RTP Payload Size is not provided, the default for the specific payload type is used.

For the IP Network Interface Series 2 card, refer to Table 3-7 for each payload type's base packetization rate (for example,1 x payload size), its default payload size and maximum payload size.

Default: 4x (G.711µ-Law 20ms)

Use TLV: 0x0101 RTP Payload Size

Silence Suppression

During a normal voice conversation, much of the time is wasted on silence from one or both ends. Ethernet bandwidth can be conserved if, during these periods of silence, RTP packets are sent with silence-encoded, compressed payloads.

You can enable or disable Silence Suppression on an established connection.

Default: Disabled

TLV: 0x0102 RTP Silence Suppression
(Do not send to an active call that is in fax/modem mode.)

The IPN-2 card supports silence suppression as follows:

A voice activity detection (VAD) algorithm determines which portions of the input signal contain active speech. At the beginning of silence periods (the end of active periods), a special silence insertion descriptor (SID) packet is generated that describes the background noise. The SID packets are generated at the onset of the silence period and whenever the characteristics of the background noise change. The speech decoder that receives the SID packet uses a comfort noise generation (CNG) algorithm to reproduce the background noise from the information in the packet and possibly information contained in past active voice packets. The average bit rate required for voice transmission is lowered considerably when using silence suppression.

Echo Cancellation

In compliance with ITU G.168, this feature eliminates echo introduced by impedance mismatched hybrids. You can enable and disable Echo Cancellation on an established connection.

Default: Enabled

TLV: 0x0103 RTP Echo Cancellation

RTP Payload Redundancy

You can set the redundancy level for RTP using the RTP Payload Redundancy TLV. See RFC 2198.

Default: Disabled

TLV: 0x01D1 RTP Payload Redundancy

Connection Mode

During call setup, the Connection Mode is automatically determined according to the presence of destination and source addresses in the *Route Control* (0x00E8) or *Outseize Control* (0x002C) message.

Source Address Only - Receive Only

Source and Destination Address - Transmit and Receive

To change the Connection Mode after a call is established using *Resource Attribute Configure* (0x00E3) message, send the Connection Mode (0x01DB) TLV with the appropriate mode set.

In hold mode, no voice is transmitted or received.

Example: After a two-way call has been established to a Call Center, the caller is put on hold and connected to music. The Connection Mode of the Call Center line is changed to Transmit Only. If callers enable the mute feature on their phone, the Connection Mode on their line is changed to Receive Only.

Default

There is no default Connection mode. If you query the Module Connection Mode, it is reported as 0xFF. This is a generic value and does not indicate a specific mode. You cannot set the Connection Mode to this value.

TLV: 0x01DB Connection Mode

> **Important!** Source Address (IP/RTP Port) is always required in *Route Control* or *Outseize Control* messages.

RFC 2833 Enable

This TLV allows DTMF signals to be relayed within the VoIP media stream, using a special RFC 2833-compliant packet format. Low bit-rate audio codecs (such as G.729 or G.723.1) can compromise the signal integrity of DTMF digits (and other telephony tones and signals), causing inaccurate detection and recognition of the DTMF digits on the recipient side. When this feature is enabled, the detected incoming DTMF digits (PSTN side) are removed from the audio stream by the IP Network Interface Series 2 card. The information for the detected and removed DTMF digits is embedded within the RTP stream, using a special RFC 2833-compliant packet format. The IP Network Interface Series 2 card on the receiving side decodes this special RTP payload carrying DTMF information, and regenerates the DTMF digits toward the receiving PSTN side. This TLV cannot be set for an active call. It can be configured only before or during the setup of the IP Network Interface Series 2 card call.

Because the VDAC-ONE card allows up to 30 milliseconds of a digit through before blocking it, the distant end might hear a very short audible sound.

> **Important!** The CSP does NOT support RFC 2833 via H.323 signaling.

Default: Disabled

TLV: 0x01E2 RFC 2833 Enable

RTP Suppression for H.245 Signaling Tones

The EXS API has been updated to allow you to configure how DTMF digits are transmitted to the IP network.

Important! This feature is for Clear Channel calls only.

This feature extends the RFC2833 Enable (0x01E2) TLV as follows:

- The new value 0x02 removes the digit from the voice stream and drops the digit.
- If you are using the VDAC card, this TLV can be configured only before or during the call establishment. It cannot be set during an active call. If you are using the IP Networking Series 2 card, this TLV can be configure on the fly while the call is active.

Source RTCP Port

This TLV is used to specify the local RTCP port number. If this port is not specified by the host during the IP Network Interface Series 2 card call setup, the source RTCP port is set one higher than the source RTP port, per RFC 1889. This TLV can be set during an active call.

Default: No default

TLV: 0x01E8 Source RTCP Port

Destination RTCP Port

This TLV is used to specify the remote RTCP port number. If this port is not specified by the host during the IP Network Interface Series 2 card call setup, the destination RTCP port is set to one higher than the destination RTP port, per RFC 1889. This TLV can be set during an active call.

Default: No default

TLV: 0x01E9 Destination RTCP Port

Minimum Jitter Buffer Delay

You configure the Minimum Jitter Buffer Delay by using the Minimum Jitter Buffer Delay TLV in the *Resource Attribute Configure* (0x00E3) message. The range of valid values is 0 - 150 milliseconds.

Default: 75 milliseconds

TLV: 0x1C2 Minimum Jitter Buffer Delay

Important! You cannot configure the Minimum Jitter Buffer Delay on an established connection.

Maximum Jitter Buffer Delay

You configure the Maximum Jitter Buffer Delay by using the Maximum Jitter Buffer Delay TLV in the *Resource Attribute Configure* (0x00E3) message. The range of valid values is 150-300 milliseconds.

Default: 150 milliseconds

TLV: 0x1C3 Maximum Jitter Buffer Delay

Important! You cannot configure the Maximum Jitter Buffer Delay on an established connection.

RFC 2198 (RTP Redundancy) Dynamic Payload Negotiation

The VDAC-ONE card and the IP Network Interface Series 2 card allows RFC 2198 dynamic payload negotiation for clear channel calls only.

You first must enable RFC 2198 with the RTP Payload Redundancy TLV (0x01D1) in the *Resource Attribute Configuration* (0x00E3) message.

You can configure the default value at configuration time or change this value per call as follows:

- To configure the default RFC 2198 Payload Type, use the RFC 2198 Dynamic Payload Type TLV (0x01F2) in the *Resource Attribute Configuration* (0x00E3) message.
- To change the default RFC 2198 payload type per call, use the TLV above within the Generic PPL ICB (0x1E) in the *Route Control* (0x00E8) or *Outseize Control* (0x002C) messages.

Default: 104

TLV: 0x01F2

Fax Attributes

Overview

The IP Network Interface Series 2 card provides real-time fax relay, compliant with ITU T.38, over IP networks. This fax capability fully supports the ITU T.30 for Group 3 facsimile transmission protocol, with speeds of up to 14.4 Kbps. To remedy network packet loss and latency issues, the IP Network Interface Series 2 card employs fax spoofing techniques and packet redundancy schemes.

With the IP Network Interface Series 2 card, in the Fax Relay mode, faxes can be sent over IP using as much as 40 to 50 percent less bandwidth than the full 64 Kbps used on a traditional circuit-switched network.

The IP Network Interface Series 2 card also provides a Fax Bypass mode for real-time fax transmission using high-speed coders. If a matrix switchover occurs during fax transmission, the transmission fails.

> **Important!** The IP Network Interface Series 2 card does not support the V.33 ITU Standard.

Configurable Fax Attributes

You can configure the following fax attributes.

Fax Type Enable

Fax Relay

The IP Network Interface Series 2 cards behavior depends upon how it is configured for detecting fax transmissions. When the host has configured a channel for Fax Relay modes and the IP Network Interface Series 2 card detects that a call is not a voice call, the codec switches from the Voice Coder mode to Answer Tone mode and then to Fax Relay mode. The packets are sent to the network as fax relay packets that comply with T.38. When the fax transmission ends, the channel reverts to Voice Coder mode.

Fax Bypass

When the host has configured a channel for bypass operation and a fax transmission is detected, the codec automatically switches from the current voice coder to the rate of the configured Coder Type. When the fax transmission ends, the channel reverts to Voice Coder mode.

Default: Disabled

TLV: 0x01C5 Fax Type Enable

0x00 Disable

0x01 Fax Relay

0x02 Fax Bypass

If the fax type is set to disabled, then the fax goes through as a voice call. Therefore, the fax transmission might not be successful depending on the codecs used.

Fax Bypass Coder Type

This attribute selects the codec to be used in a fax bypass scenario. Dialogic strongly advises selecting a data rate greater than 32 kbps.

Default: G.711 µLaw

TLV: 0x01C7 Fax Bypass Coder Type

Fax Payload Redundancy

Payload redundancy is a type of protection against network packet loss. By enabling redundancy, you can configure the CSP to send redundant packets to the network. For example, when you set the redundancy level to 2, the original payload is followed by two duplicate payload. Fax redundancy is supported in relay mode only.

Default: Disabled

TLV: 0x01D2 Fax Payload Redundancy

Source T.38 Port

This TLV is used to specify the local T.38 port number for fax relay. There is no default T.38 port value for this TLV. The host must select this port number as early as possible to ensure proper fax relay operation. If this port number is not specified in time, the fax relay fails. This TLV can be set during an active call.

Default: No default

TLV: 0x01E6 Source T.38 Port

Destination T.38 Port

This TLV is used to specify the remote T.38 port number for fax relay. There is no default T.38 port value for this TLV. The host must select this port number as early as possible to ensure proper fax relay operation. If this port number is not specified in time, the fax relay will fail. This TLV can be set during an active call.

Default: No default

TLV: 0x01E6 Destination T.38 Port

IP Connection Management

Overview

To establish an IP connection, you use the *Route Control* (0x00E8) or *Outseize Control* (0x002C) message.

Establishing a Two-way Connection

To establish a two-way connection, use the *Route Control* message with the following TLVs:

- Source IP Address
- Source RTP Port Number
- Destination IP Address
- Destination RTP Port Number
- Connection Mode: two-way

Figure 3-2 IP Network Interface Series 2 Card Connection - Two-Way

One-Way Connection

There may be situations where a one-way IP connection is needed. For example, if Gateway A wants to reserve an IP address and an RTP port, it may establish one side of an IP connection with a one-way listen only. To establish the other side of the IP connection, IP Network Interface Series 2 card A then sends a request, with the IP address and the RTP port just reserved, to Gateway B.

Upon receiving the request, Gateway B establishes a two-way IP connection and sends Gateway A the IP address and the RTP port that it reserved for the connection. In turn, IP Network Interface Series 2 card A changes the current one-way listen only connection to a full two-way connection.

To establish a one-way connection, use the *Route Control* (0x00E8) message and include the Universal ICB with the Source IP Address and Source Port Number TLVs. If the source IP address/port number TLVs are missing, an error is returned.

The Destination IP Address/Port Number and Connection Mode TLVs are optional.

Receive-only Connection

To establish a Receive-only connection, use the *Route Control* (0x00E8) message with the following TLVs:

- Source IP Address
- Source RTP Port Number
- Connection Mode: Receive-only
- Destination IP Address (Optional)
- Destination RTP Port Number (Optional)

Example: Message

```
 00 43 00 e8 00 00 ff 00 01 29 02 ff fe 03 03 00 33 00
12 00 02 27 92 00 04 0a 0a 24 15 27 93 00 04 00 00 00
01 02 1e 13 00 03 00 13 00 02 00 06 00 06 00 02 00 01
00 0f 00 01 0b 02 1e 07 00 01 01 db 00 01 01
```

Figure 3-3 IP Network Interface Series 2 Card Receive-only Connection

Transmit-Only Connection

To establish a Transmit Only connection, use the *Route Control* (0x00E8) message with the following TLVs:

- NPDI Universal ICB
- Source IP Address TLV
- Source RTP Port Number TLV
- Destination IP Address TLV
- Destination RTP Port Number TLV
- Connection Mode (Transmit Only)

Example: Message

```
00 67 00 e8 00 00 ff \
00 01 29 02 ff fe (Router AIB) 03 (ICB count)
03 00 33 00 22 00 04 (Universal ICB)
27 92 00 04 0a 0a 24 15 (Source IP Address TLV)
27 93 00 04 00 00 00 01 (Source RTP Port TLVs)
27 94 00 04 0a 0a 24 16 (Destination IP Address TLV)
27 95 00 04 00 00 00 01 (Destination RTP Port TLV)
02 1e 13 00 03 00 13 00 02 00 06 00 06 00 02 00 01 00 0f
   00 01 0b (Router ICB) 02 1e 07 00 01
01 db 00 01 02 (Connection Mode TLV)
```

Figure 3-4 IP Network Interface Series 2 Card Connection - Transmit-only Connection

Dynamic Connection Management

Introduction

To modify an existing connection, use the *Resource Attribute Configure* (0x00E3) message with various combinations of TLVs.

Change Destination Only

To change the destination of the connection without changing the Connection Mode, send a *Resource Attribute Configure* (0x00E3) message with the following TLVs:

- Address Element TLV
- Destination IP Address
- Destination RTP Port Number

Change Connection Mode

To change the Connection Mode, include the Connection Mode TLV. To also change the destination, enter the new Destination IP Address and RTP Port Numbers in the corresponding TLVs. To maintain the existing destination, enter the original destination information.

- Address Element TLV
- Destination IP Address
- Destination RTP Port Number
- Connection Mode

Use the Connection Mode TLV only when you want to modify a connection without changing the destination address. To modify a connection to a new destination, include the Destination IP Address and Destination RTP Port Number TLVs. In hold mode, no voice path is transmitted or received.

> **Important!** You cannot modify the source IP address or RTP port of an existing connection. If you use the Source IP Address/RTP Port Number TLVs, the CSP returns an error and the connection remains unchanged.

Change Destination Only

To modify the destination of an existing connection, use the *Resource Attribute Configure* (0x00E3) message with the following TLVs:

- Address Element TLV
- Destination IP Address TLV
- Destination RTP Port Number TLV

Example Message

```
00 2a 00 e3 00 00 ff 00 01 01 01 04 00 04
00 09 00 05 0d 03 00 01 00 (Address Element TLV)
27 94 00 04 0a 0a 24 16 (Destination IP Address TLV)
27 95 00 04 00 00 00 01 (Destination RTP Port TLV)
```

Figure 3-5 Change Destination Only

Change Connection Mode Only

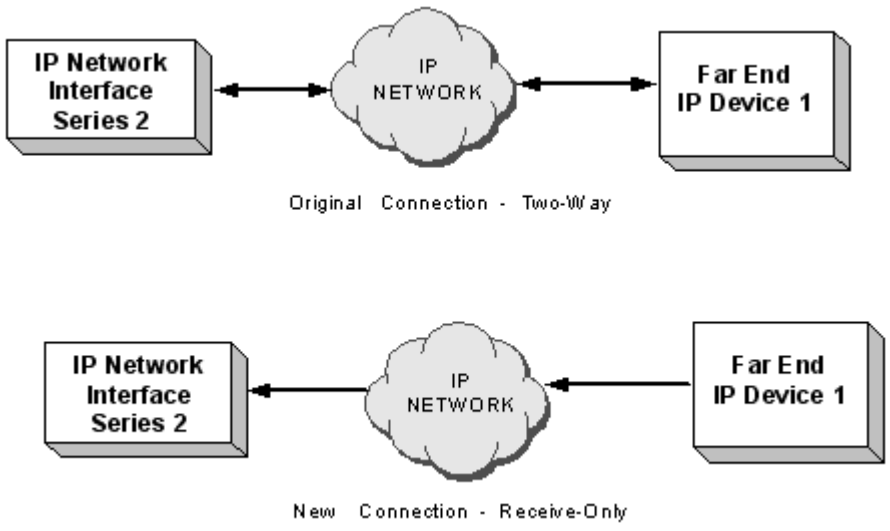
To modify the Connection Mode of an existing two-way connection to listen-only, use the *Resource Attribute Configure* (0x00E3) message with the following TLVs:

- Address Element TLV
- Connection Mode TLV; Receive-only

Example Message

```
00 1a 00 e3 00 00 ff 00 01 01 01 04 00 02
00 09 00 05 0d 03 00 01 00 (Address Element TLV)
01 db 00 01 01 (Connection Mode TLV)
```

Figure 3-6 Change Connection Mode Only



Change Destination and Connection Mode

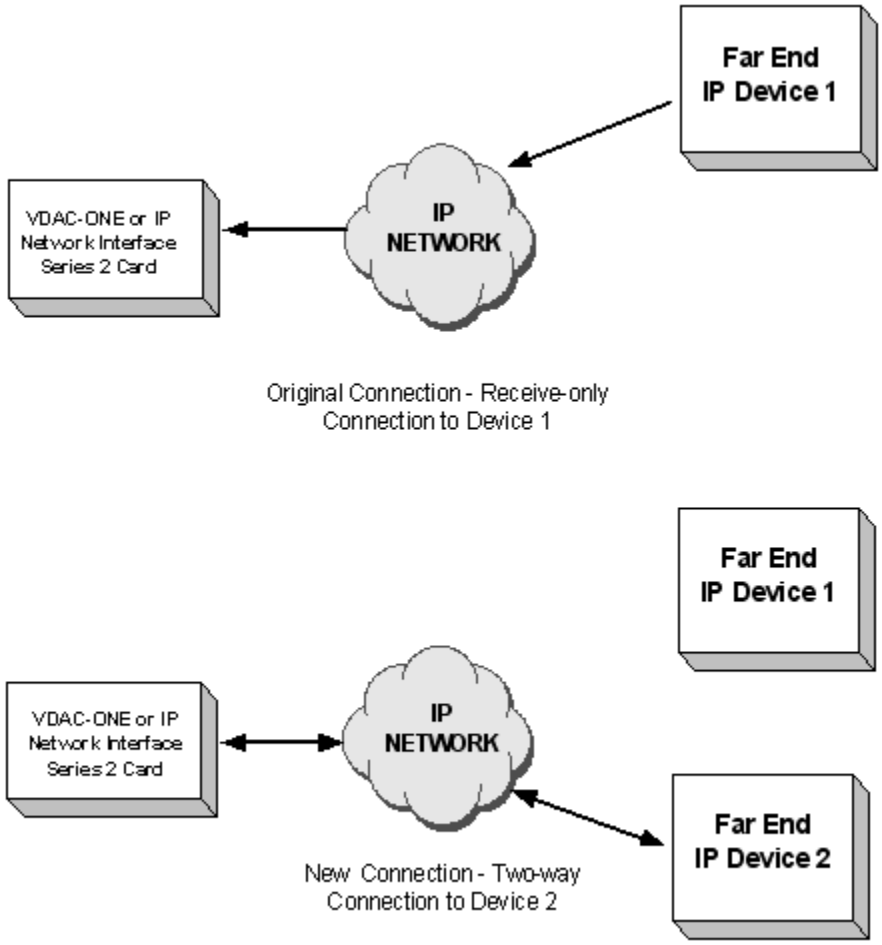
To change an existing listen-only connection to a two-way connection with a new destination, use the *Resource Attribute Configure* (0x00E3) message with the following TLVs:

- Address Element TLV
- Destination IP Address TLV
- Destination RTP Port Number TLV
- Connection Mode TLV (Two-way)

Example Message

```
00 2a 00 e3 00 00 ff 00 01 01 01 04 00 04
00 09 00 05 0d 03 00 01 00 (Address Element TLV)
27 94 00 04 0a 0a 24 16 (Destination IP Address TLV)
27 95 00 04 00 00 00 01 (Destination RTP Port TLV)
   01 db 00 01 02 (Connection Mode TLV)
```

Figure 3-7 Change Destination and Connection Mode



Release Mode Configuration

Overview

Release modes are managed by the Channel Management (CH) PPL component of Call Control. When a connection is terminated, the local- and distant-end release modes are used to determine what action to take on the released channels.

The terms *local* and *distant* are relative to the channel being configured. To specify whether a channel is released or parked when the other channel in a connection releases, you configure the channel's local-end release mode. To specify whether the other channel in the connection is released or parked when that channel releases, you configure the channel's distant-end release mode.

When a channel terminates a connection, the CSP refers first to the local-end release mode of the other end of the connection, and then to the distant-end release mode of the channel that initiated the release. If either end is set to park, the channel parks, otherwise, it is released. The host is informed of the state of a channel with either a *Channel Released* (0x0049) or a *DS0 Status Change* (0x0042) message.

Distant-end Release Mode

When you configure a channel, use the distant-end release mode so that the distant end (the other channel in the connection) is not released when the connection is terminated. For example, you would not want to release an inbound channel that is connected to a Voice Response Unit's (VRU) *Please Wait* message before it is queued up for an available agent.

When the announcement is finished, the VRU channel is released for other calls and the inbound channel is parked. The distant-end release mode of the VRU channel is set to park so that when the VRU releases at the end of the message, all inbound channels connected to the VRU park.

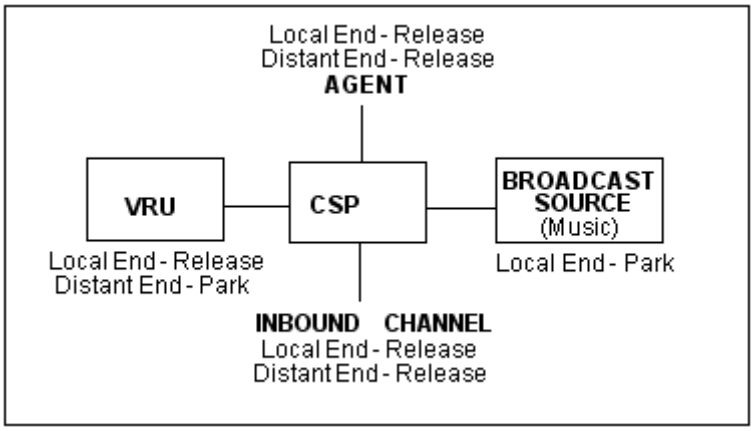
Local-end Release Mode

When you are configuring a channel, use local-end release mode so that the channel is not released when a connection is terminated. Consider the previous example for distant-end release mode. To configure a source to broadcast music to the channels waiting for an agent, the local-end release mode of the broadcast channel is set to park. When no channels are connected to the broadcast channel, it parks until another channel is connected. In contrast, if you allow the channel to release, it must be outseized after each connection is broken down.

The figure below illustrates the preceding examples. Both release modes for the inbound channel are set to release. Therefore, when the connection is torn down, it will be parked only if the distant-end release mode of the other channel is set to park (as is the case for the VRU and the Broadcast Source).

If the distant-end release mode of the other channel is set to release (as is the case with the agent channel) the inbound channel is released when the connection is torn down.

Figure 3-8 Release Mode Diagram



Configuration PPL Config Bytes

To configure Release Mode to either Park or Release, use the *PPL Configure* (0x00D7) message to modify the PPL Config Bytes of the CH PPL component.

- Local-End Release Mode - Config Byte 3
- Distant-End Release Mode - Config Byte 4

Release Mode API Messages

You can also change release modes by using either the *Local End Release Mode Configure* (0x0021) or the *Distant End Release Mode Configure* (0x00B8) message.

Loopback for Testing

Overview

To complete a local loopback voice call on a single IP Network Interface Series 2 card in a lab environment, the source and destination IP addresses for the call should be set to the IP addresses of the two local VoIP modules installed on the card.

Local Ethernet packet routing restrictions prevent loopback testing using a single VoIP module.

The *Route Control* (or *Outseize Control*) message is used to set the source and destination IP addresses and RTP ports for a call. The IP addresses must be different. The RTP ports are not required to be different. If you try to re-use a port number for a channel that is already being used for a call (RTP/RTCP/T.38), you will receive the following status error code:

0x4B: RTP Port in Use

For example, when using Profile 1 up to 512 (Profile 2 up to 256) simultaneous loopback calls can be setup between the two VoIP modules installed on a single IP Network Interface Series 2 card.

Important! In Profile 2 there are a total of 256 channels. However, one channel is used internally for ICMP support leaving 255 channels.

All RTP traffic between these two modules will be routed locally on the IP Network Interface Series 2 card (no data will be routed through the card's I/O Ethernet ports).

Routing Examples

Routing Based on Criteria Type, Source IP Address

For routing based on the Criteria Type, Source IP Address, use the Generic PPL TLV in the following format:

ICB Type	0x03
ICB Subtype	Generic PPL TLV (0x1E)
ICB Data Length	0x13
Number of TLVs	0x0003
Tag	Routing Method (0x0013)
Length	0x0002
Value	Data[0] 0x00 Criteria Type (MSB)
	Data[1] 0x08 Criteria Type (LSB)
Tag	Criteria Type (0x0008)
Length	0x0002
Value	Data[0] 0x27 (MSB) Source IP Address
	Data[1] 0x92 (LSB) Source IP Address
Tag	Router Protocol ID (0x000F)
Length	0x0001
Value	Data[0] 0x0B

Routing Based on Criteria Type, Any IP Channel

For routing based on the Criteria Type, Any IP Channel, use the Generic PPL TLV in the following format:

ICB Type	0x03
ICB Subtype	Generic PPL TLV (0x1E)
ICB Data Length	0x13
Number of TLVs	0x0003
Tag	Routing Method (0x0013)
Length	0x0002

Value	Data[0] 0x00 Criteria Type (MSB)
	Data[1] 0x08 Criteria Type (LSB)
Tag	Criteria Type (0x0008)
Length	0x0002
Value	Data[0] 0x00 (MSB) Any IP Channel
	Data[1] 0x65 (LSB) Any IP Channel
Tag	Router Protocol ID (0x000F)
Length	0x0001
Value	Data[0] 0x0B

Routing Based on Group ID

For routing based on the Route Group ID, use the Generic PPL TLV in the following format:

ICB Type	0x02
ICB Subtype	Generic PPL TLV (0x1E)
ICB Data Length	0x13
Number of TLVs	0x0003
Tag	Routing Method (0x0013)
Length	0x0002
Value	Data[0] 0x00 Route Group ID (MSB)
	Data[1] 0x06 Route Group ID (LSB)
Tag	Route Group ID (0x0006)
Length	0x0002
Value	Data[0] (MSB) Route Group ID
	Data[1] (LSB) Route Group ID
Tag	Router Protocol ID (0x000F)
Length	0x0001
Value	Data[0] 0x0B

Sample Route Control / Resource Attribute Configure

The *Route Control* message below establishes a one-way listen only connection:

```
00 43 00 e8 00 00 ff \' Header
00 01 29 02 ff fe \' Router AIB
03 \                    ' ICB count
03 00 33 00 12 00 02 27 92 00 04 0a 0a 24 15 27 93 00 04
   00 00 00 01 \' Universal ICB
02 1e 13 00 03 00 13 00 02 00 06 00 06 00 02 00 01 00 0f
   00 01 0b \' Router ICB
02 1e 07 00 01 01 db 00 01 01 \' Connection Mode TLV
```

The *Route Control* message below establishes a one-way talk only connection:

```
00 67 00 e8 00 00 ff \' Header
00 01 29 02 ff fe \' Router AIB
03 \                    ' ICB count
03 00 33 00 22 00 04 27 92 00 04 0a 0a 24 15 27 93 00 04
   00 00 00 01 \
' Universal ICB (SRC IP/RTP)

27 94 00 04 0a 0a 24 16 27 95 00 04 00 00 00 01 \'
   Universal ICB  (DST IP/RTP)
02 1e 13 00 03 00 13 00 02 00 06 00 06 00 02 00 01 00 0f
   00 01 0b \' Router ICB
02 1e 07 00 01 01 db 00 01 02 \' Connection Mode TLV
```

The *Resource Attribute Configure* message below modifies the current connection to a one-way listen only connection:

```
00 1a 00 e3 00 00 ff \' Header
00 01 01 01 04 00 02 \'Router AIB
00 09 00 05 0d 03 00 01 00 \' Address Element TLV
01 db 00 01 01      \' Connection Mode TLV
```

The *Resource Attribute Configure* message below establishes a new one-way talk only connection:

```
00 2a 00 e3 00 00 ff \' Header
00 01 01 01 04 00 04 \'Router AIB
00 09 00 05 0d 03 00 01 00 \' Address Element TLV
27 94 00 04 0a 0a 24 16 \' Destination IP address TLV
27 95 00 04 00 00 00 01 \' Destination RTP port TLV
01 db 00 01 02      ' Connection Mode TLV
```

PPL Information

Overview

The IP Network Interface Series 2 card Protocol Programmable Language (PPL) information from an external host perspective, is identical to the VDAC-ONE PPL. (Refer to *VDAC VoIP - 0x009C (2-44)*). The only difference is the L3 PPL Component on IP Network Interface Series 2 card has a new ID. The L3 PPL Component ID is 159 (0x9F). While technically, host downloadable protocols are possible, they are currently not supported.

PPL State Machines

PPL Component Name and ID	PPLcompVOCC_VOIP (0x9F)
DSD File Name	voip.dsd, boot.dsd
EXL File Name	voip.exl

PPL Events

Event ID	Event Description
0x01	Timer 1
0x02	Timer 2
0x03	Timer 3
0x04	Timer 4
0x0A	Layer 4/L3P Outseize Control
0x0B	Layer 4/L3P Clear Request
0x14	DSP Resource Available
0x15	DSP Resource Unavailable
0x16	DSP Resource Released
0x17	DSP Resource Inconsistency
0x19	RTP Port In Use
0x1A	Invalid Attributes
0x1B	Null Source in NPDI ICB
0x1E	Fax Start

Event ID	Event Description
0x1F	Fax End
0x23	Modem Start (not supported)
0x24	Modem End (not supported)
0x25	RTP Timer Expired
0x26	Modify Connection Attributes
0x27	Modify Connection Attributes Response
0x28	Media Inactivity Detection Timer Expired
0x403	Dummy Event
0x404	Channel In Service
0x405	Channel Out of Service

PPL Timers

Timer Name and Hex ID	Default Value
DSP Response Wait Timer (0x01)	12000 ms
Guard Wait Timer (0x02)	700 ms
Layer 4/L3P Clear Wait Timer (0x03)	4000 ms
Fax End Wait Timer (0x04)	5000 ms
Modem End Wait Timer (0x05) (not supported)	5000 ms
Fax Start Validation Timer (0x06)	100 ms
Modem Start Validation Timer (0x07) (not supported)	100 ms

Purge Reasons

Reason	Description
201	DSP Resource Wait Timeout
202	DSP Resource Release Wait Timeout
203	Layer 4/L3P Clear Wait Timeout
204	DSP Resource Inconsistency
205	DSP Resource Inconsistency
206	Internal PPL Error
207	Internal PPL Error

PPL Event Indications

Event ID	Event Description
0x01	Received Fax Start Event
0x02	Received Fax End Event
0x05	Received Modem Start Event (not supported)
0x06	Received Modem End Event (not supported)
0x07	Received Initial Media Inactivity Expired Event
0x08	Received Media Inactivity Timer Expired Event
0x0A	Incoming Registration Request
0x0B	Registration Timer Expired
0x0C	Incoming Deregistration Request
0x0D	Registration Query Response

Gateway Mode

Overview

You can configure the IP Network Interface Series 2 card for Gateway Mode which increases the capacity of the CSP to 4,096 physical channels for connections that comprise both circuit-switched and packet-switched channels.

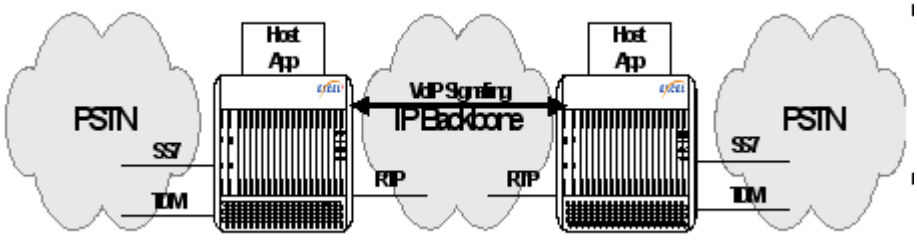
Gateway Mode is not supported on the VDAC-ONE card.

> **Important!** A successful connection requires at least one channel in Normal (Non-Gateway) Mode. A call cannot be placed between two Gateway Mode channels.

TDM/IP Gateway

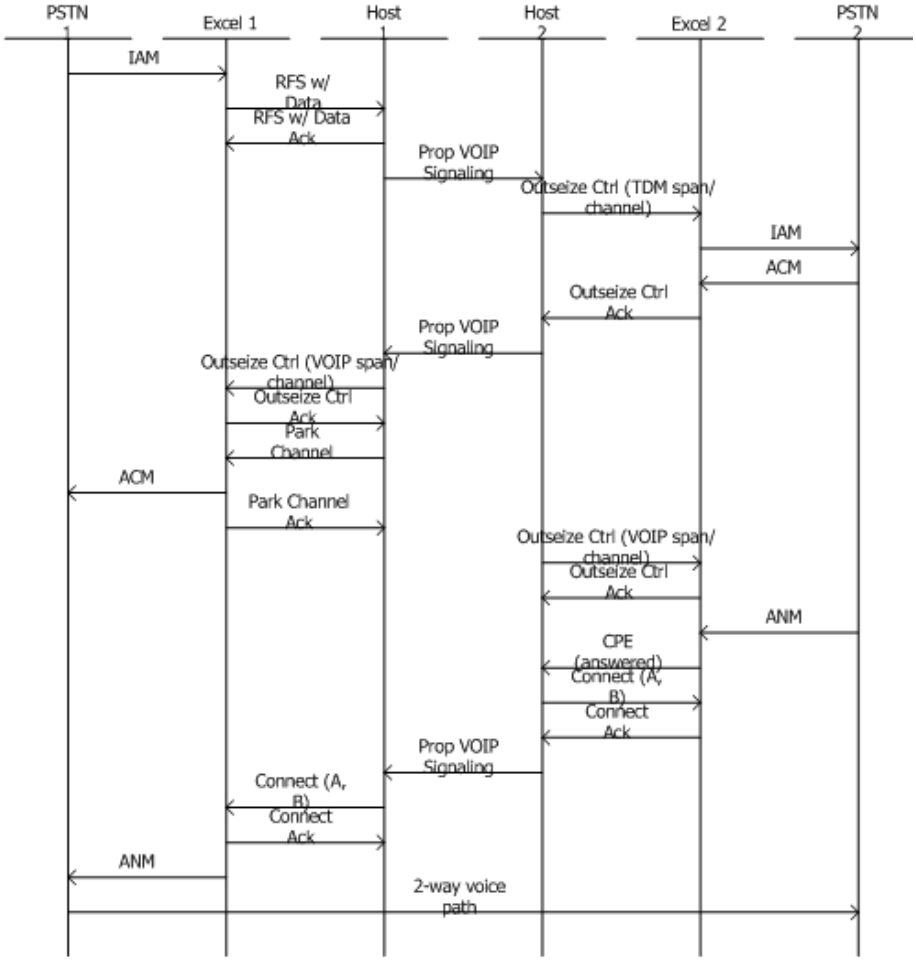
The diagram below shows how the CSP can be used in gateway mode, specifically as an SS7 front end gateway.

Figure 3-9 TDM/IP Gateway



Call Flow

The following call flow reflects a network like the one in the previous diagram.



DSP Cards

The DSP-ONE and DSP Series 2 cards both operate in Gateway Mode only. A call or connection cannot be placed between two Gateway Mode channels. Therefore you cannot have media services (such as Digit Collection, Tone Generation, and Announcement Playing) on Gateway Mode IP Network Interface Series 2 card channels. You can have a “gateway” call that involves a Normal Mode TDM channel and a Gateway Mode IP Network Interface Series 2 card channel with DSP services on the TDM channel only.

EXNET-ONE Card

The EXNET-ONE is a Gateway Mode card and does not operate with the IP Network Interface Series 2 card when it is in Gateway Mode. Connections can not be made between IPN 2 channels in Gateway mode across the Exnet® ring.

***Resource Attribute Configure* message**

See the Gateway Mode TLV in the *Resource Attribute Configure* message in the *API Reference*.

Querying the Ethernet Link Status, Duplex Type and Speed

Overview

This feature allows the host to query the following on the IP Network Interface Series 2 card at any time:

- the Ethernet Link status
- speed and duplex type of the NET1, NET2 and NET3 ports located on the Multi-Function Media I/O card

When restarting the host application or initiating a new host application, the Ethernet port status needs to be known. Based on that status, appropriate actions could be taken like taking all the channels OOS in that card. whenever the Ethernet link goes down, an *Alarm* message is sent to the host. When the Ethernet link comes back up, an *Alarm Clear* message is sent.

API Messages

This feature queries the individual Ethernet link status, speed and duplex type for the IPN Series 2 card using the *Card Query* message (0x0083). These are reported in the *Card Query* response. They will also be reported in the *Card Status Report* (0x00A6) message whenever the card comes in service after a reset.

The *Card Status Query* message response is obtained from the CSP when the host sends a *Card Status Query* message.

The *Card Status Query* message is obtained when the IPN Series 2 card comes into service after reset.

The message response in the *Card Status Query* (0x0083) and *Card Status Report* (0x00A6) messages will use bytes 12, 13 and 14 of the Hardware Configuration to report the status of ports NET1, NET2 and NET3 located on the Multi-Function Media I/O card.

Card Status Query Response

Card Information (Field length varies)

Hardware Configuration (16 bytes)
IP Network Interface Series 2 card

Data[12] Port 1 Info
Bit 0 Ethernet Link Status
0 = Ethernet link down
1 = Ethernet link up
Bit 1 Duplex Type
0 = Half Duplex
1 = Full Duplex
Bit 2-3 Speed
0 0 = 10Mbps

0 1 = 100Mbps
1 0 = 1000Mbps
Bits [4-7] Reserved (should be zero)

Data[13] Port 2 Info
Bit 0 Ethernet Link Status
0 = Ethernet link down
1 = Ethernet link up
Bit 1 Duplex Type
0 = Half Duplex
1 = Full Duplex
Bit 2-3 Speed
0 0 = 10Mbps
0 1 = 100Mbps
1 0 = 1000Mbps
Bits [4-7] Reserved (should be zero)

Data[14] Port 3 Info
Bit 0 Ethernet Link Status
0 = Ethernet link down
1 = Ethernet link up
Bit 1 Duplex Type
0 = Half Duplex
1 = Full Duplex
Bit 2-3 Speed
0 0 = 10Mbps
0 1 = 100Mbps
1 0 = 1000Mbps
Bits [4-7] Reserved (should be zero)

Data[15] Reserved (set to 0)

Card Status Report Message

Card Information (Field length varies)

Hardware Configuration (16 bytes)

IP Network Interface Series 2 card

Data[12] Port 1 Info
- Bit 0 Ethernet Link Status
  - 0 = Ethernet link down
  - 1 = Ethernet link up
- Bit 1 Duplex Type
  - 0 = Half Duplex
  - 1 = Full Duplex
- Bit 2-3 Speed
  - 0 0 = 10Mbps
  - 0 1 = 100Mbps
  - 1 0 = 1000Mbp
- Bits [4-7] Reserved (should be zero)

Data[13] Port 2 Info
- Bit 0 Ethernet Link Status
  - 0 = Ethernet link down
  - 1 = Ethernet link up
- Bit 1 Duplex Type
  - 0 = Half Duplex
  - 1 = Full Duplex
- Bit 2-3 Speed
  - 0 0 = 10Mbps
  - 0 1 = 100Mbps
  - 1 0 = 1000Mbp
- Bits [4-7] Reserved (should be zero)

Data[14] Port 3 Info
- Bit 0 Ethernet Link Status
  - 0 = Ethernet link down
  - 1 = Ethernet link up
- Bit 1 Duplex Type
  - 0 = Half Duplex
  - 1 = Full Duplex
- Bit 2-3 Speed
  - 0 0 = 10Mbps
  - 0 1 = 100Mbps
  - 1 0 = 1000Mbp
- Bits [4-7] Reserved (should be zero)

Data[15] Reserved (set to 0)

IP Network Series 2 NLP Control Support

Overview

The IP Network Series 2 card NLP Control Support feature allows the enabling or disabling of the Non-Linear Processor (NLP) sub-component within the IP Network Series 2 card Echo Cancellation control. This feature is supported by adding new bitmap values to the RTP Echo Cancellation (0x0103) TLV. This TLV remains backward compatible to allow enabling and disabling of the existing echo cancellation function.

This feature is not supported by the VDAC-ONE card.

Description

Currently, the master echo cancellation enable and disable option is the only option available for echo cancellation in the IP Network Series 2 card.

- If Echo Cancellation is enabled, the NLP is also enabled.
- If Echo Cancellation is disabled, the NLP is also disabled.

The advantage of disabling the NLP while echo cancellation is enabled, is in cases where heavy double-talk scenarios are expected. Double-talk results in clipping (removal) too much voice from the voice path, causing the voice recognition software to fail to detect voice patterns. Disabling the NLP solves this problem.

The NLP enable and disable function is allowed only when echo cancellation is enabled, since NLP cannot be enabled when echo cancellation is disabled.

API Messages Used

Resource Attribute Configure (0x00E3) message

Resource Attribute Query (0x00E4) message

Configuring and Querying

The RTP Echo Cancellation (0x0103) TLV is configured by the *Resource Attribute Configure* (0x00E3) message.

The RTP Echo Cancellation (0x0103) TLV can be queried by the *Resource Attribute Query* (0x00E4) message to show the correct value allowed.

TLV

The modified RTP Echo Cancellation (0x0103) TLV provides the following bitmap information in the value field:

- Bit 0 is the master enable and disable for Echo Cancellation

- Bit 1 enables and disables NLP

  NLP is enabled if Bit 1 is set to 0 (for backward compatibility), and disabled when Bit 1 is set to 1
- Bits 2-7 are reserved and set to 0 in all configuration attempts.

Refer to the TLV chapter in the *API Reference*.

4 IP Network Interface Series 3 Card

Overview

Purpose The document describes the IP Network Series 3 card including:

- Description
- Benefits to Customer
- Comparison of IPN-2 and IPN-3 Cards
- IP Network Interface Series 3 Voice Coder Packet Rate Information
- Software Requirements
- Obtaining Additional Software Fault Log Information

Description

The IP Network Interface Series 3 line card (hereafter referred to as the IPN-3 card) is the third generation IP Network card in the CSP product line. The following were the first two versions:

- VDAC-ONE card
- IP Network Interface Series 2 (hereafter referred to as the IPN-2 card)

Refer to *Comparison of IPN-2 and IPN-3 Cards (4-4)*.

Backward Compatible

Functionally, the IPN-3 card is very similar to the IPN-2 card. Beside some minor differences to a few OAM messages, the IPN-3 interfaces (internally and externally) are backward compatible with the IPN-2 card.

New VoIP Module

The IPN-3 has a new VoIP Module using the Mindspeed Picasso DSP Chips. There are some changes to the VoIP profiles that are described in the section, *Comparison of IPN-2 and IPN-3 Cards (4-4)*

New Physical Network Architecture

On the IPN-2 there are three Ethernet ports configured as a Link Aggregation Group (LAG) allowing a 300 Mbps pipes. The IPN-3 provides six Ethernet ports:

- two dedicated to a segregated control network
- four dedicated to a public network

Only two ports need to be trunked because the maximum bandwidth needed does not exceed 200 Mbps (about two DS3 worth of G.711).

Benefits to Customer

The following are benefits to migrating to the IPN-3 card:

- SwitchKit customers who replace their IPN-2 card with the IPN-3 card will not have to re-write their applications because the IPN-3 card is backward compatible. (Customers who do not use SwitchKit have to code the new board ID in their applications.)
- The IPN-3 has a more powerful CPU and more memory than the IPN-2 card. Refer to *Comparison of IPN-2 and IPN-3 Cards (4-4)* for the details.
- The IPN-3 binds a logical span/channel to a specific module so when a DSP fails the L4 channels can stay out of service which prevents further problems. To support this feature, the physical span offsets on the IPN-3 card is associated with a module. You assign the first eight spans (or 16 depending on the profile) to the first module and then assign the next eight (or 16) spans to the second module.

Comparison of IPN-2 and IPN-3 Cards

The following table provides the similarities and differences between the IPN Series 2 and the IPN Series 3 cards. Host developers must take the differences into account to integrate the IPN Series 3 card into a CSP

Table 4-1 IPN-2 and IPN-3 Cards

.

Information	IPN-2 Card	IPN-3 Card
CPU	PPC8260	PPC8270
Memory	128 MBs	256 MBs
Board ID	101 (0x65)	145 (0x91)
VoIP Module Board ID	104 (0x6D) Broadcom	105 (0x6E) Mindspeed
I/O Board ID	202 (0xCA)	146 (0x92) Media I/O Plus
Motherboard Dipswitch Settings	Refer to *CSP Installation and Maintenance Guide*.	Same as IPN-2 card
PCM Companding Law	u-Law	u-Law
CSP Matrix Series 3 Card Communication	HDLC mid-plane	HDLC mid-plane

Information	IPN-2 Card	IPN-3 Card
Network Interface	All ports support 10/100 Mbps full/ half duplex. Dip 4 control auto-negotiation or force full 100 Mbps Three Ethernet Ports configured as a single Link Aggregation Group. All ports connected to an unsecured public data network	All ports support 10/100 Mbps full/ half duplex. Dip 4 control auto-negotiation or force full 100 Mbps In the current release two ports are used: DATA 0 DATA 1 Future releases will include the following: - Four Ethernet Ports connected to an unsecured public data network: - Two Ethernet Ports connected the secured, Control network. - Redundant port links In the current release, the control ports and link redundancy on the data ports are disabled. For backward compatibility the *IP Address Assign* API message binds the motherboard IP address to the data network - not the control network.
Binding Logical Spans/Channels	Logical spans/channels are bound to the IPN-2 card	Logical spans are bound to a module.
L3 PPL Component ID	159 (0x9F)	159 (0x9F)
L3 PPL Protocol ID	BOOT Protocol=11 (0x0B) VOCC VoIP Protocol=12 (0x0C)	Same
Host L3 PPL Download Support	No	No
Line card download	TFTP Only	TFTP Only
TFTP Additions	VOCCVOIP_LOAD VOCCVM_H_LOAD VOCCVM_M_LOAD VOCCVM_L_LOAD	IPN3_LOAD VoIP images are embedded in the IPN-3 motherboard image

Information	IPN-2 Card	IPN-3 Card
VoIP Module Profiles	Profile #1: G.711 Only (512 channels) Profile #2: LBR+T.38 (256 channels)	Profile #4: G.711 Only (512 channels) Payload sizes 20ms and 30ms only. No RTP Redundancy. Profile #5: LBR+T.38 (256 channels)
VoIP Endpoint Terminal Capabilities	64 ms Echo Tail, 300ms Jitter Buffer	128 ms Echo Tail, 200ms Jitter Buffer
EXS API Interface and Usage	See IPN-3.	All messages used to interface with the IPN-2 card are used the same way with the IPN-3. All are identical except the following: *Card/Module Status Report/Query* (0xA6/0x83): The unused third module data is removed. *Resource Attribute Configure/ Query* (0xE3/0xE4): Slight differences in the terminal capabilities.
Default Resource Profile	LBR - G.711 + G.726 + G.723.1 + G.729A/B + T.38 (Profile #2) - 256 channel/modules	LBR - G.711 + G.726 + G.723.1 + G.729A/B + T.38 Profile #5) -256 channels/modules
Port Consumption Mode	Configure as either a port consuming or non-port consuming device. Granularity on line card level - not span level. In non-port consumption mode, can apply 3, 0, -2, -3, -4, -6, -9 dB of attenuation on both channels. Default is non-port consuming	Configure as either a port consuming or non-port consuming device. Granularity is on line card level - not span level. Can have four fixed pads: 3, 0, -3, -6 dB Default is non-port consuming
Resource Licensing	Each channel is logical span takes up one resource point. Unlicensed IPN-2 provides 96 resource points per module.	Each channel is logical span takes up one resource point. Unlicensed IPN-3 provides 96 resource points per module.
System Software Supported	8.1	8.4.1

IP Network Interface Series 3 Voice Coder Packet Rate Information

For the IP Network Interface Series 3 card there is a coupling between the Payload Type (codec) and Payload Size (packet rate). Unlike the VDAC-ONE card, the base packet rate (sampling) of a codec is different for different codecs. This implies that a Payload Size, which is defined in multiples of the base packet rate are specific to the Payload Type. The table below shows the relationship between the Payload Type and Payload Size on an IPN-3 card. It also shows a list of all Voice Coders supported by the IPN-3 card.

Table 4-2 IP Network Interface Series 3 Voice Coder Packet Rate Information

TLV Payload Type Data Value (decimal)	RTP Payload Type	Basic Packet Rate (ms)	Default Packet Rate (sampling) (ms)	Max Packet Rate (ms)
0	G.711 A-Law (64 Kbps)	5 ms	4x - 20 ms	6x - 30 ms
1	G.711 μ-Law (64 Kbps)	5 ms	4x - 20 ms	6x - 30 ms
2	G.726 (16 Kbps)	5 ms	4x - 20 ms	12x - 60 ms
3	G.726 (24 Kbps)	5 ms	4x - 20 ms	12x - 60 ms
4	G.726 (32 Kbps)	5 ms	4x - 20 ms	12x - 60 ms
5	G.726 (40 Kbps)	5 ms	4x - 20 ms	12x - 60 ms
15	G.723.1 (5.3 Kbps)	30 ms	1x - 30 ms	3x - 90 ms
16	G.723.1 (6.3 Kbps)	30 ms	1x - 30 ms	3x - 90 ms
17	G.729 (8 Kbps)	10 ms	2x - 20 ms	6x - 60 ms

Software Requirements

The IPN-3 card meets all of the software requirements specified for the IPN-2 card with the following exception:

- The Mindspeed modules do not have a software readable serial number - which is not necessary.

The following are new requirements for the IPN-3 card:

- The IPN-3 card interoperates with the IPN-2 card
- The IPN-3 can pass RTP traffic with the VDAC-ONE running RTP codecs G.711, G.723.1 and G.729

The IPN-3 can reside in the same chassis as an IPN-2 and a VDAC-ONE.

Obtaining Additional Software Fault Log Information

The IPN-3 card allows gathering additional fault log information via the debug port. This information is not available via the traditional *Fault Log Query* (0x0086) message.

Follow the steps below to capture the additional Fault Log information

1. Connect a debug cable from the IPN-3 card to a PC and run a program such as HyperTerminal. Make sure to enable writing of all information to a capture file prior to proceeding.
2. From the Debug Port command line, type the following:
   - **X** (to enable printing)
   - **o**
   - **os>f**
   - **fault>u**
   - **Format (A=ASCII B=BINARY C=RS232): c**
   - Once the card has finished printing, press **q**
   - Press **q** again to return to the main prompt.
   - **X** (to disable printing)
3. Stop the file capture.
4. Email the capture file to Dialogic Technical support along with log and configuration files.

5 Session-Initiation Protocol (SIP) Software

Purpose This chapter provides information about the CSP's implementation of Session-Initiation Protocol (SIP) software.

SIP Protocol Overview

Advanced Telephony on the Internet

Session-Initiation Protocol (hereafter referred to as SIP) is signaling protocol for Internet conferencing, telephony, presence, event notification, and instant messaging.

Dialogic not only SIP-enables its platform products but stays current as the protocol evolves.

SIP is typically required in a softswitch-controlled, converged network. In these converged networks, media gateways handle circuit/packet conversions (usually between IP and voice) and require media services such as tones, prompts, conferencing, and announcements.

Dialogic introduced SIP software to meet the demand for IP in converged services networks. This feature allows the CSP to act as an IP service node, providing application services and media resources to a softswitch or proxy server. The softswitch or SIP proxy server uses SIP to hand off a call requiring call treatment via partner-developed applications resident on a CSP. This provides services in Real-Time Protocol (RTP) streams to media gateways.

The SIP software is embedded in the CSP Matrix Series 3 Card, and interacts with host applications the same way that other Layer 3 circuit-based protocols do, such as Integrated Services Digital Network (ISDN) and SS7 Integrated Services Digital Network User Part (SS7 ISUP).

What SIP Allows

SIP allows the CSP to act as an IP Service Node, providing application services and media resources to a softswitch or SIP proxy server. A softswitch or SIP proxy server uses SIP to hand off a call requiring call treatment via partner-developed applications resident on a host.

Services Provided by SIP

SIP is part of the Internet Engineering Task Force (IETF) standards process (RFC 2543 BIS) and is modeled upon other Internet protocols, such as SMTP (Simple Mail Transfer Protocol) and HTTP (Hypertext Transfer Protocol.) It is used to establish, change, and tear down calls between one or more users in an IP-based network.

SIP Protocol Elements and Architecture

SIP is comprised of the following six elements in its architecture:

- User Agent Client (UAC)
- User Agent Server (UAS)
- SIP Terminal
- Proxy Server
- Redirect Server
- Location Server

These elements are grouped as follows:

User Agent

The User Agent is effectively the end system component for the call and has:

- The User Agent Client (UAC) as the client
- The User Agent Server (UAS) as the server

The client element initiates the calls and the server element answers the calls. This allows peer-to-peer calls to be made using a client-server protocol. A SIP terminal is a device, such as a SIP phone, that supports two-way, real time communications in a SIP network.

The CSP acts as the User Agent and derivations of a basic User Agent.

The SIP Network Server

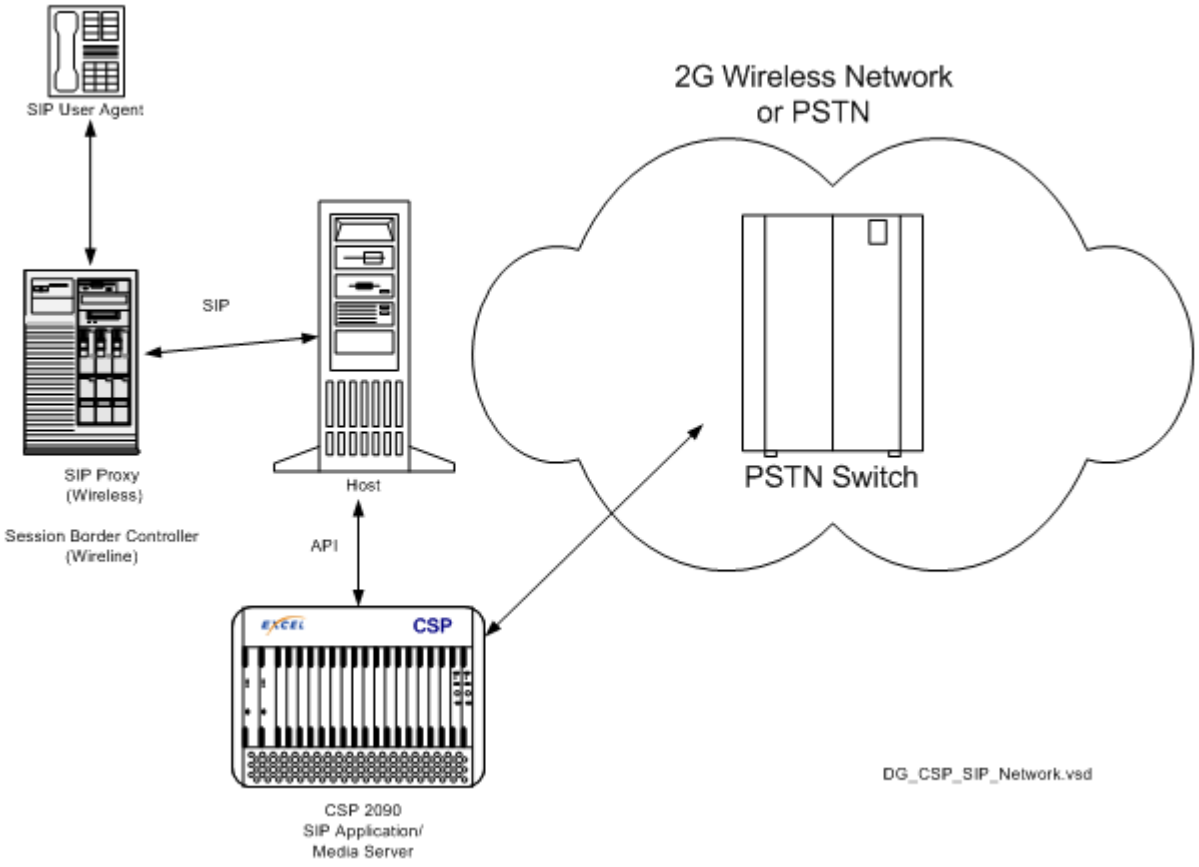
The SIP Network Server handles the signaling associated with multiple calls. There are three types of SIP Network Server elements:

- Proxy Server
- Redirect Server
- Location Server

The Proxy Server receives a SIP request, then passes the request along to one or more clients or next-hop servers. The Redirect Server accepts the SIP request, determines the new address, and returns the new IP address to the SIP client. The Location Server under SIP provides the current IP address of clients to the Redirect Server and to the Proxy Server on the network.

The following diagram shows a SIP based network architecture with the high-level functionality provided by the CSP. The network can be 3G Wireless or Wireline.

Figure 5-1 SIP Based Network Architecture



Functions

SIP must provide or enable the following functions:

Name Translation and User Location

This function ensures:

- The call reaches the called party wherever they are located.
- Descriptive information is mapped to location information.
- The details of the nature of the session (call) are supported.

Feature Negotiation

This function allows the group involved in a call (this may be a multi-party call) to agree on the supported features, recognizing that not all the parties can support the same level of features. For example, video may or may not be supported. SIP supports all Multipurpose Internet Mail Extensions (MIME). There is plenty of scope for negotiation.

Call Participant Management

During a call, a participant can bring other users into the call or cancel connections to current users. In addition, users could be transferred or placed on hold.

Call Feature Changes

A participant is able to change the call characteristics during the course of the call. For example, a call may have been set up as voice-only, but in the course of the call, the participant may need to enable a audio or fax function. A third party joining a call may need different features enabled in order to participate in the call.

CSP SIP Overview

Important! The SIP stack cannot use VDAC or IPN-2 resources on another node. All resources must be on the same node where the SIP call was received or placed.

Hardware Requirements

The following hardware is required to implement the SIP feature and supporting software applications in the current release:

- CSP Matrix Series 3 Card
- Chassis - Dialogic 2040, 2090 or 2110
- IP Network Interface Series 2 card or VDAC-ONE card for RTP streaming
- DSP-2 card or DSP-ONE card for DTMF tone detection and generation on the PSTN.

Other Prerequisites

Before you can configure your platform to accommodate SIP functionality, you must first have:

- An established, working IP network
- VDAC-ONE card or IP Network Interface card configured
- SIP Product License

SIP Call Overview

This section provides an overview of how the CSP SIP stack works including:

- *Configuration Messages (5-8)*
- *Configuration (5-9)*
- *Possible Modifications to Existing Applications (5-12)*
- *PPL Information: SIP UA 0x00A7 (5-14)*
- *Call Flow Messages (5-25)*
- *Routing with Route Control or Outseize Control messages (5-28)*
- *Tandem SIP-to-SIP Call Flow Example (5-31)*

Configuration Messages

You can use the following API messages to configure and query the SIP stack in the CSP. Refer to the *API Reference* for the details of each message.

- *VoIP Protocol Configure* (0x00EE)
  Used to configure the IP Network Interface card, VoIP modules, and VoIP Resource Attributes.
- *VoIP Protocol Query* (0x00EF)
  Used to query the VoIP configuration.
- *Resource Attribute Configure* (0x00E3)
  Use this message to configure attributes on a VDAC-ONE card, IP Network Interface Series 2 card, or a VoIP module. For VDAC-ONE cards, if a span/channel is active, this message waits until the call ends before updating the configuration.
  For IP Network Interface Series 2 cards, you can change the payload size and type during the call.
- *Resource Attribute Query* (0x00E4)
  This message queries either the default attributes of a module, or the current attributes of a channel involved in an IP call.
- *IP Address Configure (0x00A7)*
  Use this message to configure the subnet mask and assign either a single IP address or all IP addresses at once. You do this by matching the number of TLVs to the number of IP addresses being assigned. You can clear IP addresses by setting all of the IP address and subnet mask data to 0xFF. If you plan to take a card with IP addresses from one chassis and insert it into another chassis, be sure to clear all IP addresses on the card first. Otherwise, there might be a conflict with the IP addresses of the cards in the new chassis.
- *IP Address Query* (0x00A6)
  This message allows the host to query the IP address, subnet mask, and reset indicator for Common Channel Signaling cards, VDAC cards, and VDAC VoIP Modules.

Configuration

Overview

You can configure the CSP and SIP using either the SwitchKit Converged Services Administrator (CSA) GUI or the EXS API. The EXS API messages related to SIP are included in this document. For information on CSA, see the *Converged Services Administrator User's Guide.*

Basic Configuration Steps

The following are the basic steps to configure SIP software.

1. Activate SIP Software with product license keys For details, refer to *Downloading License Keys* in the *Licensing Overview* chapter in the *Developer's Guide: Overview* and the *Product License Download* message (0x0079) in the *API Reference.*
2. Assign IP Network Interface card IP Addresses.
3. Assign Logical Span IDs for IP Network Interface.
4. Configure Answer Supervision for Notify Host of Answer using the *PPL Configure (0x00D7)* message.
5. Bring the spans in service.
6. Bring the channels in service.
7. Configure optional IP Network Interface attributes.
8. Configure Routing.
9. Configure SIP attributes.

SIP Attributes

The following configuration options are supported for SIP using TLVs in the *VoIP Protocol Configure* (0x00EE) message.

Table 5-1 SIP Attribute

TLV Tag and Name	Default
0x01D0 Gateway Mode (In *Resource Attribute Configure* message)	Normal Mode (Non-Gateway Mode) (0x02)
0x01C8 Protocol Type	SIP (0x04)
0x0262 Use SIP Local Port	5060 (0x13C4)
0x0263 Use SIP Remote Port/Default Proxy Port	5060 (0x13C4)
0x0264 Use SIP Remote Host/Default Proxy Host	No default
0x0265 Use SIP Site ID	EXCEL-CSP
0x0266 Use SIP Anonymous Caller	0000
0x0267 Use SIP Invite T1 Timer	500 ms (0x1F4)
0x0268 Use SIP BYE T1 Timer	500 ms (0x1F4)
0x0269 Use SIP T2 Timer	4000 ms (0x190)
0x026A Use SIP Maximum Retransmissions Invite	7 (0x07)
0x026B Use SIP Max Retransmissions BYE	11 (0x0B)
0x0270 SIP Tunnel Type	Disabled
0x0272 Use SIP Registration Timeout	3600 s (0xE10)
0x0274 Use No Local Media	Terminate media locally (0x01)
0x0275 Local Registration Enable/ Disable	Enabled
0x0279 SIP Local Route Lookup	Disabled (0x00)
0x027A SIP Registration Mode	Do not notify host (0x00)
0x027C Min-SE Interval	300 s (0x12C)

Sample Configuration File

The following is a sample configuration file to configure SIP on the CSP:

```
Product licence for SIP

00 19 00 79 00 00 ff 01 02 24 10 46 46 42 35 4b 4a 47 58
   59 42 41 4c 4c 32 30 4b
'Deassign all spans
00 0d 00 a8 00 00 ff 00 01 11 04 ff ff ff ff



Deassign IP IP address for the VDAC
00 45 00 e7 00 00 ff 00 01 01 01 0c 00 06 01 09 ff ff ff
   ff ff ff ff ff 00 01 09 00 ff ff ff ff ff ff ff 00 01
   09 01 ff ff ff ff ff ff ff 00 01 09 02 ff ff ff ff ff
   ff ff 00 01 09 03 ff ff ff ff ff ff ff 00 02 00

00 45 00 e7 00 00 ff 00 01 01 01 0c 00 06 01 09 ff 87 77
   2C 38 ff ff ff e0 01 09 00 87 77 2C 39 ff ff ff e0 01
   09 01 87 77 2C 3a ff ff ff e0 01 09 02 87 77 2C 3b ff
   ff ff e0 01 09 03 87 77 2C 3c ff ff ff e0 02 00

'Assign spans to VDAC Card

00 0d 00 a8 00 00 ff 00 01 11 04 00 a8 0c 00
00 0d 00 a8 00 00 ff 00 01 11 04 00 a9 0c 01
00 0d 00 a8 00 00 ff 00 01 11 04 00 aa 0c 02
00 0d 00 a8 00 00 ff 00 01 11 04 00 ab 0c 03
00 0d 00 a8 00 00 ff 00 01 11 04 00 ac 0c 04

Configure Answer Supervision: Propagate answer and Notify
   host
00 17 00 bb 00 00 ff 01 02 0d 03 00 a8 00 0d 03 00 a8 1f
   00 61 01 01 02 02

00 17 00 bb 00 00 ff 01 02 0d 03 00 a9 00 0d 03 00 a9 1f
   00 61 01 01 02 02
00 17 00 bb 00 00 ff 01 02 0d 03 00 aa 00 0d 03 00 aa 1f
   00 61 01 01 02 02
00 17 00 bb 00 00 ff 01 02 0d 03 00 ab 00 0d 03 00 ab 1f
   00 61 01 01 02 02
00 17 00 bb 00 00 ff 01 02 0d 03 00 ac 00 0d 03 00 ac 1f
   00 61 01 01 02 02

VoIP Configure

00 0f 00 ee 00 00 ff 00 00 00 00 01 01 c8 00 01 04
```

Possible Modifications to Existing Applications

Overview

The SIP layer functions as any other signaling Layer 3 in the CSP, such as SS7 ISUP and ISDN. This design abstracts the application developer from SIP internals and presents a unified call control API interface.

API Messages

You might need to add or modify the following API messages to your application to implement SIP functionality:

- *Route Control 0x00E8*
  Used to initiate an outbound call. Refer to *Routing SIP and H.323 Calls Using Route Control Message (8-3)*
- *Outseize Control 0x002C*
  Refer to *Routing SIP and H.323 Calls Using Outseize Control Message (8-14)*
- *VoIP Protocol Configure 0x00EE*
  Used to configure the IP Network Interface card, VoIP modules, and VoIP Resource Attributes.
- *VoIP Protocol Query 0x00EF*
  Used to query the VoIP configuration.
- *Request For Service With Data 0x002D*
  Used to report an inbound SIP call to the host application. Embodies the parameters received in the inbound call.
- *Channel Released With Data 0x0069*
  Used to include Network Protocol Data Intelligence (NPDI) Universal TLV for SIP Release Code (0x2915).
- *Release Channel With Data 0x0036*
  Used to include NPDI TLV for SIP Release Code to reject an incoming SIP call.
- *PPL Event Request 0x0044*
  Used for SIP Registration and Call Control.
- *PPL Event Indication 0x0043*
  Used for SIP Registration and Call Control.

VDAC-ONE or IP Network Interface Card Requirements

The type of application determines the requirements of the VDAC-ONE or IP Network Interface Series card for RTP functionality in a SIP implementation on the CSP. These cards are required for feature-rich services such as Prepaid Calling Services, Unified

Messaging, or Media Server Applications that require control of multiple legs of a call, tones/announcements, and transcoding (different terminal capability set).

PPL Information: SIP UA 0x00A7

Overview The following contains information on PPL Event IDs used for the SIP User Agent (UA).

PPL Event Requests

Table 5-2 PPL Event Requests

PPL Event ID	Purpose	Description	AIB Used
SIP Registration Support (See details following this table.)			
0x000A	Accept Registration Request	The host application uses this ID to accept the registration request.	0x7F - Stack ID
0x000B	Reject Registration Request	The host application uses this ID to reject the registration request.	0x7F - Stack ID
0x000C	Authenticate Request	NPDI formatted data for scheme, realm, username and password required to authenticate the incoming SIP call.	0x7F - Stack ID
0x0010	Query All Registration	Queries all registration entries in the CSP database.	0x7F - Stack ID
0x0011	Query Single Registration	Queries individual registration information from the CSP.	0x7F - Stack ID
0x0012	Delete All PPL Registrations	Deletes all end points registered with the CSP.	0x7F - Stack ID
0x0013	Delete Specific Registration	Deletes individual registration information from the CSP. The host application must provide the To Username parameter in the request that it sends.	0x7F - Stack ID
Call Control (See details following this table.)			

PPL Event ID	Purpose	Description	AIB Used
0x000C	Authenticate Request	NPDI formatted data for scheme, realm, username and password required to authenticate the incoming SIP call.	0x0D - Channel
0x001E	Change Local Media Connection Attribute	Used to change the local media connection attribute.	0x0D - Channel
0x001F	Generate Early Media	Used by host application to generate early media by sending 183 response code with SDP for an incoming call.	0x0D - Channel
0x0020	Notify	Contains TLV *0x294B Notify Status* which contains the SIP status to be sent to the UA, within a NOTIFY message.	0x0D - Channel
0x0021	Generates SIP ACK with SDP	Used in CAM delayed media scenario to send answer media parameters in ACK. This PPL ER must contain the TLV 0x2A00 Media Remote End Point Information.	0x0D - Channel
0x0023	Subsequent Data	Used by the host application to send data to the CSP that could not be sent in the *Route Control* message.	0x0D - Channel
0x0024	Generate RE-INVITE	Generates RE-INVITE message to the SIP endpoint connected to that channel Works only if the call is in bearer-free mode.	0x0D - Channel
0x0025	Generate INFO	Generates INFO Message.	0x0D - Channel
0x0026	Generate 182	Generates 182 Queued message	0x0D - Channel

PPL Event ID	Purpose	Description	AIB Used
0x0027	Generate outbound REFER	Generates an outbound REFER request with the refer target URL filling in the TLV	0x0D - Channel
0x0029	Do Not Close Socket	Prevents automatic closing of a socket - typically a TCP socket.	0x0D - Channel

SIP Registration Support

A call handle is used for all SIP Registration PPL Event Requests. Events 0x000A 0x000B 0x000C use the Call Handle that was reported in the associated PPL Event Indication. All other registration events use a Call Handle of 0xFFFF

Example: Querying all Registrations:
00 12 00 44 00 00 ff 00 01 7f 04 ff 00 ff ff 00 a7 00 10 00

Call Control

Connection mode can be changed with TLV 0x27B3 NPDI RTP Connection Mode.

Change Local Media Connection Attribute

Example: Change the Connection to Listen Mode:

```
00 1d 00 44 00 02 ff 00 01 0d 03 00 01 04 00
a7 00 1e 01 03 00 33 00 07 00 01 27 b3 00 01
01
```

Generate Local Media

Example: Generate Early Media H->X:

```
00 11 00 44 00 00 ff 00 01 0d 03 00 01 02 00
a7 00 1f 00
```

Authenticate Request

The following TLVs are used with the PPL event ID 0x0C:

- 0x2937 NPDI SIP Authenticate Scheme
- 0x2938 NPDI SIP Authentication Realm
- 0x2939 NPDI SIP Authenticate Username
- 0x293A NPDI SIP Authenticate Password
- 0x293F NPDI SIP Authentication Timeout

The following TLVs are used with the *Route Control* message.

- 0x293B NPDI SIP Authorization Username
- 0x293C NPDI SIP Authorization Password
- 0x293D NPDI SIP Proxy Authorization Username
- 0x293E NPDI SIP Proxy Authorization Password

Call Socket

The remote_ip variable is the four-byte IP Address of the remote socket that was reported to the host application in the *Request for Service* message with the SIP Remote IP Address TLV.

The remote_port variable is the two-byte port number of the remote socket that was reported to the host application in the *Request for Service* message with the SIP Remote IP Port TLV**.**

```
00 27 00 44 00 00 FF 00 01 7f 04 ff 00 ff ff 00
a7 00 28 01 03 00 33 00 10 00 02 29 4E 00 04 0A
0A 29 4F 00 02 00 01
```

Don’t Close Socket

In response to a *Request for Service* message containing a SIP Transport Type TLV containing the following:

- a value of 2 = TCP
- the SIP Remote IP Address
- SIP Remote IP Port TLVs

The host application may determine that the underlying socket needs to remain open to allow the reuse of this socket for multiple transactions. When that is the case, the host application immediately sends this *PPL Event Request* message to prevent the socket from closing.

Example:

```
00 11 00 44 00 00 FF 00 01 0d 03 00 00 01 00
a7 00 29 00
```

PPL Event Indications

Table 5-3 PPL Event Indications

PPL Event Indication ID	Purpose	Description
SIP Registration Support		
0x000A	Incoming Registration Request	Informs the host application of an incoming REGISTER message.
0x000B	Registration Timer Expired	Informs the host application about an expired registration.
0x000C	Incoming De-registration Request	Informs the host application that a remote UA requested to delete a registration.
0x000D	Registration Query Response	Sends registration information to the host application as a response to a query message.
Call Control		
0x001E	Media Change Detected	Sent to the host application whenever the remote side changes its media connection attributes with the RE-INVITE message.

PPL Event Indication ID	Purpose	Description
0x001F	183 Progress	Notifies the host application that the remote side sent a 183 Response Code. If SDP data is in the response, the CSP sends the data to the host application in NPDI TLV format as follows: 0x2794 - Destination IP Address 0x2795 - Destination RTP Port 0x29FF - Media Local End Point Information - if Call Agent is enabled (must be less than 250 bytes).
0x0020	Answer on B-Side	Sent to the host application when the CSP receives a 200 OK response with a Session Description Protocol (SDP) for an outbound call. This PPL Event Indication contains the TLV 0x29FF Media Local End Point Information (must be less than 250 bytes).

PPL Event Indication ID	Purpose	Description
0x0021	Blind Refer PPL Event Indication	Notifies the host application that a Blind Refer event has occurred. The host application must respond by sending the PPL Event Request *Notify* with an appropriate SIP status. This PPL Event Indication will contain several TLVs as follows: 0x2916 SIP A Leg Transport Type 0x2919 NPDI SIP To User Name 0x291B NPDI SIP To Host Name 0x291C NPDI SIP to Port Name 0x291A NPDI SIP To Password (optional)

PPL Event Indication ID	Purpose	Description
0x0022	Consultative Refer PPL Event Indication	Notifies the host application that a Consultative Refer Event has occurred. The host application must respond by sending the PPL Event Request *Notify* with an appropriate SIP status. The PPL Event Indication contains the following TLVs: 0x2919 NPDI SIP To User Name 0x291B NPDI SIP To Host Name 0x291C NPDI SIP to Port Name 0x294C Refer Spans Channel 0x291A NPDI SIP To Password (optional)
0x0023	Subsequent Data	Notifies the host application that the CSP is sending more data that could not be sent in the *Request for Service with Data* message.

PPL Event Indication ID	Purpose	Description
0x0024	180 Ringing	Notifies the host application that the remote side sent a 180 Response Code. If SDP data is in the response, the CSP sends the data to the host application in NPDI TLV format as follows: 0x2794 - Destination IP Address 0x2795 - Destination RTP Port 0x29FF - Media Local End Point Information - if Call Agent is enabled (must be less than 250 bytes).
0x0025	Send subsequent data	Notifies the host application that the SIP UA is ready to accept more data from the host application.
0x0026	All subsequent data received	Notifies the host application that the SIP UA received the last message in the sequel.

PPL Event Indication ID	Purpose	Description
0x0027	Result of the sent REFER request	Reports a REFER response was received and the Response Code received. The value in the SIP Response Code TLV (0x2915) reports the exact response code Success - 2xx Failure - 4xx-6xx
0x0028	Status of the Reference	Reports a NOTIFY request was received and the response code from the NOTIFY body. The value in the SIP Response Code TLV (0x2915) reports the exact response code.
0x0029	Reported to the host when the CSP receives an ACK.	Contains the SDP received in the SIP ACK message in 0x29FF TLV (Media Local End Point Information)

PPL Event Indication ID	Purpose	Description
0x0031	422 Session Timer Too Small	Reports the Session Expiry Interval and Minimum Session Expiry interval used in the Re-transmit INVITE message. The values in the following TLVs report this information: SIP Session Expiry Interval (0x2969) SIP Minimum Session Expiry Interval (0x2970)
0x002C	INFO Received	Report receipt of INFO message.
0x002D	Response for INFO	Report the response received for the INFO message sent.
0x002E	182 Received	Report receipt of 182 Queued message.

Call Flow Messages

Overview

The generic format of the API messages below are documented in the *API Reference*. The following is specific to SIP Call Control.

***Route Control* (0x00E8)**

You must use the Generic Router AIB (0x0029) as indicated below.

Examples of *Route Control* Message

The following are three example of how to use the *Route Control* message with SIP software. This *Route Control* message assumes that you are using a local Registration on the CSP. The outbound INVITE Request message is directed using Registration information matching the specified To User Name.

```
00 41 00 e8 00 00
ff 00 01 29 02 ff fe
02 03 00 1e 00 19
00 04
00 13 00 02 00 08
00 08 00 02 00 65
00 0f 00 01 0b
00 65 00 02 00 00
03 00 33 00 12
00 02
27 7e 00 03 08 00 00        'REMOTE SIDE PROTOCOL'
29 19 00 05 35 35 35 35 00  'SIP TO USER NAME'
```

This *Route Contro*l message creates an outbound INVITE Request message directed using the specified user@host name. (The user is not locally registered.)

```
00 56 00 e8 00 00
ff 00 01 29 02 ff fe
02 03 00 1e 00 19
00 04
00 13 00 02 00 08
00 08 00 02 00 65
00 0f 00 01 0b
00 65 00 02 00 00
03 00 33 00 25
00 03
27 7e 00 03 08 00 00      'REMOTE SIDE PROTOCOL'
29 19 00 05 35 35 35 35 00'SIP TO USER NAME
29 1b 00 0f 31 33 35 2e 31 31 39 2e 35 31 2e
   31 38 37 00            'SIP TO HOST NAME'
```

This *Route Control* message creates an outbound INVITE directed to the default Proxy IP address and uses the specified “To” User Name.

```
00 54 00 e8 00 00
ff 00 01 29 02 ff fe
02 03 00 1e 00 19
00 04
00 13 00 02 00 08
00 08 00 02 00 65
00 0f 00 01 0b
00 65 00 02 00 00
03 00 33 00 25
00 03
27 7e 00 03 08 00 00        'REMOTE SIDE PROTOCOL
29 19 00 05 35 35 35 35 00  'SIP TO USER NAME
29 0e 00 0f 31 33 35 2e 31 31 39 2e 35 31 2e 31
   38 35 00                 'SIP PROXY IP ADDRESS
```

***Request for Service with Data* (0x002D)**

The CSP reports incoming calls from the SIP side using the *Request for Service with Data* (0x002D) message. An Extended Data ICB NPDI Universal Data (0x33) is used in SIP. The UPDF parameters supported are the same as those supported in the *Route Control* message.

***Release Channel with Data* (0x0036)**

The Release Data Type is defined as follows:

0x0033 NPDI Universal ICB.

You can use this ICB to add UPDF data to the *Release Channel with Data* message.

You can use the following TLV to encode the SIP response code to be used in the SIP response message generated by the CSP.

0x2915 NPDI SIP Response Code

```
To release an incoming SIP call with 404 reason

   H->X
   00 20 00 36 00 00 ff 00 02 0d 03 00 01 08 0d 03 00 01
   08 33 01 03 00 33 00 08 00 01 29 15 00 02 01 94
```

***Channel Released with Data* (0x69)**

The following TLV is used for SIP:

0x2915 NPDI SIP Response Code

Used to encode the SIP response code when an outbound SIP call is rejected by the remote SIP UA.

```
OUTBOUND call rejected by remote side with 404 reason

   X -> H
   00 1a 00 69 00 04 ff 00 01 0d 03 00 01 0b 01 03 00 33
   00 0800 01 29 15 00 02 01 94
```

Routing with *Route Control* or *Outseize Control* messages

Overview

This section provides an overview of routing VoIP calls controlled by SIP where your host application uses the *Route Control* or *Outseize Control* message to route the outbound side of the call. This section contains references to full explanations of each process.

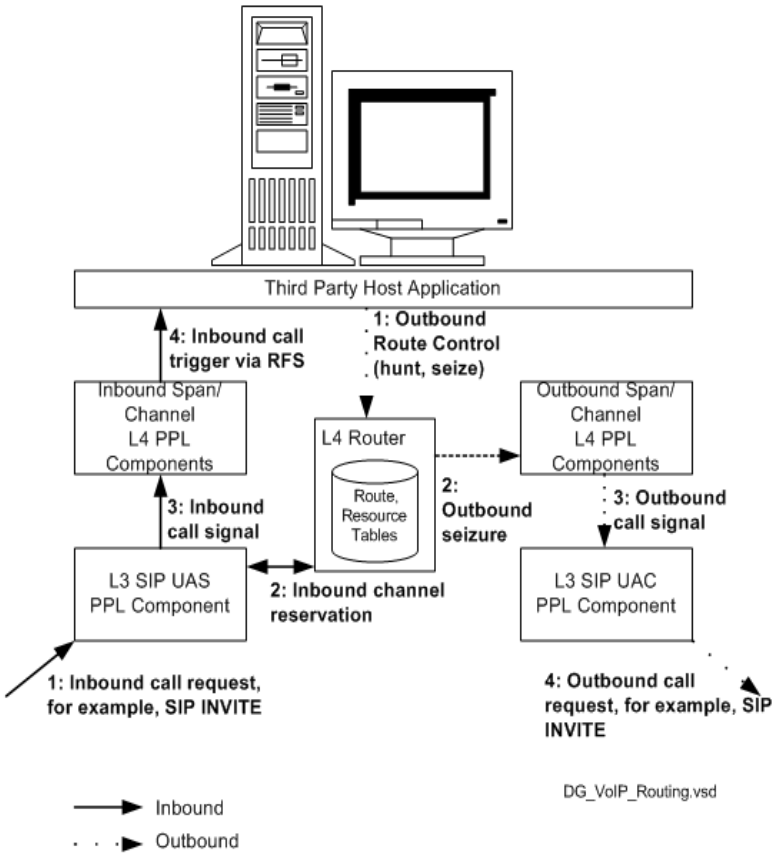
Route Control

Using the *Route Control* message allows the host application to leverage Layer 4 Router functionality thereby relieving the host application from making the span/channel assignments on the IP Network Interface card modules.

Routing Process

The following diagram shows a call controlled by SIP signaling Refer to *Routing SIP and H.323 Calls Using Route Control Message (8-3)* for a full explanation of this process.

Figure 5-2 Routing Overview - Inbound and Outbound



Outseize Control

VoIP applications can use the *Outseize Control* (0x002C) message for outbound SIP signaling. This method is required for application developers who:

- are migrating TDM applications to the VoIP environment if the TDM applications already use the *Outseize Control* message
- want to control the selection of spans/channels

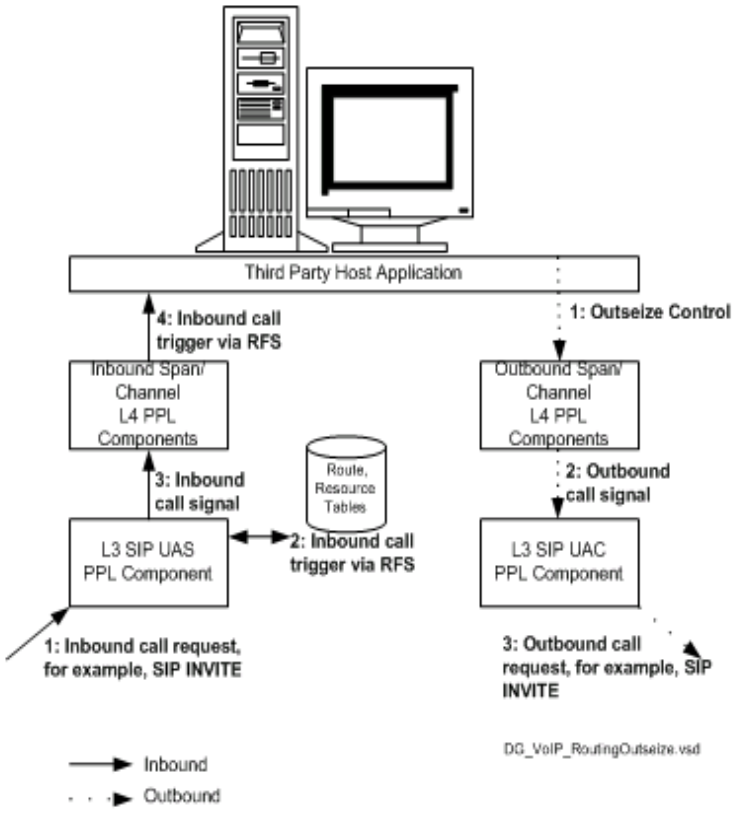
Important! The host application is responsible for selecting the span and channel for each outbound call on a per call basis. The host application also selects the source IP address and port.

If you do not want the host application to be responsible for the channel management, you can use the *Route Control* message instead. Refer to *Routing SIP and H.323 Calls Using Route Control Message.*

Routing Process

The figure below provides an overview of the routing process. Note that the inbound side of the call is the same as described in the previous section. The outbound side involves the *Outseize Control* message rather than the *Route Control* message. Refer to *Routing SIP and H.323 Calls Using Outseize Control Message (8-14)* for a full explanation of this process.

Figure 5-3 Routing Process with *Outseize Control* Message

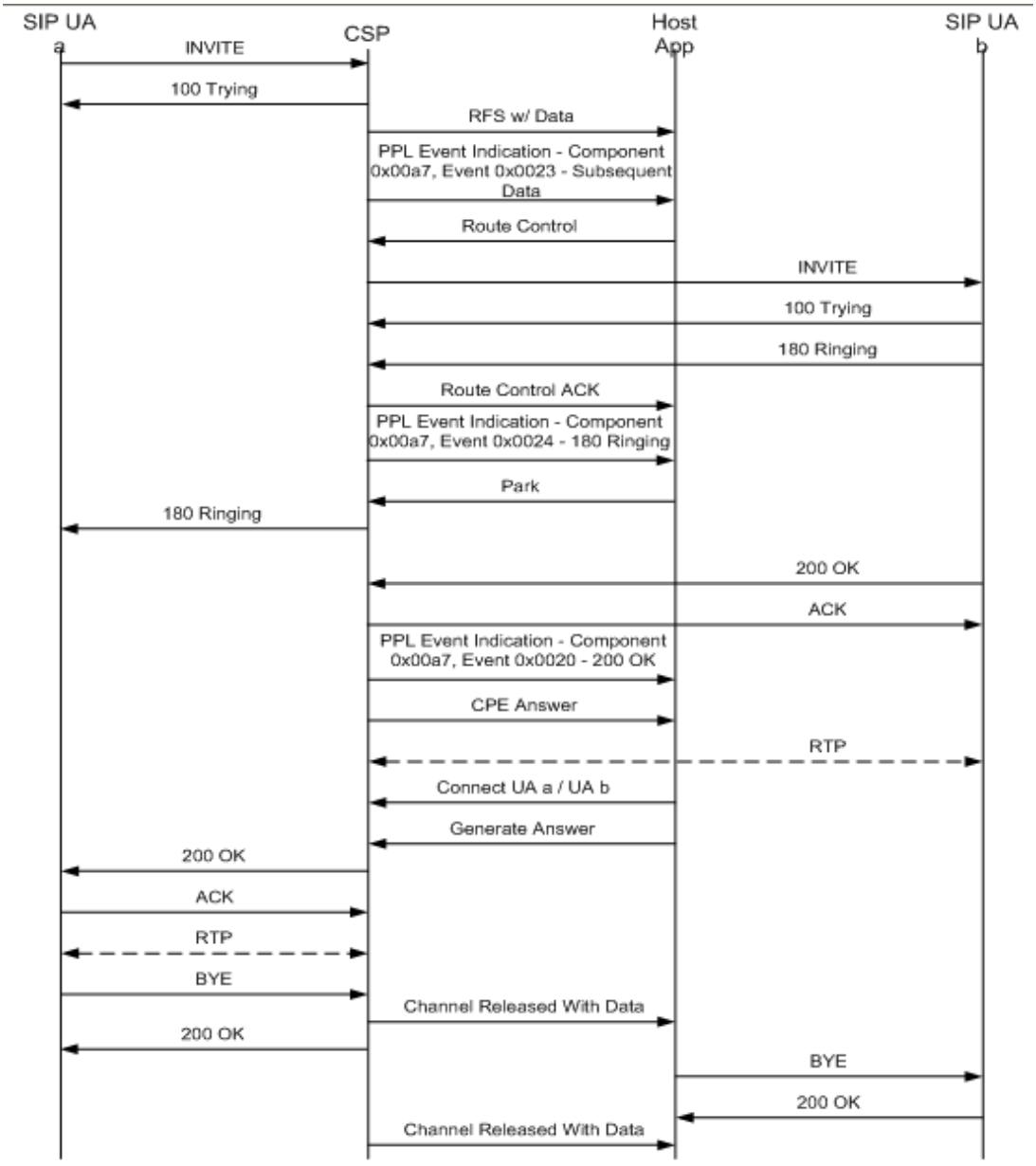


Tandem SIP-to-SIP Call Flow Example

Overview

This section contains two SIP-to-SIP call flows and message traces The first call flow uses the *Route Control* (0x00E8) message the second one uses the *Outseize Control* (0x002C) message.

With *Route Control message*



Message Trace

```
1 -RECEIVED From 192.168.1.51:51035 at 2157
INVITE sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK034df162
From: "User ID"
   <sip:8623001@192.168.1.102>;tag=00036b095130002d3ac6a6
   6f-0a11923d
To: <sip:8623000@192.168.1.102>
Call-ID: 00036b09-51300033-4fc0f34b-4c21a745@192.168.1.51
CSeq: 101 INVITE
User-Agent: CSCO/7
Contact: <sip:8623001@192.168.1.51:5060>
Expires: 180
Content-Type: application/sdp
Content-Length: 246
Accept: application/sdp

v=0
o=Cisco-SIPUA 27743 2333 IN IP4 192.168.1.51
s=SIP Call
c=IN IP4 192.168.1.51
t=0 0
m=audio 27148 RTP/AVP 0 8 18 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15


2 -SENT To 192.168.1.51:5060 at 2157
SIP/2.0 100 Trying
To: <sip:8623000@192.168.1.102>;tag=315186d
From: "User ID"
   <sip:8623001@192.168.1.102>;tag=00036b095130002d3ac6a6
   6f-0a11923d
Call-ID: 00036b09-51300033-4fc0f34b-4c21a745@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK034df162
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [01 33 00 2d 00 05 00 00 01 0d 03 00 00 05 00 33 01 03
   00 33
     01 1f 00 18 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 38 36 32 33 30 30 30 00 29 1b 00 0e 31 39 32
     2e 31 36 38 2e 31 2e 31 30 32 00 29 1c 00 04 00 00 13
     c4 29 23 00 08 38 36 32 33 30 30 31 00 29 25 00 0e 31
```

```
     39 32 2e 31 36 38 2e 31 2e 31 30 32 00 29 26 00 04 00
     00 13 c4 29 2d 00 08 38 36 32 33 30 30 31 00 29 2f 00
     0d 31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 30 00 04
     00 00 13 c4 29 33 00 01 01 27 18 00 09 02 00 00 00 07
     86 23 00 10 27 17 00 07 02 00 07 86 23 00 00 27 94 00
     04 c0 a8 01 33 27 95 00 04 00 00 6a 0c 27 b0 00 02 00
     02 27 b1 00 02 00 01 29 16 00 01 01 29 54 00 08 38 36
     32 33 30 30 30 00 29 55 00 0e 31 39 32 2e 31 36 38 2e
     31 2e 31 30 32 00 29 56 00 04 00 00 13 c4 29 50 00 31
     30 30 30 33 36 62 30 39 2d 35 31 33 30 30 30 33 33 2d
     34 66 63 30 66 33 34 62 2d 34 63 32 31 61 37 34 35 40
     31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 53 00 02 01
      00]

H->X
    [00 0c 00 2d 00 05 00 00 01 0d 03 00 00 05]

X->H
    [00 58 00 43 00 0d 00 00 01 0d 03 00 00 05 00 a7 00 23
   01 03
     00 33 00 42 00 04 29 53 00 02 02 01 29 51 00 22 30 30
     30 33 36 62 30 39 35 31 33 30 30 30 32 64 33 61 63 36
     61 36 36 66 2d 30 61 31 31 39 32 33 64 00 29 52 00 08
      33 31 35 31 38 36 64 00 2a 0e 00 04 c0 a8 01 33]

H->X
    [00 05 00 43 00 0d 00]

H->X
    [00 72 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 43 00 05 27 7e 00
     03 08 00 00 29 19 00 08 38 36 32 33 30 30 30 00 29 1b
     00 0d 31 39 32 2e 31 36 38 2e 31 2e 31 31 00 29 23 00
     08 38 36 32 33 30 30 31 00 29 25 00 0d 31 39 32 2e 31
      36 38 2e 31 2e 35 31 00]


3 -SENT To 192.168.1.11:5060 at 2158
INVITE sip:8623000@192.168.1.11:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.11:5060>
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=1934922386e
Call-ID: EXCEL-CSP0.78e.2158.430@192.168.1.102
Contact: 8623001<sip:8623001@192.168.1.102:5060>
User-Agent: Excel_CSP/83.10.57
Supported: timer
Session-Expires: 1800
Min-SE: 300
```

```
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.132
t=0 0
m=audio 15736 RTP/AVP 0


4 -RECEIVED From 192.168.1.11:5060 at 2158
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1934922386e
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=2645473871
Contact: <sip:8623000@192.168.1.11:5060>
Call-ID: EXCEL-CSP0.78e.2158.430@192.168.1.102
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


5 -RECEIVED From 192.168.1.11:5060 at 2158
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1934922386e
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=2645473871
Contact: <sip:8623000@192.168.1.11:5060>
Call-ID: EXCEL-CSP0.78e.2158.430@192.168.1.102
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


X->H
    [00 45 00 e8 00 00 00 00 10 02 02 1e 09 00 01 00 39 00
   03 00
     08 01 03 00 33 00 2c 00 01 29 50 00 26 45 58 43 45 4c
     2d 43 53 50 30 2e 37 38 65 2e 32 31 35 38 2e 34 33 30
      40 31 39 32 2e 31 36 38 2e 31 2e 31 30 32 00]

X->H
    [00 11 00 43 00 0e 00 00 01 0d 03 00 08 01 00 a7 00 24
   00]
```

```
H->X
    [00 05 00 43 00 0e 00]

H->X
    [00 11 00 bf 00 00 ff 00 02 0d 03 00 00 05 0d 03 00 00
     05]


6 -SENT To 192.168.1.51:5060 at 2158
SIP/2.0 180 Ringing
To: <sip:8623000@192.168.1.102>;tag=315186d
From: "User ID"
   <sip:8623001@192.168.1.102>;tag=00036b095130002d3ac6a6
   6f-0a11923d
Call-ID: 00036b09-51300033-4fc0f34b-4c21a745@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK034df162
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [00 07 00 bf 00 00 00 00 10]


7 -RECEIVED From 192.168.1.11:5060 at 2159
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1934922386e
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=2645473871
Contact: <sip:8623000@192.168.1.11:5060>
Call-ID: EXCEL-CSP0.78e.2158.430@192.168.1.102
CSeq: 1 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 298

v=0
o=8623000 104287537 104288338 IN IP4 192.168.1.11
s=X-Lite
c=IN IP4 192.168.1.11
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
```

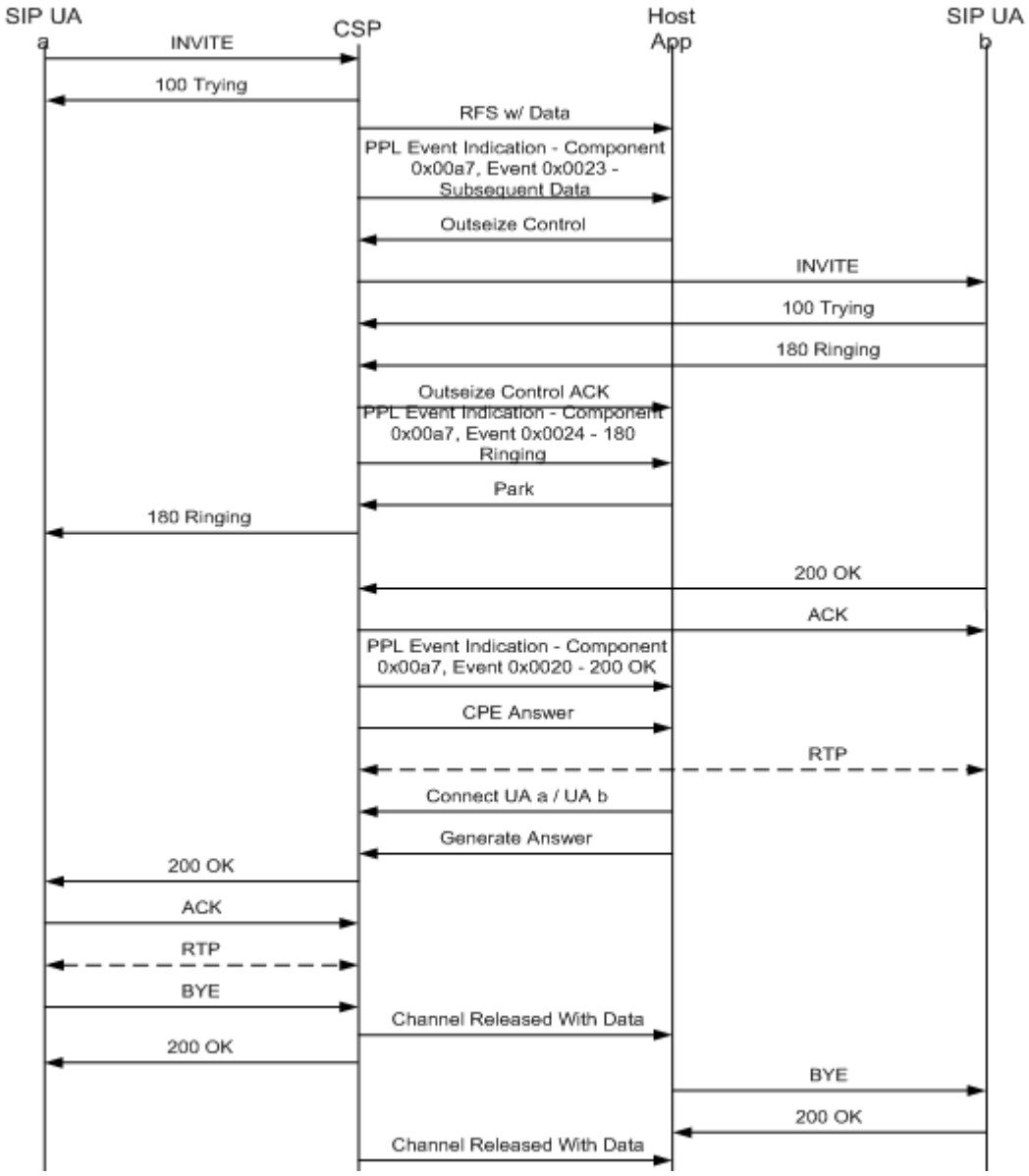
```
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15


8 -SENT To 192.168.1.11:5060 at 2159
ACK sip:8623000@192.168.1.11:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.11:5060>;tag=2645473871
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=1934922386e
Call-ID: EXCEL-CSP0.78e.2158.430@192.168.1.102
CSeq: 1 ACK
Content-Length: 0


X->H
    [00 2e 00 43 00 0f 00 00 01 0d 03 00 08 01 00 a7 00 20
   01 03
     00 33 00 18 00 03 27 94 00 04 c0 a8 01 0b 27 95 00 04
      00 00 1f 40 27 b0 00 02 00 02]

H->X
    [00 05 00 43 00 0f 00]

X->H
    [00 0d 00 2e 00 04 00 00 01 0d 03 00 08 01 20]

H->X
    [00 05 00 2e 00 04 00]

H->X
    [00 11 00 00 00 00 ff 00 02 0d 03 00 00 05 0d 03 00 08
     01]

X->H
    [00 07 00 00 00 00 00 00 10]

H->X
    [00 0d 00 ba 00 00 ff 00 01 0d 03 00 00 05 01]

X->H
    [00 07 00 ba 00 00 00 00 10]


9 -SENT To 192.168.1.51:5060 at 2159
SIP/2.0 200 OK
To: <sip:8623000@192.168.1.102>;tag=315186d
```

```
From: "User ID"
   <sip:8623001@192.168.1.102>;tag=00036b095130002d3ac6a6
   6f-0a11923d
Call-ID: 00036b09-51300033-4fc0f34b-4c21a745@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK034df162
User-Agent: Excel_CSP/83.10.57
Content-Type: application/sdp
Content-Length: 144

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 15996 RTP/AVP 0 101
a=rtpmap:101 telephone-event/8000


10-RECEIVED From 192.168.1.51:51035 at 2159
ACK sip:8623000@192.168.1.102:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK1e6721b2
From: "User ID"
   <sip:8623001@192.168.1.102>;tag=00036b095130002d3ac6a6
   6f-0a11923d
To: <sip:8623000@192.168.1.102>;tag=315186d
Call-ID: 00036b09-51300033-4fc0f34b-4c21a745@192.168.1.51
CSeq: 101 ACK
User-Agent: CSCO/7
Content-Length: 0


11-RECEIVED From 192.168.1.51:51035 at 2163
BYE sip:8623000@192.168.1.102:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK6dcd4cc5
From: "User ID"
   <sip:8623001@192.168.1.102>;tag=00036b095130002d3ac6a6
   6f-0a11923d
To: <sip:8623000@192.168.1.102>;tag=315186d
Call-ID: 00036b09-51300033-4fc0f34b-4c21a745@192.168.1.51
CSeq: 102 BYE
User-Agent: CSCO/7
Content-Length: 0

X->H
    [00 57 00 69 00 08 00 00 01 0d 03 00 00 05 02 02 1e 2a
   00 05
     01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 ba 01 11
```

```
     00 04 00 00 03 0b 01 10 00 04 00 00 74 40 01 12 00 04
     00 00 79 b8 03 00 33 00 18 00 03 27 4e 00 02 00 10 27
      92 00 04 c0 a8 01 83 27 93 00 04 00 00 3e 7c]

H->X
    [00 05 00 69 00 08 00]


12-SENT To 192.168.1.11:5060 at 2163
BYE sip:8623000@192.168.1.11:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=2645473871
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1934922386e
Call-ID: EXCEL-CSP0.78e.2158.430@192.168.1.102
CSeq: 2 BYE
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


13-SENT To 192.168.1.51:5060 at 2163
SIP/2.0 200 OK
To: <sip:8623000@192.168.1.102>;tag=315186d
From: "User ID"
   <sip:8623001@192.168.1.102>;tag=00036b095130002d3ac6a6
   6f-0a11923d
Call-ID: 00036b09-51300033-4fc0f34b-4c21a745@192.168.1.51
CSeq: 102 BYE
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK6dcd4cc5
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


14-RECEIVED From 192.168.1.11:5060 at 2163
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1934922386e
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=2645473871
Contact: <sip:8623000@192.168.1.11:5060>
Call-ID: EXCEL-CSP0.78e.2158.430@192.168.1.102
CSeq: 2 BYE
Server: X-Lite release 1103m
Content-Length: 0


X->H
    [00 57 00 69 00 09 00 00 01 0d 03 00 08 01 02 02 1e 2a
   00 05
```

```
01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 00 01 11
00 04 00 00 03 4f 01 10 00 04 00 00 00 00 01 12 00 04
00 00 84 58 03 00 33 00 18 00 03 27 4e 00 02 00 10 27
92 00 04 c0 a8 01 84 27 93 00 04 00 00 3d 78]

H->X
    [00 05 00 69 00 09 00]
```

Using *Outseize Control* message



Message Trace

```
1 -RECEIVED From 192.168.1.51:50606 at 666
INVITE sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK3fef366f
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c251500162fba92
   44-7689e187
To: <sip:8623000@192.168.1.102>
Call-ID: 00036b3c-25150014-4d8706ac-79e7e25f@192.168.1.51
CSeq: 101 INVITE
User-Agent: CSCO/7
Contact: <sip:8623001@192.168.1.51:5060>
Expires: 180
Content-Type: application/sdp
Content-Length: 246
Accept: application/sdp

v=0
o=Cisco-SIPUA 7880 24344 IN IP4 192.168.1.51
s=SIP Call
c=IN IP4 192.168.1.51
t=0 0
m=audio 32676 RTP/AVP 0 8 18 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15


2 -SENT To 192.168.1.51:5060 at 666
SIP/2.0 100 Trying
To: <sip:8623000@192.168.1.102>;tag=278629a
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c251500162fba92
   44-7689e187
Call-ID: 00036b3c-25150014-4d8706ac-79e7e25f@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK3fef366f
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [01 33 00 2d 00 02 00 00 01 0d 03 00 00 01 00 33 01 03
   00 33
     01 1f 00 18 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 38 36 32 33 30 30 30 00 29 1b 00 0e 31 39 32
```

```
     2e 31 36 38 2e 31 2e 31 30 32 00 29 1c 00 04 00 00 13
     c4 29 23 00 08 38 36 32 33 30 30 31 00 29 25 00 0e 31
     39 32 2e 31 36 38 2e 31 2e 31 30 32 00 29 26 00 04 00
     00 13 c4 29 2d 00 08 38 36 32 33 30 30 31 00 29 2f 00
     0d 31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 30 00 04
     00 00 13 c4 29 33 00 01 01 27 18 00 09 02 00 00 00 07
     86 23 00 10 27 17 00 07 02 00 07 86 23 00 00 27 94 00
     04 c0 a8 01 33 27 95 00 04 00 00 7f a4 27 b0 00 02 00
     02 27 b1 00 02 00 01 29 16 00 01 01 29 54 00 08 38 36
     32 33 30 30 30 00 29 55 00 0e 31 39 32 2e 31 36 38 2e
     31 2e 31 30 32 00 29 56 00 04 00 00 13 c4 29 50 00 31
     30 30 30 33 36 62 33 63 2d 32 35 31 35 30 30 31 34 2d
     34 64 38 37 30 36 61 63 2d 37 39 65 37 65 32 35 66 40
     31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 53 00 02 01
      00]

H->X
    [00 0c 00 2d 00 02 00 00 01 0d 03 00 00 01]

X->H
    [00 58 00 43 00 02 00 00 01 0d 03 00 00 01 00 a7 00 23
   01 03
     00 33 00 42 00 04 29 53 00 02 02 01 29 51 00 22 30 30
     30 33 36 62 33 63 32 35 31 35 30 30 31 36 32 66 62 61
     39 32 34 34 2d 37 36 38 39 65 31 38 37 00 29 52 00 08
      32 37 38 36 32 39 61 00 2a 0e 00 04 c0 a8 01 33]

H->X
    [00 05 00 43 00 02 00]

H->X
    [00 71 00 2c 00 00 ff 00 01 0d 03 00 00 00 01 03 00 33
     00 5f 00 09 27 7e 00 03 08 00 00 27 92 00 04 c0 a8 01
     83 27 93 00 04 00 00 20 00 27 b0 00 02 01 02 27 b1 00
     02 01 01 29 19 00 08 38 36 32 33 30 30 30 00 29 1b 00
     0d 31 39 32 2e 31 36 38 2e 31 2e 31 31 00 29 23 00 08
     38 36 32 33 30 30 31 00 29 25 00 0d 31 39 32 2e 31 36
      38 2e 31 2e 35 31 00]


3 -SENT To 192.168.1.11:5060 at 667
INVITE sip:8623000@192.168.1.11:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.11:5060>
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=1927207729b
Call-ID: EXCEL-CSP0.787.667.910@192.168.1.102
Contact: 8623001<sip:8623001@192.168.1.102:5060>
User-Agent: Excel_CSP/83.10.57
Supported: timer
```

```
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 104

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 8192 RTP/AVP 0


4 -RECEIVED From 192.168.1.11:5060 at 667
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1927207729b
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=3982133866
Contact: <sip:8623000@192.168.1.11:5060>
Call-ID: EXCEL-CSP0.787.667.910@192.168.1.102
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


5 -RECEIVED From 192.168.1.11:5060 at 668
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1927207729b
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=3982133866
Contact: <sip:8623000@192.168.1.11:5060>
Call-ID: EXCEL-CSP0.787.667.910@192.168.1.102
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


X->H
    [00 38 00 2c 00 00 00 00 10 01 03 00 33 00 2b 00 01 29
   50 00
     25 45 58 43 45 4c 2d 43 53 50 30 2e 37 38 37 2e 36 36
     37 2e 39 31 30 40 31 39 32 2e 31 36 38 2e 31 2e 31 30
      32 00]

X->H
```

```
    [00 11 00 43 00 03 00 00 01 0d 03 00 00 00 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 03 00]

H->X
    [00 11 00 bf 00 00 ff 00 02 0d 03 00 00 01 0d 03 00 00
     01]


6 -SENT To 192.168.1.51:5060 at 668
SIP/2.0 180 Ringing
To: <sip:8623000@192.168.1.102>;tag=278629a
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c251500162fba92
   44-7689e187
Call-ID: 00036b3c-25150014-4d8706ac-79e7e25f@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK3fef366f
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [00 07 00 bf 00 00 00 00 10]


7 -RECEIVED From 192.168.1.11:5060 at 669
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1927207729b
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=3982133866
Contact: <sip:8623000@192.168.1.11:5060>
Call-ID: EXCEL-CSP0.787.667.910@192.168.1.102
CSeq: 1 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 298

v=0
o=8623000 283848162 283850034 IN IP4 192.168.1.11
s=X-Lite
c=IN IP4 192.168.1.11
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
```

```
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15


8 -SENT To 192.168.1.11:5060 at 669
ACK sip:8623000@192.168.1.11:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.11:5060>;tag=3982133866
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=1927207729b
Call-ID: EXCEL-CSP0.787.667.910@192.168.1.102
CSeq: 1 ACK
Content-Length: 0


X->H
    [00 2e 00 43 00 04 00 00 01 0d 03 00 00 00 00 a7 00 20
   01 03
     00 33 00 18 00 03 27 94 00 04 c0 a8 01 0b 27 95 00 04
      00 00 1f 40 27 b0 00 02 00 02]

H->X
    [00 05 00 43 00 04 00]

X->H
    [00 0d 00 2e 00 00 00 00 01 0d 03 00 00 00 20]

H->X
    [00 05 00 2e 00 00 00]

H->X
    [00 05 00 2e 00 00 00]

H->X
    [00 11 00 00 00 00 ff 00 02 0d 03 00 00 01 0d 03 00 00
      00]

X->H
    [00 07 00 00 00 00 00 00 10]

H->X
    [00 0d 00 ba 00 00 ff 00 01 0d 03 00 00 01 01]

X->H
    [00 07 00 ba 00 00 00 00 10]
```

```
9 -SENT To 192.168.1.51:5060 at 670
SIP/2.0 200 OK
To: <sip:8623000@192.168.1.102>;tag=278629a
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c251500162fba92
   44-7689e187
Call-ID: 00036b3c-25150014-4d8706ac-79e7e25f@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK3fef366f
User-Agent: Excel_CSP/83.10.57
Content-Type: application/sdp
Content-Length: 144

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 15740 RTP/AVP 0 101
a=rtpmap:101 telephone-event/8000


10-RECEIVED From 192.168.1.51:50606 at 670
ACK sip:8623000@192.168.1.102:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK6b2d2bc4
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c251500162fba92
   44-7689e187
To: <sip:8623000@192.168.1.102>;tag=278629a
Call-ID: 00036b3c-25150014-4d8706ac-79e7e25f@192.168.1.51
CSeq: 101 ACK
User-Agent: CSCO/7
Content-Length: 0


11-RECEIVED From 192.168.1.51:50606 at 675
BYE sip:8623000@192.168.1.102:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK16c337d6
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c251500162fba92
   44-7689e187
To: <sip:8623000@192.168.1.102>;tag=278629a
Call-ID: 00036b3c-25150014-4d8706ac-79e7e25f@192.168.1.51
CSeq: 102 BYE
User-Agent: CSCO/7
Content-Length: 0
```

```
X->H
    [00 57 00 69 00 00 00 00 01 0d 03 00 00 01 02 02 1e 2a
   00 05
     01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 eb 01 11
     00 04 00 00 03 ce 01 10 00 04 00 00 92 e0 01 12 00 04
     00 00 98 30 03 00 33 00 18 00 03 27 4e 00 02 00 10 27
      92 00 04 c0 a8 01 83 27 93 00 04 00 00 3d 7c]


12-SENT To 192.168.1.11:5060 at 675
BYE sip:8623000@192.168.1.11:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=3982133866
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1927207729b
Call-ID: EXCEL-CSP0.787.667.910@192.168.1.102
CSeq: 2 BYE
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


13-SENT To 192.168.1.51:5060 at 675
SIP/2.0 200 OK
To: <sip:8623000@192.168.1.102>;tag=278629a
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c251500162fba92
   44-7689e187
Call-ID: 00036b3c-25150014-4d8706ac-79e7e25f@192.168.1.51
CSeq: 102 BYE
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK16c337d6
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


H->X
    [00 05 00 69 00 00 00]


14-RECEIVED From 192.168.1.11:5060 at 675
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1927207729b
To: 8623000
   <sip:8623000@192.168.1.11:5060>;tag=3982133866
Contact: <sip:8623000@192.168.1.11:5060>
Call-ID: EXCEL-CSP0.787.667.910@192.168.1.102
CSeq: 2 BYE
Server: X-Lite release 1103m
```

```
Content-Length: 0


X->H
    [00 57 00 69 00 01 00 00 01 0d 03 00 00 00 02 02 1e 2a
   00 05
     01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 00 01 11
     00 04 00 00 04 34 01 10 00 04 00 00 00 00 01 12 00 04
     00 00 a8 20 03 00 33 00 18 00 03 27 4e 00 02 00 10 27
      92 00 04 c0 a8 01 83 27 93 00 04 00 00 20 00]

H->X
    [00 05 00 69 00 01 00]
```

Common SIP Functions and Call Features

This section describes several common SIP functions and call features including:

- *Registration Overview (5-49)*
- *SIP Redundancy (5-53)*
- *SIP Support for TCP (5-57)*
- *Reporting/Generating SIP Headers or Parameters (5-61)*
- *Reporting/Generating SIP Responses (5-101)*

Registration Overview

This section provides an overview of registration including the following features:

- *Host Controlled SIP Registration Failure Response Code (5-50)*
- *SIP Registration Duration Check (5-52)*

Host Controlled SIP Registration Failure Response Code

Overview

This feature allows host application developers to control the SIP response code when the CSP rejects incoming registration requests.

Prior to the release of this feature, the SIP stack sent the hard-coded 503 response code when rejecting an incoming registration.

Description

The host application developer provides a final response code that the SIP stack will send to reject a SIP REGISTER message. The PPL Event request type Reject Registration contains the SIP Response Code TLV (0x2915).

The response can be code types 4xx, 5xx, or 6xx. Refer to the *Responses (B-7)* section.

This feature is meant to reject and not authenticate registration requests.

In order to authenticate a registration request, send a 401/407 response code in the *PPL Event Request* message (Authenticate Request).

Configuring

If you want to reject a registration with a 423 response (Interval Too Brief) use the *VoIP Protocol Configure* (0x00EE) message with the Minimum Registration Duration (0x0283) TLV. Refer to *Host Control Method of SIP 180 Provisional Response Generation (5-166)*.

For all other rejections use the *PPL Event Request* message with SIP Response Code TLV (0x2915).

Example:

Below is the PPL event indication sent to the host application when a REGISTER is received. The following PPL event request is the rejection (cause code 403 which is value 0x0193 in SIP Response Code TLV (0x2915).

Following the API is the SIP messaging.

API Message

```
X->H
    [00 8e 00 43 00 00 00 00 01 7f 04 ff 00 2d 09 00 a7 00 0a 01
     03 00 33 00 77 00 0b 29 19 00 05 38 35 36 36 00 29 1b
     00 0e 31 39 32 2e 31 36 38 2e 31 2e 31 30 31 00 29 1c
     00 04 00 00 13 c4 29 23 00 05 38 35 36 36 00 29 25 00
     0e 31 39 32 2e 31 36 38 2e 31 2e 31 30 31 00 29 26 00
```

```
     04 00 00 13 c4 29 2d 00 05 38 35 36 36 00 29 2f 00 0d
     31 39 32 2e 31 36 38 2e 31 2e 35 30 00 29 30 00 04 00
     00 13 c4 29 33 00 01 01 29 31 00 04 00 00 0e 10]

H->X
    [00 05 00 43 00 00 00]

H->X
    [00 05 00 43 00 00 00]

H->X
    [00 1f 00 44 00 00 ff 00 01 7f 04 ff 00 2d 09 00 a7 00
     0b 01 03 00 33 00 08 00 01 29 15 00 02 01 93]

X->H
    [00 07 00 44 00 00 00 00 10]

2
```

SIP Messaging

```
1 -RECEIVED From 192.168.1.50:1071 at 915
REGISTER sip:192.168.1.101 SIP/2.0
From: sip:8566@192.168.1.101;tag=6551c6551
To: sip:8566@192.168.1.101
Call-Id: e8b903dc85bc74c9af9a27e10dbd1030
Cseq: 113 REGISTER
Contact:
   <sip:8566@192.168.1.50;LINEID=c97ec9224a2f2cf6c6238268208227
   91>
Expires: 3600
Date: Mon, 26 Jul 2004 12:36:12 GMT
Accept-Language: en
Supported: sip-cc, sip-cc-01, timer, replaces
User-Agent: Pingtel/2.1.11 (VxWorks)
Content-Length: 0
Via: SIP/2.0/UDP 192.168.1.50


2 -SENT To 192.168.1.50:5060 at 915
SIP/2.0 100 Trying
To: sip:8566@192.168.1.101;tag=9150
From: sip:8566@192.168.1.101;tag=6551c6551
Call-ID: e8b903dc85bc74c9af9a27e10dbd1030
CSeq: 113 REGISTER
Via: SIP/2.0/UDP 192.168.1.50
User-Agent: Excel_CSP/82.30.112
Content-Length: 0


3 -SENT To 192.168.1.50:5060 at 915
SIP/2.0 403 Forbidden
To: sip:8566@192.168.1.101;tag=9150
From: sip:8566@192.168.1.101;tag=6551c6551
Call-ID: e8b903dc85bc74c9af9a27e10dbd1030
CSeq: 113 REGISTER
Via: SIP/2.0/UDP 192.168.1.50
User-Agent: Excel_CSP/82.30.112
Content-Length: 0
```

SIP Registration Duration Check

Overview

As a SIP Registrar, the CSP SIP stack is subject to SIP Register requests from a diverse set of endpoints. Some of these endpoints might try to refresh registrations frequently enough to cause processor and memory overload conditions on the CSP Matrix Series 3 Card card or host application.

SIP specification RFC 2543 recommend that an endpoint that tries to register for an expiration interval less than the Registrar's configurable minimum threshold can be denied with a SIP 423 response (Interval Too Brief).

Prior to this feature, the SIP stack acted as a pass-through for all REGISTER requests. Because of this scenario, the host application was involved with the PPL Event Request handshake to generate the SIP 423 response.

Pertinent Specification

RFC 2543

Description

This feature allows you to set a minimum registration duration threshold in the SIP stack configuration. The SIP REGISTER message that requests a lower expiration duration than the minimum duration configured is automatically rejected by the CSP with the SIP 423 response code.

The SIP 423 response generated by the CSP contains a Min-Expires header field filled with the lowest registration duration acceptable to the registrar. By being able to configure the minimum registration duration threshold, you eliminate potential processor and memory overload conditions.

Configuring

This feature is enabled (that is, a check for minimum registration duration is made) only when you include the Minimum Registration Duration (0x0283) TLV in the *VoIP Protocol Configure* (0x00EE) message. Once enabled, to disable this feature, set the duration to a significantly low value.

Refer to the *API Reference*.

Querying

You can query the Registration Minimum Duration (0x0283) with the *VoIP Protocol Query* (0x00EF) message.

SIP Redundancy

This section describes SIP Redundancy including the following two features:

- *Dual Ethernet Port (5-54)*
- *SIP User Agent Redundancy (5-55)*

Dual Ethernet Port

Overview

You can configure the second Ethernet port on the CSP Matrix Series 3 Card I/O card to separate the SIP signaling traffic from the host application control traffic. Dialogic recommends that you have the host-to-CSP traffic carried on Ethernet port A and the SIP signaling traffic carried on the Ethernet port B.

The BOOTP server is required to configure Ethernet port B.

Description

The additional information is carried in the vendor specific area of the BOOTP response. You modify the BOOTP configuration to include new entries to carry the IP Address, Gateway IP Address, and Subnet Mask for the second physical Ethernet port.

Configuring the second Ethernet port is optional but if you enter an IP Address for Ethernet port B you must enter an associated Subnet Mask.

> **Important!** Although a gateway IP Address may be configured for both Port A and Port B, only one is utilized. If two gateway IP addresses are configured, one for Port A and one for Port B the one specified for Port A will be used. Therefore, Dialogic recommends that you configure only one gateway IP Address and that it be on Port B.

If you do not configure the second Ethernet port, it is internally set to the same values configured for the first Ethernet port.

Configuring

For the associated procedure, refer to *Configuring Dual Ethernet Ports on IP Signaling and CSP Matrix Series 3 Cards* in the *Application Development* chapter in the *Developer Guide: Overview.*

SIP User Agent Redundancy

Overview

The SIP software supports redundancy allowing the SIP UA to continue functioning after a CSP Matrix Series 3 Card switchover.

In an CSP with redundant CSP Matrix Series 3 Cards, the cards themselves are physically identical.

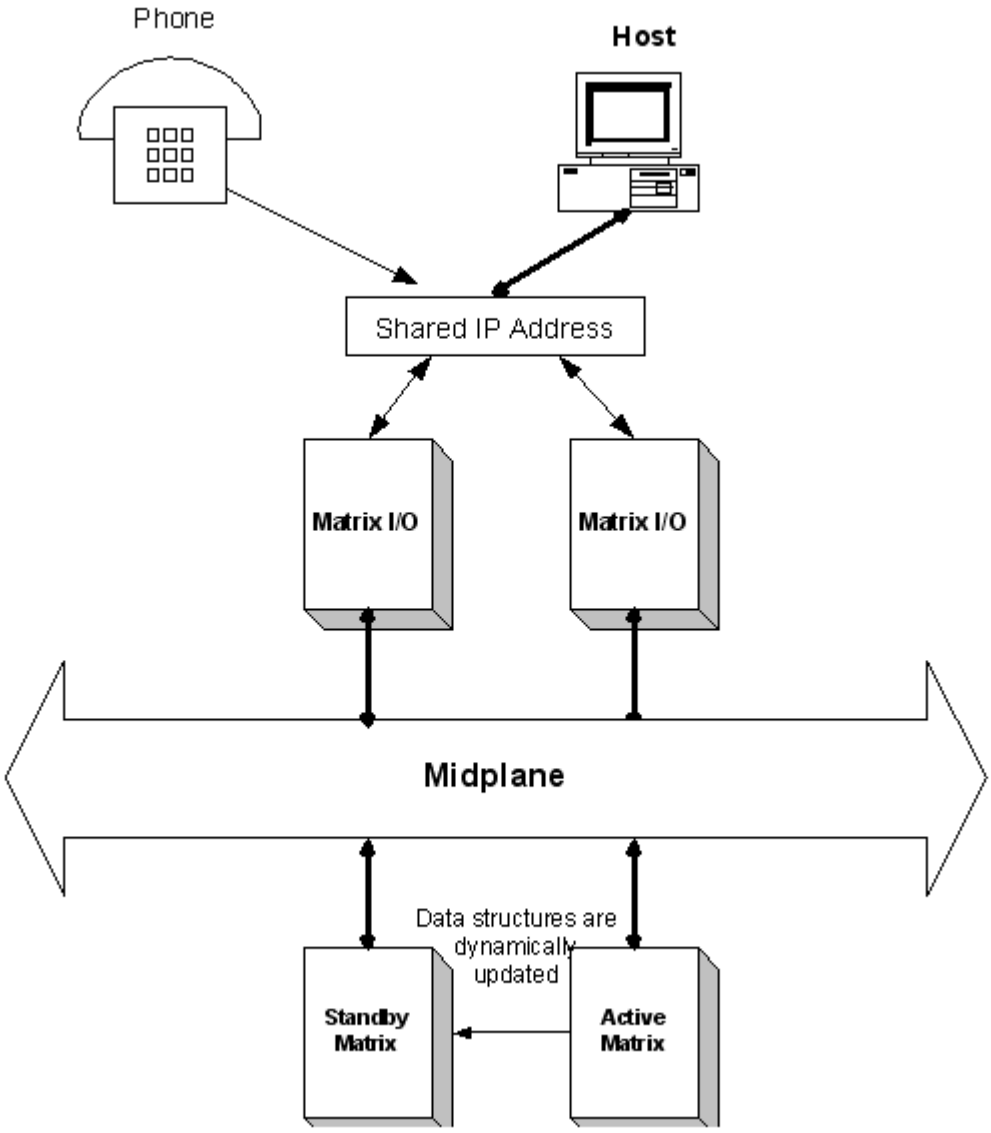
> **Important!** To enable SIP UA Redundancy, you must configure a shared IP address. The shared IP address is for redundancy only. All API message should be sent directly to the active matrix card's IP address - not to the shared IP address.

Description

All communication to the CSP Matrix Series 3 Cards are over this one shared IP address. You create this shared IP configuration using DHCP or BOOTP. See *Redundant CSP Matrix Series 3 Card* in the *API Developer's Guide: Overview* for general information about redundancy on the CSP Matrix Series 3 Card.

The two matrix cards communicate with each other over a High-Level Data Link Control (HDLC) link. The SIP software data structures are dynamically updated from the active CSP Matrix Series 3 Card to the standby CSP Matrix Series 3 Card on this midplane bus so calls in the connected or answered state are maintained during a switchover. Calls in transition are dropped.

Figure 5-4 SIP UA Redundancy



Configuring

See either of the following two procedures in the *Application Development* chapter of the *Developer's Guide: Overview* to set up the shared IP address:

Configuring a Shared IP Address Using BOOTP Server

Configuring a Shared IP Address Using DHCP Server

SIP Support for TCP

Overview

This feature allows SIP signaling to be reliably transported using the Transmission Control Protocol (TCP). TCP is a transport layer, connection-oriented, end-to-end protocol. It provides reliable, sequenced, and unduplicated bytes to a remote or local user.

This feature adds to the CSP's current support of SIP signaling transport using User Datagram Protocol (UDP) transport layer.

This section explains how the CSP supports SIP using TCP, and the different scenarios involved.

Pertinent Specification

With this feature, the CSP is compliant with the transport aspects of SIP RFC 3261.

Description

Outbound SIP Calls - SIP User Agent Client Registered with the CSP

Whenever a SIP User Agent Client registers with the CSP, the Registration message specifies the desired transport type for accessing that User Agent Client. This registered transport type takes precedence over all other mechanisms to set up the outbound transport type.

The initial Invite message that leaves the CSP travels over the transport type specified in the Registration's Contact header specification. Subsequent transactions within this SIP session are allowed to switch the transport type as required.

> **Important!** If the transport type is TCP and the SIP session fails to get established, the CSP will not retry the Invite message using the UDP transport.

Outbound SIP Calls - Default Outbound Proxy Specified

You enable the SIP mode with a *VoIP Protocol Configure* message that specifies the Use Default SIP Proxy Host TLV (0x0264).

When the SIP User Agent Client is not registered with the CSP, the Invite message is sent to the SIP proxy specified in the Use Default SIP Proxy Host TLV. In this case, the CSP must select a transport type to use to send the Invite message. The default outbound transport type is UDP. You can change the default transport by sending a *VoIP Protocol*

Configure message containing the SIP Transport Type TLV (0x027D). This message sets the initial transport type for all messages originating from the CSP to the outbound SIP proxy.

If you want the host application controlling the CSP to change the transport type for individual calls, include the SIP B Leg Transport Type TLV in the *Route Control* message (in the NPDI ICB 0x0033).

This TLV overrides the current setting of the outbound transport type for this call. Subsequent calls that do not contain this TLV uses the current setting of the outbound transport type

Important! The following explains what happens when connected CSPs have different default transport types.

Platform 1 is configured to send UDP. Platform 2 is configured to send TCP.
Platform 1 sends an Invite message to Platform 2 using UDP. If Platform 2 has to send a Re-invite message back to the Platform 1, the message uses UDP since that was the original transport type used by Platform 1 in the Invite message.

Inbound SIP Calls

The originator of the message selects the transport type of the inbound call. Whenever an Invite message arrives, the CSP sends a *Request for Service with Data* (0x002D) message to the host application. This message contains the SIP A Leg Transport Type TLV (0x2916) (in the NPDI ICB 0x0033) which specifies the transport type used in the Invite message.

If this TLV specifies that the TCP transport type is used, two additional TLVs appear in the *Request for Service Data (0x002D)* message (in the NPDI ICB 0x0033):

- SIP Remote IP Address (0x294E)
- SIP Remote IP Port (0x294F)

These TLVs are used in inbound SIP sessions as described in Inbound SIP Sessions Over a Persistent TCP Socket below.

Persistent Sockets

You can enable persistent sockets for Inbound and Outbound SIP sessions. Persistent sockets prevent the SIP Idle Timeout from expiring and closing the socket.

SIP Idle Timeout

If you do not use persistent sockets, each TCP socket expires when the SIP Idle Timeout is reached. The default is 32 seconds. The host application can change the default by using the *VoIP Protocol Configure (*0x00EE*)* message and sending a SIP Idle Socket Timeout TLV (0x0281) with a new value. The minimum time is five (5) seconds.

Outbound SIP Sessions Over a Persistent TCP Socket

You can enable this mode two ways:

- If you want all TCP sockets for all outbound SIP sessions to be over a persistent TCP socket, enable this mode with the SIP Persistent Sockets TLV (0x027E) within a *VoIP Protocol Configuration* (0x00EE) message.
- If you want to have a mix of persistent and non-persistent TCP sockets, include the SIP Do Not Close Socket TLV (0x294D) in the *Route Control (*0x00E8*)* message used to initiate the Invite message. Do not issue the SIP Persistent Sockets TLV.

The CSP does not report the remote IP and port address to the host application because the host application already supplied this information when placing the outbound call.

Important! Nothing prevents the peer endpoint from closing this socket, unless the endpoint is an CSP.

Inbound SIP Sessions Over a Persistent TCP Socket

After the host application receives a *Request for Service with Data* (0x002D) message with the following TLVs (in the NPDI ICB 0x0033) the host application determines if it will make the TCP socket persistent:

- SIP Remote IP Address (0x294E)
- SIP Remote IP Port (0x294F)

If the host application determines that the TCP socket should be persistent, the host application issues a *PPL Event Request* (0x0044) message with an Event ID of 0x29, which keeps the socket from closing.

Reusing TCP Sockets

You can enable the reuse of TCP sockets. This feature:

- must be enabled when persistent sockets are enabled.
- can be used when persistent sockets are disabled.

To enable this feature, send the *VoIP Protocol Configure* (0x00EE) message containing the SIP Existing Socket Reuse TLV (0x0280).

When you enable this feature and a SIP message needs to be transported from the CSP, the CSP checks to determine if a TCP socket to the destination is currently open. If one is open, then that socket is used to transport the message.

Configuring

Refer to the following to configure TCP sockets:

API Messages in the *API Reference*

- *VoIP Protocol Configure (0x00EE)*
- *PPL Event Request (0x0044)*
- *Request for Service with Data (0x002D)*
- *Route Control (0x00E8)*

PPL Event Requests

- Do Not Close Socket - Event ID 0x29

(Refer to Call Control Event Requests in PPL Information: *PPL Information: SIP UA 0x00A7 (5-14)*

TLVs

Refer to the *API Reference*.

- SIP Transport Type (0x027D)
- SIP A Leg Transport Type (0x2916)
- SIP B Leg Transport Type (0x2917)
- SIP Do Not Close Socket (0x29D4)
- SIP Remote IP Address (0x294E)
- SIP Remote IP Port (0x294F)
- SIP Persistent Sockets (0x027E)
- SIP Existing Socket Reuse (0x0280)
- SIP Idle Socket Timeout (0x0281)

Reporting/Generating SIP Headers or Parameters

The section describes the following features that report or generate SIP headers or parameters.

- *Host Notification of Selective SIP Header Information (5-62)*
- *Via Heading Reporting (5-63)*
- *Call-ID Reporting for Outbound SIP Calls (5-70)*
- *Subject Header Field Access (5-71)*
- *SIP Display Name Parameter in From Header (5-83)*
- *Access To Contact Header (5-85)*
- *SIP Access To Parameters in To Header (5-87)*
- *Report and Control of P-Asserted and P-Access Network Info Headers (5-91)*
- *SIP Remote Party ID and RPID Privacy for Outbound Calls (5-94)*
- *Remote Party ID (5-99)*

Host Notification of Selective SIP Header Information

Overview

With this feature, the CSP can send selective SIP header data to the host application in the *Request For Service with Data* message and, if required, in the *PPL Event Indication* message. The Call-ID header information allows the host application to uniquely identify the call with other elements in the SIP network.

Important! You should disable all additional SIP header notification if calls are being internally routed without host application intervention.

Description

The process works as follows:

1. The host application specifies what additional fields from the SIP message header that it wants to see in the *Request For Service with Data* message.
2. You specify the fields in the 0x027F SIP Message Information Mask in the *VoIP Protocol Configure* message.
3. If the CSP can fit all of the data in the *Request For Service with Data* message, it does so. (The data is formatted in the NPDI TLVs listed below these steps.)
4. If the size exceeds the maximum size that the CSP allows (320 bytes of NPDI data), the CSP sends the following:
   - the *Request For Service with Data* message with NPDI data (not to exceed 320 bytes) including the Subsequent Info Status 0x2953 TLV indicating that there is more data to follow.
   - The CSP sends “n number” of *PPL Event Indications* messages where each indication would carry NPDI data (not to exceed 320 bytes) including the Subsequent Info Status 0x2953 TLV
5. If the CSP sends the data in multiple messages (*Request for Service* and subsequent *PPL Event Indication*) the CSP adds the Subsequent Info Status 0x2953 TLV. This TLV indicates the sequence number of the message and whether it is the last message in the sequence.
6. If the data has to go into an API message, and not into TLVs, the call is rejected with the “500 Internal Server Error” response.

API Messages Used

- *Request For Service with Data* (0x002D)
- *PPL Event Indications* (0x0043)

Via Heading Reporting

Overview This feature reports to the host application the IP address and Port Number of the top most Via header field of the inbound SIP sessions.

Pertinent Specification RFC 2543/3261

Description The information in the Via header could be of the last gateway or router to handle the call.

- The host application can route the outbound SIP session based on the reported Via headers.
- The Via head is reported for the session-establishing the INVITE. It is not reported for session-modifying INVITEs (Re-INVITES).
- Only hostnames with length up to 80 bytes are reported. If an INVITE message with hostname length greater than 80 bytes is obtained then the SIP Header Field TLV (0x299C) itself will not be reported. But the other TLVs, SIP Header Field Container and SIP Header Field will be reported.
- If port number is absent or if it is NULL then the SIP Port Number TLV will not be reported.
- If port number is equal to or greater than 2147483647 then it will be reported as 7f ff ff ff.

SIP INVITE RFS with Data

The reporting of the Via header field is within the context of the SIP INVITE and corresponding *Request for Service with Data* (0x002D) messages to the host application as follows:

- The SIP stack reports the Hostname in the Via header field present in the inbound SIP INVITE message in raw null-terminated ASCII format and the Port Number as a variable.
- The NPDI TLV, SIP Header Field Container (0x299A) is used in the *Request for Service with Data* message or in the *Subsequent Data PPL Event Indication* (0x0043) for reporting purposes.

The SIP Header Field Container TLV is used with the NPDI Universal ICB (0x0033).

The NPDI TLV, SIP Header Field ID (0x299B) is used within the Header Field Container TLV and indicates which Header Field is reported - in this case the Via header (0x0034).

Another NPDI TLV, SIP Header Field (0x299C) is also used within the Header Field Container TLV will contain the reported data.

The following two NPDI TLVs are used within the SIP Header Field TLV to report the Via Parameters:

- SIP Host Name 0x295E
- SIP Port Number 0x295F

Example

The following example shows how the TLVs contain the Via header data:

```
29 9A 00 21
               - ID -
  29 9B 00 02 00 34
  29 9C 00 17
                ----------------Hostname-------
    29 5E 00 0B 31 30 2E 31 30 2E 31 2E 31 34 00
                --Port Num--
    29 5f 00 04 00 00 13 C4
```

Process

The Via header is reported as follows:

1. Enable Via header reporting as described in the *Configuring (5-64)* section.
2. When the CSP receives a SIP INVITE message, it sends a SIP Header Field Container TLV (0x299A) to the host application in the *Request for Service with Data* (0x002D) message.
3. If this message exceeds 320 bytes, the TLV goes in subsequent PPL Event Indications.

API Messages Used

- *Request for Service with Data* (0x002D)
- *PPL Event Indication* (0x0043)

Configuring

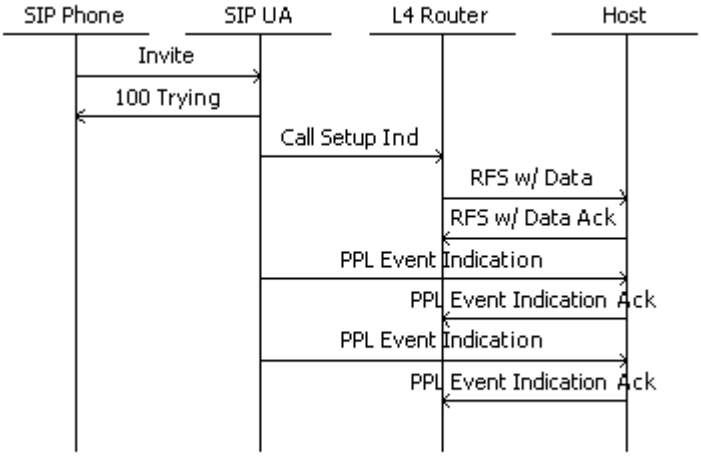
SIP Stack Configuration

Reporting the Via header field is disabled by default. To enable, set a bit mask during the SIP stack configuration.

The existing SIP Message Information Mask TLV (0x027F) is used in the *VoIP Protocol Configure* (0x00EE) message. Set bit 10.

Call Flow

The *Request for Service with Data* (0x002D) message or the PPL Event Indication in the diagram below report the Via Header Field to the host application.



Sample Messages

API Messages

```
X->H
    [00 e0 00 2d 00 0d 11 00 01 0d 03 00 36 0e 00 33 01 03
   00 33
     00 cc 00 13 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 54 65 73 74 69 6e 67 00 29 1b 00 0c 31 30 2e
     31 30 2e 31 2e 31 33 33 00 29 1c 00 04 00 00 13 c4 29
     23 00 06 33 35 37 35 31 00 29 25 00 0b 31 30 2e 31 30
     2e 31 2e 31 34 00 29 26 00 04 00 00 13 c5 29 2d 00 06
     33 35 37 35 31 00 29 2f 00 0b 31 30 2e 31 30 2e 31 2e
     31 34 00 29 30 00 04 00 00 13 c5 29 33 00 01 01 27 18
     00 08 02 00 00 00 05 35 75 10 27 94 00 04 0a 0a 01 0e
     27 95 00 04 00 00 1f 40 27 b0 00 02 00 02 27 b1 00 02
     00 01 29 16 00 01 01 29 9a 00 21 29 9b 00 02 00 22 29
     9c 00 17 29 5e 00 0b 31 30 2e 31 30 2e 31 2e 31 34 00
     29 5f 00 04 00 00 13 c5]
```

```
H->X
    [00 0c 00 2d 00 0d 11 00 01 0d 03 00 36 0e]

H->X
    [00 0d 00 ba 00 00 11 00 01 0d 03 00 36 0e 01]

X->H
    [00 07 00 ba 00 00 11 00 10]

H->X
    [00 11 00 08 00 00 11 00 02 0d 03 00 36 0e 0d 03 00 36
     0e]

X->H
    [00 07 00 08 00 00 11 00 10]

X->H
    [00 57 00 69 00 00 11 00 01 0d 03 00 36 0e 02 02 1e 2a
   00 05
     01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 00 01 11
     00 04 00 00 00 00 01 10 00 04 00 00 00 00 01 12 00 04
     00 00 00 00 03 00 33 00 18 00 03 27 4e 00 02 00 10 27
     92 00 04 0a 0a 01 7c 27 93 00 04 00 00 2a 9c]

H->X
    [00 05 00 69 00 00 11]
```

SIP Messages

```
14-RECEIVED From 10.10.1.14:5061 at 4014
INVITE sip:Testing@10.10.1.133 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.14:5061;rport;branch=z9hG4bK93EC4CE485054D9491
   95EB8C0F6
9D88A
From: Naina <sip:35751@10.10.1.14:5061>;tag=4130294919
To: <sip:Testing@10.10.1.133>
Contact: <sip:35751@10.10.1.14:5061>
Call-ID: AC811980-1464-41EB-A437-6376C80AEC79@10.10.1.14
CSeq: 57304 INVITE
Max-Forwards: 70
Content-Type: application/sdp
User-Agent: X-Lite release 1103m
Content-Length: 288

v=0
o=35751 5679296 5679312 IN IP4 10.10.1.14
s=X-Lite
```

```
c=IN IP4 10.10.1.14
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

15-SENT To 10.10.1.14:5061 at 4014
SIP/2.0 100 Trying
To: <sip:Testing@10.10.1.133>;tag=6716fae
From: Naina <sip:35751@10.10.1.14:5061>;tag=4130294919
Call-ID: AC811980-1464-41EB-A437-6376C80AEC79@10.10.1.14
CSeq: 57304 INVITE
Contact: Testing<sip:Testing@10.10.1.133:5060>
Via: SIP/2.0/UDP
   10.10.1.14:5061;rport;branch=z9hG4bK93EC4CE485054D9491
   95EB8C0F6
9D88A
User-Agent: Excel_CSP/83.10.161
Content-Length: 0

16-SENT To 10.10.1.14:5061 at 4015
SIP/2.0 180 Ringing
To: <sip:Testing@10.10.1.133>;tag=6716fae
From: Naina <sip:35751@10.10.1.14:5061>;tag=4130294919
Call-ID: AC811980-1464-41EB-A437-6376C80AEC79@10.10.1.14
CSeq: 57304 INVITE
Contact: Testing<sip:Testing@10.10.1.133:5060>
Via: SIP/2.0/UDP
   10.10.1.14:5061;rport;branch=z9hG4bK93EC4CE485054D9491
   95EB8C0F6
9D88A
User-Agent: Excel_CSP/83.10.161
Content-Length: 0


17-SENT To 10.10.1.14:5061 at 4015
SIP/2.0 200 OK
To: <sip:Testing@10.10.1.133>;tag=6716fae
From: Naina <sip:35751@10.10.1.14:5061>;tag=4130294919
Call-ID: AC811980-1464-41EB-A437-6376C80AEC79@10.10.1.14
CSeq: 57304 INVITE
Contact: Testing<sip:Testing@10.10.1.133:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
```

```
Via: SIP/2.0/UDP
   10.10.1.14:5061;rport;branch=z9hG4bK93EC4CE485054D9491
   95EB8C0F6
9D88A
User-Agent: Excel_CSP/83.10.161
Content-Type: application/sdp
Content-Length: 140

v=0
o=sip 0 0 IN IP4 10.10.1.133
s=SIP_Call
c=IN IP4 10.10.1.124
t=0 0
m=audio 10908 RTP/AVP 0 101
a=rtpmap:101 telephone-event/8000

18-RECEIVED From 10.10.1.14:5061 at 4015
ACK sip:Testing@10.10.1.133:5060 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.14:5061;rport;branch=z9hG4bK85AFB31A086E4612B6
   C99389623
9CF01
From: Naina <sip:35751@10.10.1.14:5061>;tag=4130294919
To: <sip:Testing@10.10.1.133>;tag=6716fae
Contact: <sip:35751@10.10.1.14:5061>
Call-ID: AC811980-1464-41EB-A437-6376C80AEC79@10.10.1.14
CSeq: 57304 ACK
Max-Forwards: 70
Content-Length: 0


19-RECEIVED From 10.10.1.14:5061 at 4018


20-SENT To 10.10.1.14:5061 at 4021
BYE sip:35751@10.10.1.14:5061 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.133
To: Naina <sip:35751@10.10.1.14:5061>;tag=4130294919
From: <sip:Testing@10.10.1.133>;tag=6716fae
Call-ID: AC811980-1464-41EB-A437-6376C80AEC79@10.10.1.14
CSeq: 57305 BYE
User-Agent: Excel_CSP/83.10.161
Content-Length: 0

21-RECEIVED From 10.10.1.14:5061 at 4021
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.133
From: <sip:Testing@10.10.1.133>;tag=6716fae
To: Naina <sip:35751@10.10.1.14:5061>;tag=4130294919
Contact: <sip:35751@10.10.1.14:5061>
```

```
Call-ID: AC811980-1464-41EB-A437-6376C80AEC79@10.10.1.14
CSeq: 57305 BYE
Server: X-Lite release 1103m
Content-Length: 0
```

Call-ID Reporting for Outbound SIP Calls

Overview

This feature allows host application developers to solicit Call-ID information for outbound SIP calls.

Pertinent Specification

RFC 2543/3261

Description

The SIP Call-ID is a globally unique session identifier that you can use for several purposes, including call logging and billing correlation.

The CSP SIP software reports Call-ID information for inbound calls. With this feature, the same capability is provided for outbound calls.

API Messages

- *Route Control* (0x00E8)
- *Outseize Control* (0x002C)

Configuring

It is optional to report the Call-ID for an outbound SIP call. You use bit number 16 in the SIP Message Information Mask TLV (0x027F) to enable or disable the information reporting.

If this bit is set in the SIP software configuration, the software will report the Call-ID information in the ACK response to the *Route Control* and *Outseize Control* messages. The information is contained in the SIP Call ID TLV (0x2950).

Sample Traces

The following are message traces for ACK responses to the *Route Control* (0x00E8) and *Outseize Control* (0x002C) messages.

The Call-ID information is contained in the SIP Call-ID TLV 0x2950 as highlighted below.

Route Control ACK

```
00 46 00 e8 00 00 ff 00 10 02 02 1e 09 00 01 00 39 00 03 00
01 02 03 00 33 00 2d 00 01 29 50 00 27 45 58 43 45 4c 2d 43
53 50 32 35 35 2e 36 31 30 2e 35 30 31 35 38 30 2e 37 36 30
40 31 30 2e 31 30 2e 31 2e 33 32 00
```

Outseize Control ACK

```
00 3a 00 2c 00 00 ff 00 10 01 03 00 33 00 2d 00 01 29 50 00
27 45 58 43 45 4c 2d 43 53 50 32 35 35 2e 36 30 38 2e 35 30
31 38 35 34 2e 39 39 30 40 31 30 2e 31 30 2e 31 2e 33 32 00
```

Subject Header Field Access

Overview

This feature allows host application developers to provide read and write access to the Subject header field.

Read access allows the host application to receive the content of the Subject header-field if it is present in the inbound INVITE message. In the SIP stack on the CSP, you can configure whether you want to report the Subject header field.

Write access allows the host application to fill information into the Subject header field of the outbound INVITE message.

> **Important!** Reading or writing the Subject header field works only if the subject field is less than 255 bytes.

Pertinent Specification

RFC 2543/3261

Description

Subject Header Field - Read Access

This section describes the Read Access portion of this feature.

The SIP software reports the Subject header field if it is present in the inbound INVITE message - in a raw, null-terminated ASCII format from the CSP to the host application.

The reporting provided with this feature is within the context of the SIP INVITE message and corresponding *Request for Service with Data* (0x002D) messages only.

The Subject header field is reported using NPDI TLV, the SIP Subject TLV (0x295B) in *Request for Service with Data* (0x002D) message or in *Subsequent Data - PPL Event Indication* message. The SIP Subject TLV will be used within the NPDI Universal ICB (0x0033).

Process

The following assumes that the Subject header-field reporting is turned on:

When the CSP receives a SIP INVITE that includes the Subject header-field, the CSP sends the SIP Subject TLV (0x295B) to the host application in *Request for Service with Data* message or in one of the subsequent PPL Event Indications (in case the *Request for Service with Data* exceeds 320 bytes).

The SIP Subject TLV, as any other TLVs except Subsequent Information Status TLV (0x2953), will be fully present in any one of the API messages.

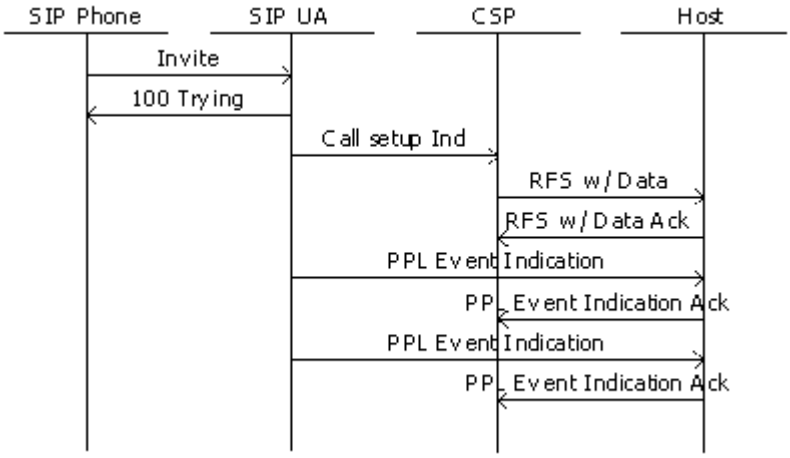
Call Flow

The *Request for Service with Data* message and the *PPL Event Indications* message shown in the call flow below include the Subsequent Information Status TLV.

The SIP Subject TLV will be present in any one of the messages from the CSP to the host application provided the follows is in place:

- reporting is turned on
- the inbound INVITE message has a Subject header

Figure 5-5 Read Access



Subject Header Field - Write Access

This section describes the Write Access portion of this feature. Refer to the *API Reference* for the specific formats of the API changes to support this functionality.

The write access is within the context of SIP INVITE and corresponding *Route Control* (0x00E8) and *Outseize Control* (0x002C) messages as described below.

The SIP software allows the host application to fill the Subject header-field of the outbound INVITE message using the NPDI TLV, SIP Subject (0x295B).

This TLV can go in the following messages:

- *Route Control* message (0x00E8)
- *Outseize Control* (0x002C)
- *Subsequent Data PPL Event Request* message (0x0044) - if the *Route Control* or *Outseize Control* message exceed 512 bytes. Refer to *Segmented Call Request (5-73)*.

The SIP Subject TLV is used within the NPDI Universal ICB (0x0033).

Process

If the host application is going to fill in the Subject header field, then the SIP Subject TLV should be present in either the *Outseize* or *Route Control* message or in any of the *PPL Event Request* messages for subsequent data (in the case where more data in the *Outseize/Route Control* message cannot be packed in a single message.) The PPL Event Request has component 0x00A7 with event 0x0023.

The SIP Subject TLV has to be present (not broken up) in any one of these API messages, as any other NPDI TLVs used with the *Outseize/ Route Control* message except the Subsequent Information Status TLV. The Subsequent Information Status TLV must be present in the initial *Outseize/Route Control* message and subsequent *PPL Event Request* messages.

Call Flow

Refer to *Call Flow (5-72)*.

Segmented Call Request

The SIP software allows the host application to send NPDI data that could not be packed into the *Route Control* or *Outseize Control* messages (due to 512 bytes message size limitation), in a PPL Event Request.

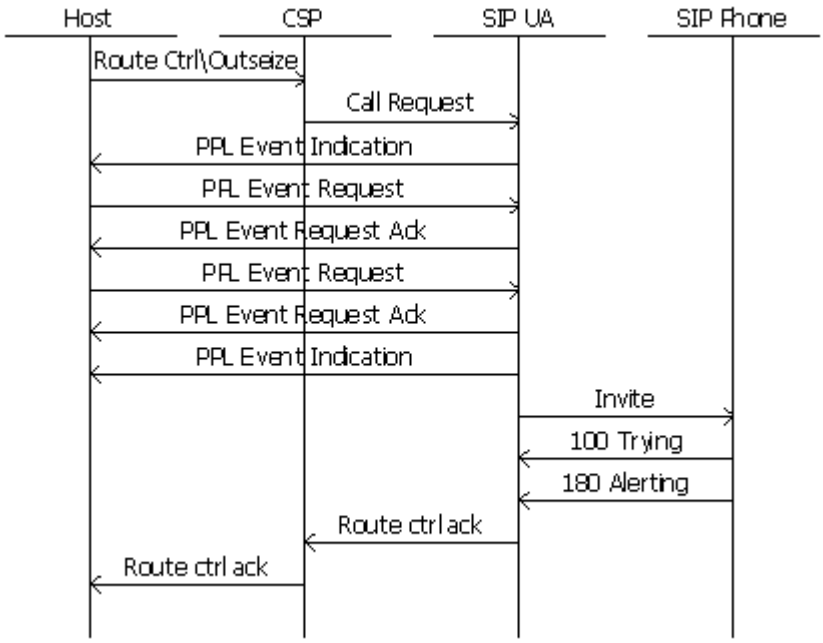
The *Route Control* and *Outseize Control* messages and the subsequent *PPL Event Request* message must contain the Subsequent Information Status TLV (0x2953).

The CSP uses this TLV to validate the message sequence and to identify the last message in the sequence. Any other TLVs have to fully contained in one of the API messages.

Call Flow

In the figure below, the *Route Control* or *Outseize Control* message and the *PPL Event Request* message include the Subsequent Information Status TLV. The call flow steps are described below.

Figure 5-6 Write Access



Call Flow Steps

1. The host application sends the *Route Control* or *Outseize Control* message with the Subsequent Information Status TLV filled in.
2. As soon as the SIP UA receives the Call Request from the L4_CH, it sends out a *PPL Event Indication* message of event type (0x0025). This indication is sent out only for the *Route Control* or *Outseize Control* message having the Subsequent Information Status TLV with correct data. This *PPL Event Indication* message will report the following to the host application:
   - The channel AIB in the indication will have the local Span Channel used, so the host application could use this same addressing in the Subsequent Data PPL Event Requests.
   - The channel reported in the indication will be in the Call Setup status.
   - The sequence number of the *Route Control* or *Outseize Control* message that initiated this indication. This indication is reported using the Message Header Information TLV (0x0014) embedded in the Generic PPL Data ICB (0x1E).

This step is required for the host application to map the PPL Event Indication to the corresponding *Route Control* or *Outseize Control* message.

- The SIP UA is ready to accept more data.

3. The TLVs received from *Route Control* or *Outseize Control* messages (with Subsequent Information TLV) and the subsequent *PPL Event Request* messages are stored in a buffer in the *call_info* data structure. After the final message is received, the SIP UA sends out a *PPL Event Indication* message of event type (0x0026), informing the host application that all data have been received. The TLVs stored in the buffer are validated, extracted and stored in the *call_info* data structure, which is used to create the Invite message. The TLVs received from *Route Control* or *Outseize Control* message, without Subsequent Information TLV, are not buffered.
4. If any one of the TLVs received is has incorrect NPDI data (in the NPDI TLVs), then the *Route Control* or *Outseize Control* message is NACKed with the status 0x1304.
5. If the received subsequent PPL Event Request message does not have the Subsequent Information Status TLV, then the *Route Control* or *Outseize Control* message is NACKed with status 0x1314.
6. If the subsequent *PPL Event Request* message received has a wrong sequence number in the Subsequent Information Status TLV, then the *Route Control* or *Outseize Control* message is NACKed with status 0x1315.
7. The SIP UA PPL starts a timer with an interval of 10 seconds and waits for a subsequent *PPL Event Request* message. If the timer expires the *Route Control/Outseize Control* message is NACKed with status 0x1316.
8. When a Release Channel message is received when waiting for a subsequent *PPL Event Request* message, the channel is released immediately and the *Route Control* or *Outseize Control* message is NACKed with status 0x1304.

Response Status

The following Response Status support this feature.

Table 5-4 Response Status

Response Status	Name	Description
0x1303	Invalid PPL Event	Used in *Route Control* and *PPL Event Request* messages.
0x1304	Invalid Data	Used in *Route Control* and *PPL Event Request* messages.
0x130E	TLV missing	Subsequent Information Status TLV is missing in the *PPL Event Request* message.
0x130F	Incorrect sequence number	The Subsequent Information Status TLV has incorrect sequence number in the *Route Control / Outseize Control* or subsequent *PPL Event Request* message.
0x1310	Timed out	Timed out waiting for subsequent *PPL Event Request* message from the host application. Reported by *Route Control* or *Outseize Control* message.

API Messages

- *Outseize Control* (0x002C)
- *Route Control* (0x00E8)
- *PPL Event Indication* (0x0044)
- *PPL Event Request* (0x0043)

Configuring

You can configure the SIP software to report the Subject header field. By default it is turned off.

You enable this feature by setting a bit mask during the SIP configuration.

You use the existing SIP Message Information Mask TLV (0x027F) in the *VoIP Protocol Configure* (0x00EE) message to enable or disable this feature.

Refer to the *API Reference.*

Samples

The following samples include inbound and outbound SIP messages and the associated API messages.

Inbound Messages

Inbound SIP INVITE message and *Request for Service with Data* Message

```
INVITE sip:1000@135.119.55.40:5060 SIP/2.0
Via: SIP/2.0/UDP 135.119.55.37:5060
From: sipp
   <sip:7000@135.119.55.37:5060>;tag=1;epid=d825335aae
To: sut <sip:1000@135.119.55.40:5060>
Call-ID: 1-4180@135.119.55.37
CSeq: 1 INVITE
Contact: <sip:7000@135.119.55.37:5060>
Max-Forwards: 70
Subject: F-0624 Subject Header Now Supported
Remote-Party-ID: "John Doe"
   <sip:jdoe@foo.com>;party=calling;id-
   type=subscriber;privacy=full;screen=yes
RPID-Privacy: full
Content-Type: applicatoin/sdp
Content-Length: 136
v=0
s=-
c=IN IP4 127.0.0.1
t=0 0
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000


X->H
[00 ee 00 2d 00 00 01 00 01 0d 03 00 64 0e 00 33 01 03 00
   33
00 da 00 14 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
19 00 05 31 30 30 30 00 29 1b 00 0e 31 33 35 2e 31 31
39 2e 35 35 2e 34 30 00 29 1c 00 04 00 00 13 c4 29 23
00 05 37 30 30 30 00 29 25 00 0e 31 33 35 2e 31 31 39
2e 35 35 2e 33 37 00 29 26 00 04 00 00 13 c4 29 2d 00
05 37 30 30 30 00 29 2f 00 0e 31 33 35 2e 31 31 39 2e
35 35 2e 33 37 00 29 30 00 04 00 00 13 c4 29 33 00 01
01 27 18 00 07 02 00 00 00 04 70 00 27 17 00 05 02 00
04 10 00 27 94 00 04 7f 00 00 01 27 95 00 04 00 00 27
10 27 b0 00 02 00 02 27 b1 00 02 00 04 29 16 00 01 01
29 5b 00 24 46 2d 30 36 32 34 20 53 75 62 6a 65 63 74
20 48 65 61 64 65 72 20 4e 6f 77 20 53 75 70 70 6f 72
74 65 64 00
```

Inbound SIP INVITE and Request for Service message with Subsequent Information TLV

In the following sample, the *Request for Service with Data* message uses the Subsequent Information TLV and the *PPL Event Indication* message that carries the Subject Head Field with 250 Bytes of Data.

```
INVITE sip:1000@135.119.55.40:5060 SIP/2.0
Via: SIP/2.0/UDP 135.119.55.37:5060
From: sipp
   <sip:7000@135.119.55.37:5060>;tag=1;epid=d825335aae
To: sut <sip:1000@135.119.55.40:5060>
Call-ID: 1-3840@135.119.55.37
CSeq: 1 INVITE
Contact: <sip:7000@135.119.55.37:5060>
Max-Forwards: 70
Subject: F-0624 Subject Header Now Supported
But It Must Be A Longer To Enable Reported Within The PPL
   Event Indication. This is a test to find out how many
   if the CSP can handle 250
   bytes.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Remote-Party-ID: "John Doe"
   <sip:jdoe@foo.com>;party=calling;id-
   type=subscriber;privacy=full;screen=yes
RPID-Privacy: full
Content-Type: applicatoin/sdp
Content-Length: 136
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
c=IN IP4 127.0.0.1
t=0 0
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000]

X->H
[00 cc 00 2d 00 04 01 00 01 0d 03 00 64 10 00 33 01 03 00
   33
00 b8 00 14 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
19 00 05 31 30 30 30 00 29 1b 00 0e 31 33 35 2e 31 31
39 2e 35 35 2e 34 30 00 29 1c 00 04 00 00 13 c4 29 23
00 05 37 30 30 30 00 29 25 00 0e 31 33 35 2e 31 31 39
2e 35 35 2e 33 37 00 29 26 00 04 00 00 13 c4 29 2d 00
05 37 30 30 30 00 29 2f 00 0e 31 33 35 2e 31 31 39 2e
35 35 2e 33 37 00 29 30 00 04 00 00 13 c4 29 33 00 01
01 27 18 00 07 02 00 00 00 04 70 00 27 17 00 05 02 00
04 10 00 27 94 00 04 7f 00 00 01 27 95 00 04 00 00 27
10 27 b0 00 02 00 02 27 b1 00 02 00 04 29 16 00 01 01
29 53 00 02 01 00]
```

```
X->H
[01 1d 00 43 00 31 01 00 01 0d 03 00 64 10 00 a7 00 23 01
   03
00 33 01 07 00 02 29 53 00 02 02 01 29 5b 00 fb 46 2d
30 36 32 34 20 53 75 62 6a 65 63 74 20 48 65 61 64 65
72 20 4e 6f 77 20 53 75 70 70 6f 72 74 65 64 20 42 75
74 20 49 74 20 4d 75 73 74 20 42 65 20 41 6c 6c 6c 6c
6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c
6c 6c 6c 6f 74 20 4c 6f 6e 67 65 72 20 54 6f 20 45 6e
61 62 6c 65 20 52 65 70 6f 72 74 65 64 20 57 69 74 68
69 6e 20 54 68 65 20 50 50 4c 20 45 76 65 6e 74 20 49
6e 64 69 63 61 74 69 6f 6e 2e 20 54 68 69 73 20 69 73
20 61 20 74 65 73 74 20 74 6f 20 66 69 6e 64 20 6f 75
74 20 68 6f 77 20 6d 61 6e 79 20 69 66 20 74 68 65 20
43 53 50 20 63 61 6e 20 68 61 6e 64 6c 65 20 32 35 30
20 62 79 74 65 73 2e 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 00]
```

Outbound Messages

The following are several *Outseize and Route Control* messages followed by the SIP INVITE message.

Outseize Control Message with the Subject Header TLV

```
H->X
[01 4f 00 2c 00 00 ff 00 01 0d 03 00 64 00 01 03 00 33
01 3d 00 05 29 19 00 05 32 30 30 30 00 27 7e 00 03 08
01 00 29 ff 00 23 2a 0e 00 04 87 77 37 2b 2a 01 00 17
2a 03 00 01 00 2a 07 00 04 00 00 5b 90 2a 02 00 06 2a
08 00 02 00 12 29 16 00 01 01 29 5b 00 fb 46 2d 30 36
32 34 20 53 75 62 6a 65 63 74 20 48 65 61 64 65 72 20
4e 6f 77 20 53 75 70 70 6f 72 74 65 64 20 42 75 74 20
49 74 20 4d 75 73 74 20 42 65 20 41 6c 6c 6c 6c 6c 6c
6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c
6c 6f 74 20 4c 6f 6e 67 65 72 20 54 6f 20 45 6e 61 62
6c 65 20 52 65 70 6f 72 74 65 64 20 57 69 74 68 69 6e
20 54 68 65 20 50 50 4c 20 45 76 65 6e 74 20 49 6e 64
69 63 61 74 69 6f 6e 2e 20 54 68 69 73 20 69 73 20 61
20 74 65 73 74 20 74 6f 20 66 69 6e 64 20 6f 75 74 20
68 6f 77 20 6d 61 6e 79 20 69 66 20 74 68 65 20 43 53
50 20 63 61 6e 20 68 61 6e 64 6c 65 20 32 35 30 20 62
79 74 65 73 2e 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 00]
```

Route Control Message with the Subject Header TLV

```
H->X
[01 76 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00
23 00 06 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
01 0b 01 00 00 01 11 01 01 00 01 01 00 65 00 02 00 00
03 00 33 01 3d 00 05 29 19 00 05 32 30 30 30 00 27 7e
00 03 08 01 00 29 ff 00 23 2a 0e 00 04 87 77 37 2b 2a
01 00 17 2a 03 00 01 00 2a 07 00 04 00 00 5b 90 2a 02

29 5b 00 fb 46
2d 30 36 32 34 20 53 75 62 6a 65 63 74 20 48 65 61 64
65 72 20 4e 6f 77 20 53 75 70 70 6f 72 74 65 64 20 42
75 74 20 49 74 20 4d 75 73 74 20 42 65 20 41 6c 6c 6c
6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c
6c 6c 6c 6c 6f 74 20 4c 6f 6e 67 65 72 20 54 6f 20 45
6e 61 62 6c 65 20 52 65 70 6f 72 74 65 64 20 57 69 74
68 69 6e 20 54 68 65 20 50 50 4c 20 45 76 65 6e 74 20
49 6e 64 69 63 61 74 69 6f 6e 2e 20 54 68 69 73 20 69
73 20 61 20 74 65 73 74 20 74 6f 20 66 69 6e 64 20 6f
75 74 20 68 6f 77 20 6d 61 6e 79 20 69 66 20 74 68 65
20 43 53 50 20 63 61 6e 20 68 61 6e 64 6c 65 20 32 35
30 20 62 79 74 65 73 2e 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 00]
```

Outseize Control Message that contains TLV 0x2953 (NPDI Subsequent Information Status)

```
H->X
[00 56 00 2c 00 00 ff 00 01 0d 03 00 64 00 01 03 00 33
00 44 00 05 29 19 00 05 32 30 30 30 00 27 7e 00 03 08
01 00 29 ff 00 23 2a 0e 00 04 87 77 37 2b 2a 01 00 17
2a 03 00 01 00 2a 07 00 04 00 00 5b 90 2a 02 00 06 2a
08 00 02 00 12 29 16 00 01 01 29 53 00 02 01 00]
```

PPL Event Indication that Is reported to the host application carrying the TLV 0x0014 (Message Header Information)

```
X->H
[00 1d 00 43 00 33 01 00 01 0d 03 00 64 00 00 a7 00 25 01
   02
1e 0a 01 00 14 00 04 00 2c 00 00]
```

PPL Event Request Message that contains TLV 0x2953 and TLV 0x295B (SIP Subject Header) with 250 Bytes

```
H->X
[01 1d 00 44 00 00 ff 00 01 0d 03 00 64 00 00 a7 00 23
```

```
01 03 00 33 01 07 00 02 29 53 00 02 02 01 29 5b 00 fb
46 2d 30 36 32 34 20 53 75 62 6a 65 63 74 20 48 65 61
64 65 72 20 4e 6f 77 20 53 75 70 70 6f 72 74 65 64 20
42 75 74 20 49 74 20 4d 75 73 74 20 42 65 20 41 6c 6c
6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c
6c 6c 6c 6c 6c 6f 74 20 4c 6f 6e 67 65 72 20 54 6f 20
45 6e 61 62 6c 65 20 52 65 70 6f 72 74 65 64 20 57 69
74 68 69 6e 20 54 68 65 20 50 50 4c 20 45 76 65 6e 74
20 49 6e 64 69 63 61 74 69 6f 6e 2e 20 54 68 69 73 20
69 73 20 61 20 74 65 73 74 20 74 6f 20 66 69 6e 64 20
6f 75 74 20 68 6f 77 20 6d 61 6e 79 20 69 66 20 74 68
65 20 43 53 50 20 63 61 6e 20 68 61 6e 64 6c 65 20 32
35 30 20 62 79 74 65 73 2e 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 00]
```

Route Control Message that contains TLV 0x2953 (NPDI Subsequent Information Status)

H->X

[00 7d 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00

```
23 00 06 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
01 0b 01 00 00 01 11 01 01 00 01 01 00 65 00 02 00 00
03 00 33 00 44 00 05 29 19 00 05 32 30 30 30 00 27 7e
00 03 08 01 00 29 ff 00 23 2a 0e 00 04 87 77 37 2b 2a
01 00 17 2a 03 00 01 00 2a 07 00 04 00 00 5b 90 2a 02
00 06 2a 08 00 02 00 12 29 16 00 01 01 29 53 00 02 01
00]
```

PPL Event Indication that Is reported to the host application carrying the TLV 0x0014 (Message Header Information)

```
X->H
[00 1d 00 43 00 36 01 00 01 0d 03 00 64 13 00 a7 00 25 01
   02
1e 0a 01 00 14 00 04 00 e8 00 00]
```

PPL Event Request Message that contains TLV 0x2953 and TLV 0x295B (SIP Subject Header) with 250 Bytes

```
H->X
[01 1d 00 44 00 00 ff 00 01 0d 03 00 64 13 00 a7 00 23
01 03 00 33 01 07 00 02 29 53 00 02 02 01 29 5b 00 fb
46 2d 30 36 32 34 20 53 75 62 6a 65 63 74 20 48 65 61
64 65 72 20 4e 6f 77 20 53 75 70 70 6f 72 74 65 64 20
42 75 74 20 49 74 20 4d 75 73 74 20 42 65 20 41 6c 6c
6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c 6c
6c 6c 6c 6c 6c 6f 74 20 4c 6f 6e 67 65 72 20 54 6f 20
45 6e 61 62 6c 65 20 52 65 70 6f 72 74 65 64 20 57 69
74 68 69 6e 20 54 68 65 20 50 50 4c 20 45 76 65 6e 74
20 49 6e 64 69 63 61 74 69 6f 6e 2e 20 54 68 69 73 20
```

```
69 73 20 61 20 74 65 73 74 20 74 6f 20 66 69 6e 64 20
6f 75 74 20 68 6f 77 20 6d 61 6e 79 20 69 66 20 74 68
65 20 43 53 50 20 63 61 6e 20 68 61 6e 64 6c 65 20 32
35 30 20 62 79 74 65 73 2e 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 00
```

INVITE Message

The following INVITE message is generated with 250 bytes from the *Route Control* or *Outseize Control* message:

```
INVITE sip:2000@135.119.55.61:5060 SIP/2.0
Via: SIP/2.0/UDP 135.119.55.40
To: 2000<sip:2000@135.119.55.61:5060>
From:
   00000000<sip:00000000@135.119.55.40:5060>;tag=39173412
   d3c
Call-ID: EXCEL-CSP1.27.77116.230@135.119.55.40
Contact: 00000000<sip:00000000@135.119.55.40:5060>
User-Agent: Excel_CSP/82.30.171
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Subject: F-0624 Subject Header Now Supported But It Must
   Be Allllllllllllllllllllllllot Longer To Enable
   Reported Within The PPL Event Indication. This is a
   test to find out if the CSP can handle 250
   bytes.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Content-Type: application/sdp
Content-Length: 124
v=0
o=sip 1117807282 1117807282 IN IP4 135.119.55.40
s=SIP_Call
c=IN IP4 135.119.55.43
t=0 0

m=audio 23440 RTP/AVP 18
```

SIP Display Name Parameter in From Header

Overview This feature allows access to a display name parameter in the From Header field in SIP methods. The display name in the From Header field will be reported to the Host for incoming request method and can be configured for specific value for an outgoing request method.

Pertinent Specification RFC 2543/3261

Description

Display Name Write Access

The Display name parameter write access is within the context of the SIP From Header field.

The SIP stack allows the host to fill the Display name parameter using the NPDI TLV SIP From Display Name (0x2928) in the *Route Control* (0x00E8) or *Outseize Control* messages (0x002C). The functionality works for data up to 250 bytes (including the null). If the TLV (0x2928) is sent with data greater than 250 bytes then the message is nacked with a status of 0x1304 (Invalid data).

Example:

```
Parameters supplied in the form of TLVs by the user only
   applied to FROM header field
Displayname(0x2928):Robert
Username (0x291E): bob
Hostname (0x2920): biloxi.example.net

The final Header will look as follows:
From: Robert<sip:bob@biloxi.example.net>;tag=40972175214
```

Display name parameters read access

The reporting of the Display name parameter is within the context of the SIP From Header field. The Display name parameter is reported as a null terminated string using the TLV, SIP Display Name (0x2928) in the *Request for Service* (0x002D) message.

This functionality works for data up to 250 bytes (including the null). If the received display name is greater than 250 bytes then the TLV is not reported.

API Messages Used

The following messages are used by this feature. Refer to the *API Reference* for the formats.

- *Route Control (0x00E8)* message
- *Outseize Control (0x002C)* message

Access To Contact Header

Overview

For an endpoint to endpoint call, one endpoint receives an INVITE message (in this case the CSP) and returns a 200OK response. The endpoint that sent the INVITE message uses the Contact information from the 200OK response to send the ACK message to the CSP and subsequent messages such as BYE.

Likewise, the endpoint that receives the INVITE (in this case the CSP) uses the Contact header to send future requests such as BYE. Final responses to INVITE message have to be routed back the same path.

This feature allows the host read and write access to the following fields in the contact header in the 200OK response for an INVITE message:

- Contact Display Name
- Contact Username
- Contact Parameters

Note that the host portion cannot be changed because if it is changed, the remote endpoint would not be able to route future requests to the CSP correctly.

Pertinent Specification

RFC 2543/3261

API Messages Used

- *PPL Event Request* (0x0044)
- *PPL Event Indication* (0x0043)
- *Connect with Data* (0x0005)

Description

Write Access to Contact Header

The write access to the display name, username, and parameter in the Contact header involves the outbound 200OK response for the INVITE message received and outbound REFER message.

The SIP stack allows the host to fill in the Display Name, Username, and Parameter in the Contact header using the following NPDI TLVs:

- *0x292B - NPDI SIP Contact Display Name*
- *0x292C - NPDI SIP Contact Parameters*
- *0x292D - NPDI SIP Contact Username*

To modify the Contact header fields in the 200OK response for the INVITE message, the host includes the TLVs either of the following:

- PPL Event Request for answer in Layer 4 (Component 0x61, Event 0xC9)
- *Connect with Data* (0x0005) message

Important! For coupled Call Agent Mode enabled calls, where the host has provided the physical span/channel, the CSP will ignore the above TLVs unless the host continues to provide the coupled span/channel.

Read Access to Contact Header

The read access to the Contact Header fields involves the inbound 200OK response for the INVITE message sent and the inbound REFER message.

The SIP stack reports to the host the Display Name, Username and Parameters in the Contact header with the following NPDI TLVs.

- *0x292B - NPDI SIP Contact Display Name*
- *0x292C - NPDI SIP Contact Parameters*
- *0x292D - NPDI SIP Contact Username*

Configuring the CSP to Report Contact Header

The SIP stack reports the Contact Header as follows:

- For Remote End Answered messages- PPL Event Indication (Component 0x00A7, Event 0x0020)
- For Inbound REFER messages- PPL Event Indication (Component 0x00A7, Event 0x0020)

This feature is disabled by default. You can enable this feature with the EXS API as follows.

To enable this feature, set bit 13 in the SIP Message Information Mask (0x027F) TLV in the *VoIP Protocol Configure* (0x00EE) message.

Refer to *Configuring SIP* in the *Converged Services Administrator User's Guide* to configure with SwitchKit CSA.

SIP Access To Parameters in To Header

Overview

This feature gives the host read and write access to the parameter field of the "To" header of the initial INVITE request. The SIP stack reports to the host the parameters (if present) in the "To" header of the INVITE within the *Request for Service with Data* or *PPL Event Indication* for subsequent data. This reporting is disabled by default.

The SIP stack uses the parameters provided by the host in the *Outseize* or *Route Control* messages (and *PPL Event Request* message for subsequent data) for the "To" header of an outbound initial INVITE message.

Pertinent Specifications

- RFC 3398 - ISUP to SIP Mapping
- RCF 3261 - Session Initiation Protocol
- Internet *draft-yu-tel-url-08.txt*

Benefits to Customer

The Internet *draft-yu-tel-url-08.txt* introduced the following three parameters in the tel URI to support number portability (NP) and free phone service:

- routing number ("rn=")
- npdi-dip-indicator ("npdi")
- carrier-id-code ("cic=")

The required or desired parameters might be included in the parameter field of the "To" header in the INVITE request. In particular, customers can have the CSP report the "cic=" parameter to the host. For outbound calls, the host must supply this parameter.

> **Important!** The SIP stack does not support telephone URI fully but it allows telephone numbers to be used as username or hostname.

The following is an example of the INVITE message with the "cic=" parameter highlighted.

```
INVITE sip:+02253618336108866.116.123.10 SIP/2.0
From: <sip:6187851270@216.138.56.76>;tag=fdf55690-
   d88a384c-13c4-348ce5-46943ed9-348ce5
To: <sip:+02253618336108866.116.123.10>;cic=+1-0008
Call-ID: 591b57d3-29373-22811-cd4572bd-22483-
   52549@216.138.56.76
CSeq: 5101 INVITE
Via: SIP/2.0/UDP 216.138.56.76:5060;branch=z9hG4bK-
   348ce5-cd466144-dfe1fb7
User-Agent: Nortel DMS-10UA/v2.1.001
Accept: application/sdp
P-Asserted-Identity:
   <sip:6187851270@216.138.56.76;user=phone>
Privacy: none
Remote-Party-ID:
   <sip:6187851270@216.138.56.76;user=phone>;
   party=calling; privacy=off
Max-Forwards: 70
Supported: 100rel,replaces
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, REFER, PRACK
Contact: <sip:6187851270@216.138.56.76>
Content-Type: application/SDP
Content-Length: 167

v=0
o=- 162175548 463391812 IN IP4 216.138.56.252
s=SIP Call
c=IN IP4 216.138.56.252
t=0 0
m=audio 20008 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=ptime:20
a=sendrecv
```

API and TLVs

Messages

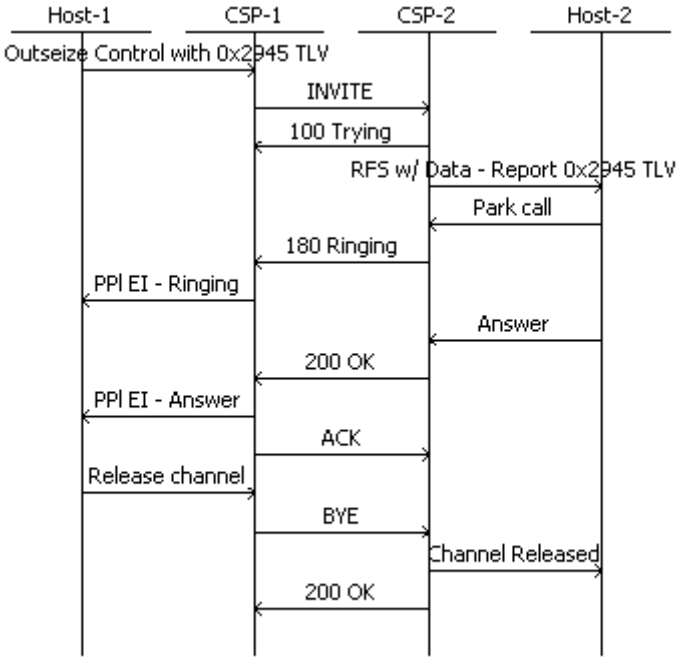
- *PPL Event Indication* (0x0043)
- *PPL Event Request* (0x0044)
- *Route Control* (0x00E8)
- *Outseize Control* (0x002C)
- *Request for Service with Data* (0x002D)

TLVs

- SIP To Parameter (0x2945)
- SIP Message Information Mask TLV (0x027F)

Call Flows

The following call flow shows a SIP call between two CSPs.



Description

This feature allows the host to instruct the SIP stack to insert parameters in the “To” header of the outbound INVITE request. By default, the SIP stack does not add this parameter.

The CSP acts as the User Agent Server (UAS) reporting to the host the parameter, in the inbound INVITE message, within the *Request for Service* message or *PPL Event Indication* message for subsequent data. This parameter is reported using the new TLV - SIP To Parameter (0x2945).

Configuring and Querying with API

This feature is disabled by default. To enable it, set bit 22 of the data part in the SIP Message Information Mask TLV (0x027F) and send it within the *VoIP Protocol Configure* (0x00EE) message.

Configuring with CSA

Refer to *Configuring SIP* in the *Converged Services Administrator User’s Guide* to configure with SwitchKit CSA.

Report and Control of P-Asserted and P-Access Network Info Headers

Overview

The CSP SIP stack can report the “Remote-Party-ID” and the “RPID Privacy” headers if present in the SIP INVITE message.

By default, the SIP stack does not report these P-headers. You have to enable this functionality. See *Configuring the CSP to Report P-headers (5-93)*.

The SIP stack does not modify any other SIP headers for privacy related to this feature.

Pertinent Specifications

- Privacy (Defined in RFC 3323)

Private headers (P-Headers):

- P-Asserted-Identity (Defined in RFC 3325)
- P-Preferred-Identity (Defined in RFC 3325)
- P-Access-Network-Info (Defined in RFC 3455)

Description

Reporting Privacy Header and P-Headers in TLVs

The TLVs in this section report the Privacy header and P-Headers to the host within the NPDI ICB 0x0033 in the following messages:

- *Request for Service with Data* message
- *PPL Event Indications* message (for subsequent data)

0x2960 Privacy Header

This TLV reports the “Privacy” header if present in the inbound INVITE message. The following are the defined set of Privacy header contents but they are not limited to these.

“header” / “session” / “user” / “none” / “critical” / “id”

Byte	Description	
0, 1	Tag	0x2960
2, 3	Length	Variable (Maximum of 250)
4-n	Value	Null terminated ASCII string

0x2961 Private Header Container

This TLV contains nested TLVs that in turn contain the P-Header type and P-Header content. This TLV uses the following nested TLVs:

- 0x2962 Private Header Type TLV
- 0x2963 Private Header Data TLV

This TLV along with the two nested TLVs is reported in the RFS as many number of times as the number of P-Headers appear in the INVITE message. The total length of the NPDI data that can be reported to the host limits this number.

Byte	Description	
0, 1	Tag	0x2961
2, 3	Length	Variable (Maximum of 250)

0x2962 Private Header Type

This TLV contains the P-header type. Use it within the Private Header Container TLV (0x2961) as a nested TLV and in combination with the Private Header data TLV (0x2963). It does not have any meaning if used as a standalone TLV.

Byte	Description	
0, 1	Tag	0x2962
2, 3	Length	0x0001
4-n	Value	0x01 - P-Asserted-Identity 0x02 - P-Preferred-Identity 0x03 - P-Access-Network-Info

0x2963 Private Header Data

This TLV contains the content of the P-header. Use it within the Private Header info TLV (0x2961) as a nested TL and in combination with the Private Header Type TLV (0x2962). It does not have any meaning if used as a standalone TLV.

Byte	Description	
0, 1	Tag	0x2963
2, 3	Length	Variable (Maximum of 241)
4-n	Value	Null terminated ASCII string

Example

The following example shows the format of the TLVs:

```
29 61 - Private Header Container TLV
00 32
     29 62 - Private Header Type TLV
     00 01
        01
     29 63 - Private Header Data TLV
     00 29
        22 43 75 6C 6C 65 6E 20 4A 65 6E 6E 69 6E 67 73 22
        20 3C 73 69 70 3A 66 6C 75 66 66 79 40 63 69 73 63
        6F 2E 63 6F 6D 3E 00
```

Write Access to P-Headers

Use the Privacy Header TLV (0x2960) to add the Privacy header in the outbound INVITE message.

Use the combination of Private Header Type TLV (0x2962) and Private Header Data TLV (0x2963) within the Private Header Container TLV (0x2961) to add the following P-Headers:

- P-Asserted-Identity
- P-Preferred-Identity
- P-Access-Network-Info

Include the TLVs above in the NPDI ICB (0x0033) in the *Outseize Control (0x002C)* or *Route Control* (0x00E8) messages.

The Private Header Container TLV (0x2961) could appear multiple times depending on the number of P-Headers that the host needs to add.

The CSP limits this number based on the NPDI size that it can accept.

API Messages Used

- *Request for Service with Data* (0x002D)
- *PPL Event Indications* (0x0043)
- *Outseize Control* (0x002C)
- *Route Control* (0x00E8)

Configuring the CSP to Report P-headers

By default, the CSP will not report the Privacy header and P-Headers. To enable the reporting, set bit 15 of the Signaling Information TLV (0x027F).

SIP Remote Party ID and RPID Privacy for Outbound Calls

Overview

Important! This feature contains information for the SIP Remote Party ID and RPID Privacy for both inbound and outbound calls.

The SIP Remote Party ID and SIP RPID Privacy header fields allow certain telephony services as well as some regulatory and public safety requirements.

These services include the following:

- calling identity delivery
- calling identity delivery blocking
- tracing originator of call

Baseline SIP supports each of these services independently, but cannot support all combinations. For example, a caller that wants to maintain privacy and consequently provides unintelligible information in the SIP From header field will not be identifiable by intermediaries. However, since SIP does not allow the contents of the From header field to be modified by intermediaries, the intermediaries that do not directly perform cannot perform certain services.

API Messages

- *VoIP Protocol Configur*e (0x00EE)
- *Request for Service with Data* (0x002D)
- *PPL Event Indication* (0x0043)
- *PPL Event Request* (0x0044)
- *Route Control* (0x00E8)
- *Outseize Control* (0x002C)

Configuring with API

You configure the SIP stack on the CSP to report Remote Party ID and RPID Privacy to host applications. The stack disables this functionality by default.

Follow the information below to configure this feature with the API. Otherwise, refer to *Configuring with CSA (5-97)*.

To enable this functionality, set bits 7and 8 in the SIP Message Information Task (0x027F) TLV when you configure the SIP stack with the *VoIP Protocol Configur*e message (0x00EE).

Example

Follow the steps below and the references to the Trace and API Message that follow:

1. When the CSP receives a SIP INVITE message that includes the Remote Party ID header and RPID Privacy header.

   Important! The SIP stack reports the Remote Party ID and RPID Privacy headers in raw, null-terminated ASCII format.

2. The CSP sends the SIP Remote Party ID TLV (0x2959) to the host application in the *Request for Service with Data* (0x002D) or *PPL Event Indication* (0x0043) message.
3. The CSP sends the SIP RPID Privacy TLV (0x295A) to the host application in the *Request for Service with Data* (0x002D) or *PPL Event Indication* (0x0043) message.
4. Since there can be more than one occurrence of Remote Party ID and RPID Privacy headers, the CSP SIP stack will report more than one occurrence of the corresponding SIP Remote Party ID (0x2959) and SIP RPID Privacy (0x295A) TLVs (up to 250 bytes per TLV) in the *Request for Service with Data* (0x002D) or *PPL Event Indication* (0x0043) message.
5. If the contents of the TLV in the *Request for Service with Data* (0x002D) message exceeds 1024 bytes, the remainder of the data will be sent in one or more subsequent *PPL Event Indication* messages.
6. The host shall insert any number of Remote Party ID and RPID Privacy headers, and the total length of the API that will be supported restricts the number.
7. For write access, the SIP Remote Party ID (0x2959) and SIP RPID Privacy (0x295A) TLVs only support a maximum of 250 bytes per TLV, including the null-terminator. This maximum length of 250 bytes is applicable for each occurrence of the TLV in the API and is not the cumulative length.
8. If the TLV exceeds the maximum length or is not terminated by the null-terminator, the CSP returns a 0x1304 NACK.

   Important! The maximum recommended NPDI size is 780 bytes for the *Route Control* (0x00E8) message that includes the NPDI Universal (0x0033) ICB size. The maximum recommended NPDI size is 820 bytes for the *Outseize Control* (0x002C) message and the subsequent *PPL Event Request* (0x0044) message that includes the NPDI Universal (0x0033) ICB size. The maximum byte size supported per SIP message is 1500 bytes.

Trace

```
INVITE sip:service@10.10.1.32:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.2:5060
From: sipp <sip:sipp@10.10.1.2:5060>;tag=1
To: sut <sip:service@10.10.1.32:5060>
Call-ID: 1.8524.10.10.1.2@sipp.call.id
CSeq: 1 INVITE
Contact: sip:sipp@10.10.1.2:5060
Max-Forwards: 70
Subject: Performance Test
Remote-Party-ID: "John Doe"
   <sip:jdoe@foo.com>;party=calling;id-
   type=subscriber;privacy=full;screen=yes
RPID-Privacy: full
Content-Type: application/sdp
Content-Length: 136
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
c=IN IP4       127.0.0.1
t=0 0
m=audio 10000 RTP/AVP 0
```

API Message

```
    [00 f3 00 2d 00 04 ff 00 01 0d 03 00 07 02 00 33 01 03
   00 33
     00 df 00 0f 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 73 65 72 76 69 63 65 00 29 1b 00 0b 31 30 2e
     31 30 2e 31 2e 33 32 00 29 1c 00 04 00 00 13 c4 29 23
     00 05 73 69 70 70 00 29 25 00 0a 31 30 2e 31 30 2e 31
     2e 32 00 29 26 00 04 00 00 13 c4 29 2d 00 05 73 69 70
     70 00 29 2f 00 0a 31 30 2e 31 30 2e 31 2e 32 00 29 30
     00 04 00 00 13 c4 29 33 00 01 01 29 16 00 01 01 29 59
     00 58 22 4a 6f 68 6e 20 44 6f 65 22 20 3c 73 69 70 3a
     6a 64 6f 65 40 66 6f 6f 2e 63 6f 6d 3e 3b 70 61 72 74
     79 3d 63 61 6c 6c 69 6e 67 3b 69 64 2d 74 79 70 65 3d
     73 75 62 73 63 72 69 62 65 72 3b 70 72 69 76 61 63 79
     3d 66 75 6c 6c 3b 73 63 72 65 65 6e 3d 79 65 73 20 00
      29 5a 00 05 66 75 6c 6c 00]
```

An example for write access:

```
H->X
    [01 2e 00 2c 00 00 01 00 01 0d 03 00 00 00 02 03 00 1e
     00 0f 00 02 01 16 00 02 00 00 01 1a 00 03 00 00 00 03
     00 33 01 08 00 10 27 7e 00 03 08 00 00 29 19 00 06 32
     32 32 32 00 29 1b 00 0c 31 30 2e 31 30 2e 31 2e 32
     35 32 00 29 1c 00 04 00 00 13 c4 29 23 00 06 31 31 31
     31 31 00 29 25 00 0c 31 30 2e 31 30 2e 31 2e 32 35 30
     00 29 26 00 04 00 00 13 c4 27 92 00 04 0a 0a 01 bf 27
```

```
93 00 04 00 00 10 7c 29 17 00 01 01 29 5a 00 2e 70 72
69 76 61 63 79 3d 66 75 6c 6c 3b 70 61 72 74 79 3d 63
61 6c 6c 69 6e 67 3b 69 64 2d 74 79 70 65 3d 73 75 62
73 63 72 69 62 65 72 00 29 59 00 0c 61 62 63 40 78 79
7a 2e 63 6f 6d 00 29 5a 00 1e 70 61 72 74 79 3d 63 61
6c 6c 69 6e 67 3b 72 70 69 2d 70 72 69 76 61 63 79 3d
6f 66 66 00 29 59 00 0c 62 61 63 40 78 79 7a 2e 63 6f
6d 00 29 5a 00 1e 70 61 72 74 79 3d 63 61 6c 6c 69 6e
67 3b 72 70 69 2d 70 72 69 76 61 63 79 3d 75 72 69 00
29 59 00 0c 62 63 61 40 78 79 7a 2e 63 6f 6d 00]
```

Trace

```
1 -SENT To 10.10.1.252:5060 at 256
INVITE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
To: 22222<sip:22222@10.10.1.252:5060>
From: 11111<sip:11111@10.10.1.250:5060>;tag=19276987100
Call-ID: EXCEL-CSP1.787.256.550@10.10.1.250
Contact: 11111<sip:11111@10.10.1.250:5060>
User-Agent: Excel_CSP/83.10.62
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
RPID-Privacy: privacy=full;party=calling;id-
   type=subscriber
RPID-Privacy: party=calling;rpi-privacy=off
RPID-Privacy: party=calling;rpi-privacy=uri
Remote-Party-ID: abc@xyz.com
Remote-Party-ID: bac@xyz.com
Remote-Party-ID: bca@xyz.com
Content-Type: application/sdp
Content-Length: 100
v=0
o=sip 0 0 IN IP4 10.10.1.250
s=SIP_Call
c=IN IP4 10.10.1.191
t=0 0
m=audio 4220 RTP/AVP 0
```

To provide SIP Remote Party ID for outbound calls, the CSP SIP stack allows the host to insert Remote-Party-ID and RPID-Privacy headers within the SIP INVITE message.

Configuring with CSA

Refer to *Configuring SIP* in the *Converged Services Administrator User's Guide* to configure with SwitchKit CSA.

TLV Modifications

The following TLVs have been modified to support this feature. The change bars indicate the modifications.

0x2959 SIP Remote Party ID

Used in:
Request for Service with Data (0x002D)
PPL Event Indication (0x0043)
PPL Event Request (0x0044)
Route Control (0x00E8)
Outseize Control (0x002C)

Byte	Description	
0, 1	Tag	0x2959
2, 3	Length	Variable (Maximum of 250 bytes when used in Outseize or Route Control or subsequent PPL Event Request messages)
4-n	Value	Contains the value of the Remote Party ID header as a null-terminated ASCII string.

0x295A SIP RPID Privacy

Used in:
Request for Service with Data (0x002D)
PPL Event Indication (0x0043)
PPL Event Request (0x0044)
Route Control (0x00E8)
Outseize Control (0x002C)

Byte	Description	
0, 1	Tag	0x295A
2, 3	Length	Variable (Maximum of 250 bytes when used in Outseize or Route Control or subsequent PPL Event Request messages)
4-n	Value	Contains the value of the RPID Privacy header as a null terminated ASCII string.

Remote Party ID

Overview

In SIP, the Remote Party ID header field enables popular services as well as some regulatory and public safety requirements.

These services include the following:

- calling identity delivery
- calling identity delivery blocking
- tracing originator of call

The SIP specification (RFC 2543) supports each of these services independently but cannot support all combinations. For example, a caller who wants to maintain privacy and provides unintelligible information in the SIP From header field will not be identifiable by intermediaries. However, since SIP does not allow the contents of the From header field to be modified by intermediaries, the intermediaries that do not directly perform SIP authentication cannot perform certain services.

Pertinent Specification

RFC 3325

API Messages

- *Request for Service with Data* (0x002D)
- *PPL Event Indication* messages (0x0043)

Configuring

You can configure the SIP stack on the CSP to report Remote Party ID and Remote Party ID Privacy to host applications. The stack disables this functionality by default.

To enable this functionality, set 7 in the SIP Message Information Task (0x027F) TLV when you configure the SIP stack with the *VoIP Protocol Configur*e message (0x00EE).

Example

Follow the steps below and the references to the Trace and API Message that follow.

1. The CSP receives a SIP INVITE message that includes the Remote Party ID header (bold in trace below) and RPID Privacy header (italics in trace below).
2. The CSP sends the SIP Remote Party ID TLV (0x2959) to the host application in the *Request for Service with Data (0x002D)*. See bold text in the Example API message below.

3. The CSP sends the SIP RPID- Privacy ID TLV (0x295A) to the host application in the *Request for Service with Data (0x002D)*. See italic text in the Example API message below.
4. If the contents of the *Request for Service with Data (0x002D)* message exceeds 512 bytes, the remainder is sent in one or more subsequent *Request for Service with Data* or *PPL Event Indication* messages (0x0043).

Trace

```
INVITE sip:service@10.10.1.32:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.2:5060
From: sipp <sip:sipp@10.10.1.2:5060>;tag=1
To: sut <sip:service@10.10.1.32:5060>
Call-ID: 1.8524.10.10.1.2@sipp.call.id
CSeq: 1 INVITE
Contact: sip:sipp@10.10.1.2:5060
Max-Forwards: 70
Subject: Performance Test
Remote-Party-ID: "John Doe"
   <sip:jdoe@foo.com>;party=calling;id-
   type=subscriber;privacy=full;screen=yes
RPID-Privacy: full
Content-Type: application/sdp
Content-Length: 136
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
c=IN IP4       127.0.0.1
t=0 0
m=audio 10000 RTP/AVP 0
```

API Message

```
[00 f2 00 2d 00 04 ff 00 01 0d 03 00 07 02 00 33 01 03 00 33
     00 de 00 0f 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 73 65 72 76 69 63 65 00 29 1b 00 0b 31 30 2e
     31 30 2e 31 2e 33 32 00 29 1c 00 04 00 00 13 c4 29 23
     00 05 73 69 70 70 00 29 25 00 0a 31 30 2e 31 30 2e 31
     2e 32 00 29 26 00 04 00 00 13 c4 29 2d 00 05 73 69 70
     70 00 29 2f 00 0a 31 30 2e 31 30 2e 31 2e 32 00 29 30
     00 04 00 00 13 c4 29 33 00 01 01 29 16 00 01 01 29 59
     00 57 22 4a 6f 68 6e 20 44 6f 65 22 20 3c 73 69 70 3a
     6a 64 6f 65 40 66 6f 6f 2e 63 6f 6d 3e 3b 70 61 72 74
     79 3d 63 61 6c 6c 69 6e 67 3b 69 64 2d 74 79 70 65 3d
     73 75 62 73 63 72 69 62 65 72 3b 70 72 69 76 61 63 79
     3d 66 75 6c 6c 3b 73 63 72 65 65 6e 3d 79 65 73 00
     29 5a 00 05 66 75 6c 6c 00]
```

Reporting/Generating SIP Responses

This section describes the following features that report and generate SIP responses:

- *Call Progress Notification to Host with PPL Event Indication (5-102)*
- *Early Media (5-103)*
- *SIP 182 Queued Message (5-106)*
- *SIP/VoIP Media Parameters Synchronization (5-109)*

Call Progress Notification to Host with PPL Event Indication

Overview

When the host application initiates an outbound SIP call using the *Route Control* or *Outseize Control* message, the CSP can receive one or more provisional responses from the remote endpoint before receiving the 200 OK message.

Pertinent Specification

RFC 2543/3261

API Messages

- *Route Control* (0x00E8)
- *Outseize Control* (0x002C)
- *PPL Event Indication (0x0043)*

Configuring

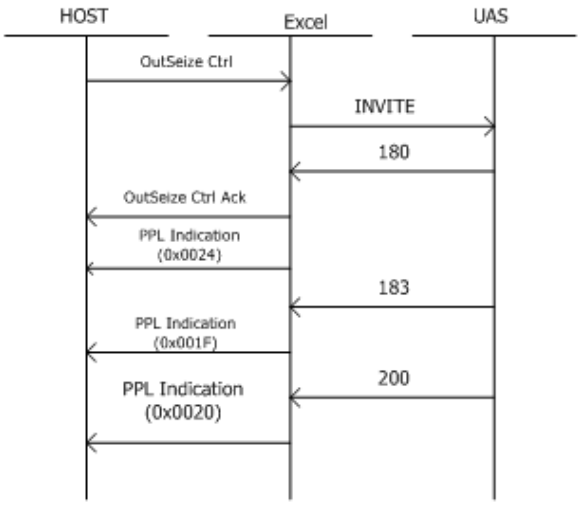
Use the PPL Event Notification Mask (0x0282) TLV in the *VoIP Configure* message to configure the PPL events that you want the host application to see.

Refer to the *Tag Length Value Blocks* chapter in the *API Reference*.

Refer to the180 Ringing PPL Event Indication 0x0024 for the PPL Information: SIP UA 0x00A7.

Call Flow

If the call flow has both 180 and 183 responses, the CSP sends the host application the PPL Event Indication for 180 Ringing (0x0024) and PPL Event Indication for 183 Ringing (0x001F) as indicated below.



Early Media

Overview

Early media is the ability of two SIP User Agents to communicate before a SIP call is actually established. Typically, this scenario occurs when the called party is a PSTN gateway. Before the call is set up, the gateway might provide in-band tones or announcements that inform the caller of the call progress.

Early media can involve the transfer of media from caller to callee. Within the PSTN, forward channels can be established to convey DTMF signaling to select a final destination to call. This feature can be used to access Interactive Voice Response (IVR) systems “behind” 800 numbers.

Pertinent Specification

RFC 2543/3261

Description

The SIP implementation:

- connects the media path prior to the 200 OK message.
- supports pre-answer DTMF and announcements.
- converts the SS7 Call Progress (CPG) message to a 183 response message with SDP.

When the called party wishes to send early media to the caller, the called party sends a 183 response to the caller. That response contains the SDP. When the caller receives the 183 response, it suppresses any local alerting of the user (for example, audible ring tones or pop-up window) and plays out media that it receives.

If the call is ultimately rejected, that called party generates a non-2xx final response. When the caller receives this response, the caller stops playing out or sending media. If the call is accepted, the called party generates a 2xx response and sends that to the caller. Media transmission continue as before.

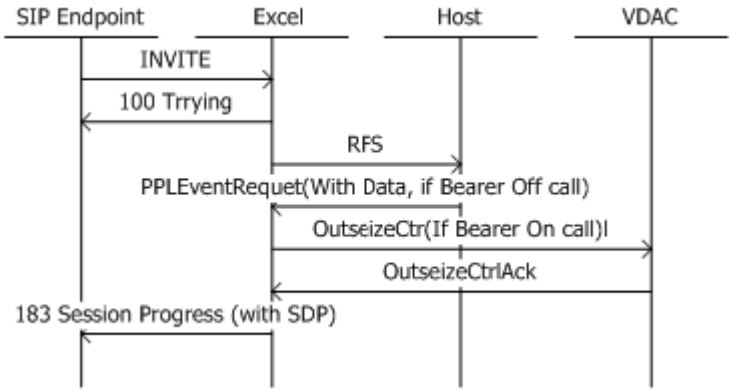
API Messages

See the *PPL Event Request* message (Event ID 0x001F) which is used to generate early media.

Call Flow

Inbound Call - Initiated by host application

In the following call flow, the host application can use PPL Event Request (event id 31) to initiate early media.



Example: For Bearer On Call:

```
X->H - 00 11 00 44 00 02 ff 00 01 0d 03 00 00 00 00 a7 00
   1f 00
```

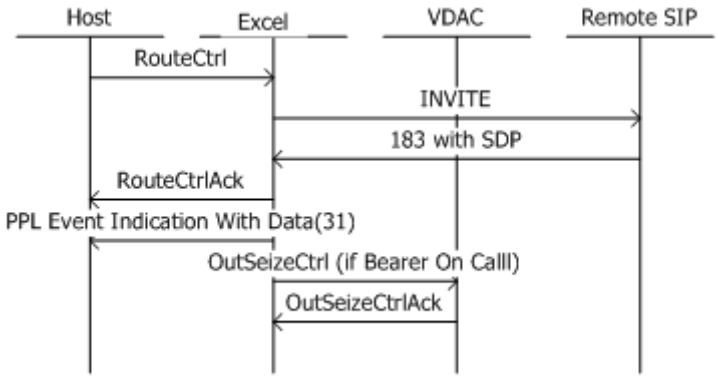
Example: For Bearer Off Call:

```
X->H
00 3f 00 44 00 06 ff 00 01 0d 03 00 07 01 00 a7 00 1f 01
   03
   00 33 00 29 00 01 29 ff 00 23 2a 0e 00 04 0a 0a 41 87
   2a 01 00 17 2a 03 00 01 00 2a 07 00 04 00 00 15 a0 2a
   02 00 06 2a 08 00 02 00 02
```

Outbound Calls

For outbound calls, the CSP detects early media when it receives the 183 response with the Session Description Protocol (SDP) from the host application.

The SIP layer would inform early media detection to host application via PPL Event Indication (Event ID 31) and send the internal Call Progress message.



Example:

X ->H

```
00 3f 00 43 00 06 ff 00 01 0d 03 00 07 01 00 a7 00 1f 01 03
00 33 00 29 00 01 29 ff 00 23 2a 0e 00 04 0a 0a 41 87
2a 01 00 17 2a 03 00 01 00 2a 07 00 04 00 00 15 a0 2a
02 00 06 2a 08 00 02 00 02
```

The following is a typical 183 response message:

```
   SIP/2.0 183 Session Progress
To: <sip:4566@10.10.65.20>;tag=414327a01
From: "User id" <sip:36388@10.10.65.20>
Call-ID: c3943000-22807c6-2737ef30-312e3031@10.10.65.137
CSeq: 101 INVITE
Contact: 4566<sip:4566@10.10.65.20:5060>
Via: SIP/2.0/UDP 10.10.65.137:5060
User-Agent: Excel/80.60.133
Content-Type: application/SDP
Content-Length: 143

v=0
o=sip 162471 162471 IN IP4 10.10.65.20
s=SIP_Call
c=IN IP4 10.10.65.101
t=2209151129 0
m=audio 10844 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

SIP 182 Queued Message

Overview

One usage of the 182 Queued message is as follows: the called party is temporarily unavailable but the server decides to queue the call rather than reject it.

When the called party becomes available, it returns the appropriate final status response.

The reason phrase might give further details about the status of the call for example, “5 calls queued; expected waiting time is 15 minutes.”

The server might issue several 182 (Queued) responses to update the caller about the status of the queued call.

With this feature, the SIP stack is enhanced to allow the host to:

- generate the 182 Queued message
- report the receipt of the 182 Queued message to the host

This feature is supported in Call Agent and Non-Call Agent calls.

Pertinent Specification

RFC 2543/3261

Description

When the CSP receives a 182 Queued message for a call in a pre-answered state, the CSP reports it to the host as a PPL Event Indication message. The PPL Event (0x002E) in SIP component 0x00A7 reports this information.

Reporting 182 Queued Message

When the CSP receives a 182 Queued message for a call in a pre-answered state, the CSP reports it to the host as a PPL Event Indication. The PPL Event (0x002E) in SIP component 0x00A7 reports this information.

See *Call Flow (5-107)* for the format of this PPL Event Indication. The PPL Event Indication message contains the SIP Response Reason Phrase (0x2949) TLV if the Reason phrase is present in the received 182 message and if it is less than 250 bytes in length.

NACK Values

The PPL Event Request that generates the 182 Queued message has the following NACK Values.

Nack Values	Name	Description
0x1303	Invalid PPL Event	PPL Event not expected at this point
0x1304	Invalid Data	Check the date in the API
0x1305	Network Error	Unable to send the message

Generating 182 Queued Message

You do not configure the CSP to generate the 182 Queued message.

The SIP Stack allows the host to generate outbound 182 Queued message using PPL Event Request (0x0026) in SIP UA component (0x00A7).

See *NACK Values (5-107)* for the format of this message.

The SIP Response Reason Phrase (0x2949) TLV is supported in this PPL Event Request.

API Messages Used

- *PPL Event Indications* (0x0043)
- *PPL Event Request* (0x0044)

PPL Information

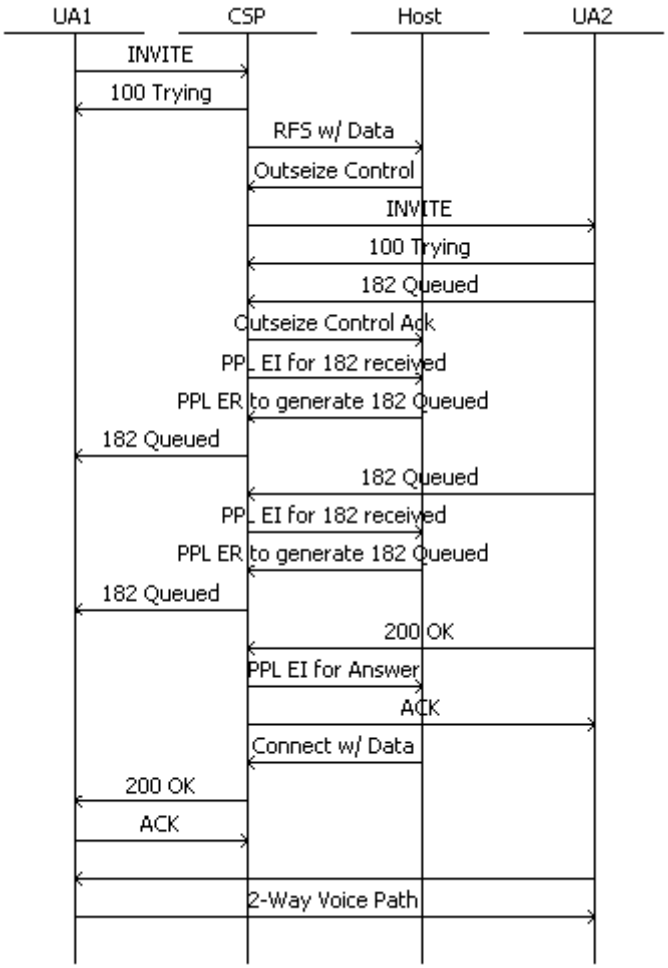
PPL Event Request (0x0026) in SIP UA component (0x00A7).

Configuring

This functionality is disabled by default. Set bit 7 in the PPL Event Notification Mask TLV in the *VoIP Protocol Configure* message to enable this feature.

Call Flow

The following call flow shows messages involved with the reporting and generating SIP 182 Queued message.


UA1
CSP
Host
UA2
INVITE
100 Trying
RFS w/ Data
Outseize Control
INVITE
100 Trying
182 Queued
Outseize Control Ack
PPL EI for 182 received
PPL ER to generate 182 Queued
182 Queued
182 Queued
PPL EI for 182 received
PPL ER to generate 182 Queued
182 Queued
200 OK
PPL EI for Answer
ACK
Connect w/ Data
200 OK
ACK
2-Way Voice Path

SIP/VoIP Media Parameters Synchronization

Overview

The SIP/VoIP Media Parameters Synchronization feature provides a way to maintain SIP signaling in synchronization with the parameters on the following cards:

- VDAC-ONE
- IP Network Interface Series 2

Description

Prior to this feature, the IP Signaling Layer 3 protocol, SIP, maintained a locally cached copy of VoIP media parameters on a per-module basis. When the VoIP media cards are reconfigured through the *Resource Attribute Configure* (0x00E3) message, the local copy of configuration data maintained by SIP on the CSP Matrix Series 3 Card remains unchanged. In this way, the SIP and VoIP media parameters become unsynchronized.

The SIP stack needs to maintain an up-to-date, locally cached copy of VoIP media parameters in order to stay in synchronization with the VoIP media parameters.

Configuring

In order to do so, the host application initiates the SIP stack configuration message, *VoIP Protocol Configure* (0x00EE) with the Resynchronize VoIP Media Parameters TLV (0x0284). This TLV enables the SIP stack to resynchronize with all the VoIP modules that have been previously cached by the SIP stack. For the details of this TLV, refer to the *API Reference*.

Example:

Below is an example of the *VoIP Protocol Configure* message with the Resynchronize VoIP Media Parameters TLV (0x0284) sent by the host application to ensure SIP and VoIP media parameter synchronization.

00 14 00 EE 00 00 FF 00 00 00 00 02 01 C8 00 01 04 02 84 00 01 01

SIP Signaling Support for T.38 Fax Media Sessions (for Non-Call Agent Mode)

The Session Initiation Protocol (SIP) is a signaling protocol used for establishing sessions in an IP network. A session can be a simple two-way telephone call or a collaborative multi-media conference session. SIP uses Session Description Protocol (SDP) to convey and negotiate media session information.

Prior to this feature, the CSP signaling stack establishes only audio media sessions. This feature enhances the SIP and SDP stacks to allow the existing CSP media resources to originate and terminate T.38 fax media sessions. The CSP can now:

- Signal T.38 fax parameters using the network
- Control of local DSP resources to start a T.38 fax session

The CSP SIP signaling state machine can initiate a switchover from audio to T.38 fax or Pass-Through image sessions.

- A session starts with audio capabilities, and upon fax tone detection, the T.38 fax capabilities are negotiated.
- Upon successful negotiation, the session continues with the fax capabilities. The media termination hosts exchange T.38 Internet fax packets.

SIP allows the type and properties of a media session to be changed in the middle of a media session. In SIP this capability is a “re-INVITE”. All session modifications to and from the T.38 fax media will be accomplished using re-INVITEs.

The CSP supports the fax pass-through mode. In this mode, the facsimile communication is handled as a PCM audio call (PCMA/ PCMU as specified in ITU-T recommendation G.711). The fax pass-through mode is important to support interoperability with SIP communication peers that do not support T.38 fax.

The bypass audio connection has the following characteristics:

- PCM G.711 codec
- Silence suppression OFF
- Echo cancellation ON

Pertinent Specifications

- ITU-T Recommendation T.38
- ITU-T Recommendation T.38 Annex D

Benefits to Customer

This feature provides the following benefits to the customer:

- Ease in initiating a T.38 fax session
- Enables the CSP to automatically switch between audio and imaging (fax)

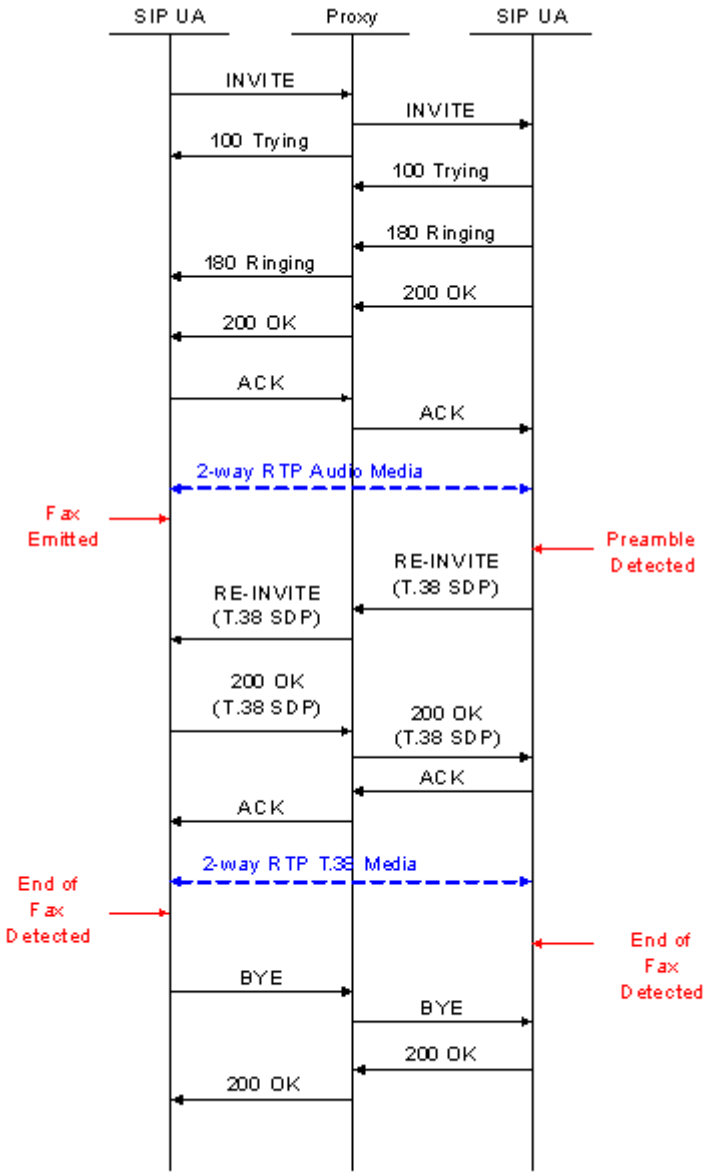
API Messages

- *Outseize Control* (0x002C)
- *Route Control* (0x00E8)
- *Table Download* (0x00D6)

SIP T.38 and Fax Pass-Through Call Flows

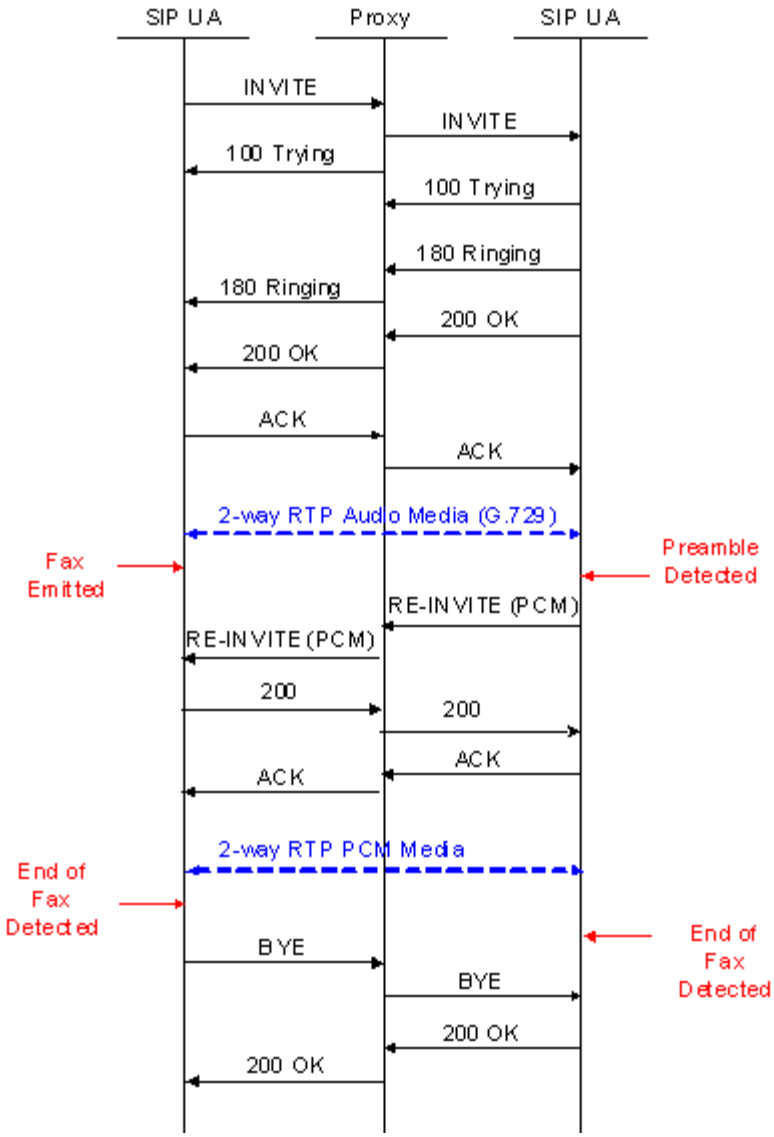
Normal SIP T.38 Call Flow

The call flow below shows a typical T.38 fax call between two SIP User Agents (UAs) and an intermediary proxy. The call begins with an audio media session and then switches over to T.38 fax.


SIP UA
Proxy
SIP UA
INVITE
INVITE
100 Trying
100 Trying
180 Ringing
180 Ringing
200 OK
200 OK
ACK
ACK
2-way RTP Audio Media
Fax Emitted
Preamble Detected
RE-INVITE (T.38 SDP)
RE-INVITE (T.38 SDP)
200 OK (T.38 SDP)
200 OK (T.38 SDP)
ACK
ACK
2-way RTP T.38 Media
End of Fax Detected
End of Fax Detected
BYE
BYE
200 OK
200 OK

Fax Pass-Through Call Flow

The call flow below shows a typical fax pass-through scenario between two SIP UAs and an intermediary proxy. The call begins with an audio media session and then switches over to an audio connection.



Detailed Description

The CSP supports both T.38 fax and Pass-Through (also known as bypass) as two independent modes of fax transmission but Dialogic does not recommend running both modes on the same IPN-2 card.

Either mode is configurable in the CSP on a per Layer 4 (L4)-Router Resource Group basis.

The host application controls the fax mode for outbound calls on a per call basis. This control is done through populating the NPDI SIP Fax Mode (0x27AF) TLV in the *Route Control* (0x00E8) or *Outseize Control* (0x002C) messages. The NPDI SIP Fax Mode (0x27AF) TLV is also supported in the L4-Router Resource Group by using the *PPL Table Download* (0x00D6) message.

CSP Channel Call Setup

For inbound calls, the fax type is extracted from the L4 Call Control PPL Information-Router.

For outbound calls, the fax type is extracted from the L4-Router if the host application issues a *Route Control* (0x00E8) message and does not include the NPDI SIP Fax Mode (0x27AF) TLV.

If the host application issues an *Outseize Control* (0x002C) message, the NPDI SIP Fax Mode (0x27AF) TLV must be included in the message if a fax session is desired. Once the fax type is provided to the DSP (even the session is only audio at this point in time), it cannot be changed in the middle of the call.

The L4-Router stores the NPDI SIP Fax Mode (0x27AF) TLV on a per resource group basis. This TLV is used for inbound SIP calls and, if not overridden by the host, it can be used for outbound SIP calls.

Audio to Fax Switchover

The CSP sends and receives fax media only after a switchover from audio media. A session must always begin as an audio session and then it will switch over to a fax session if in-band fax tones are detected.

Once a session switches over from audio to fax, the CSP does not support any subsequent modifications of the session. For instance, the session can not be switched back to audio once it has made the switchover to fax.

Media Gateway (IP/TDM) and/or a SIP B2BUA (IP/IP)

The CSP can be used as a Media Gateway (IP/TDM) and/or a SIP B2BUA (IP/IP) based on this feature. In the SIP B2BUA mode, the two UA contexts within CSP do not propagate fax events (such as re-INVITEs) between each other. Instead, the end-to-end in-band tones trigger the underlying T.30 engines to drive the session from audio to fax independently for each UA context.

Transparent Mode

The CSP does not support a mode where fax tone detection can be completely disabled in the DSPs. From a signaling perspective, the CSP supports disabling the fax mode. This is “transparent mode.”

The transparent mode works as follows: The packetization DSPs do not report any fax events to the signaling stacks, but they internally detect non-audio transmissions and automatically try to disable silence suppression, echo cancellation etc. for the session.

In-Band Fax Tones

The CSP will react to in-band fax tones on its own without host intervention. The host has overall control through configuration and even on a per call basis for outbound calls. However, in the interest of meeting timing constraints in negotiating fax sessions, the audio to image switchover process will be completely handled within the CSP. The CSP continues to generate PPL event indications for inbound re-INVITEs. Outbound re-INVITEs that are triggered by the IPN-2 DSPs are not indicated to the host.

The CSP reacts to in-band fax tones regardless of the direction in which the session was setup (inbound or outbound). Since the CSP reacts to in-band fax tones by sending re-INVITEs, the lack of sense of call setup direction almost always causes re-INVITE glare. The CSP handles re-INVITE glare. It handles the re-INVITE race based on a first come first served basis. If the CSP SIP signaling stack receives the fax started event from the underlying DSP first, then it initiates a re-INVITE. This re-INVITE will be an atomic operation in the CSP, which means that the CSP automatically accepts nothing but a response to the re-INVITE at this point. Any intermediary events are queued up for later processing.

Glare Effect

A fax transmission can be detected by the receiving side, the emitting side or both. For the emitting side, a 'glare' effect may appear. The CSP SIP stack state machine is designed to not introduce glare and to handle glare introduced by the other party. The CSP SIP stack ignores the L3_L3Pn_FAX_START indication from the DSP if it arrives after a SIP re-INVITE (T.38 fax or bypass) has been received. If the CSP sends a SIP re-INVITE and receives a re-INVITE glare from the other party and if the two do not crossover on the wire than both the

transactions will complete independent of each other. The IPN-2 DSP receives two back-to-back *Resource Attribute Configure* messages in this case.

No Fallback Mode

If the remote SIP gateway or endpoint does not support Fax T.38, the CSP does not fallback to Fax G711. Depending on the CSP's configuration, the CSP will transmit either Fax T.38 or ByPass only.

Configuring and Querying with API

Configuring

Enabled by setting the SIP Stack T.38 Fax Support (0x011C) TLV in the *VoIP Protocol Configure* (0x00EE) message.

Querying

The host can query this feature by querying the SIP Stack T.38 Fax Support (0x011C) TLV in the *VoIP Protocol Query* (0x00EF) message.

Configuring with CSA

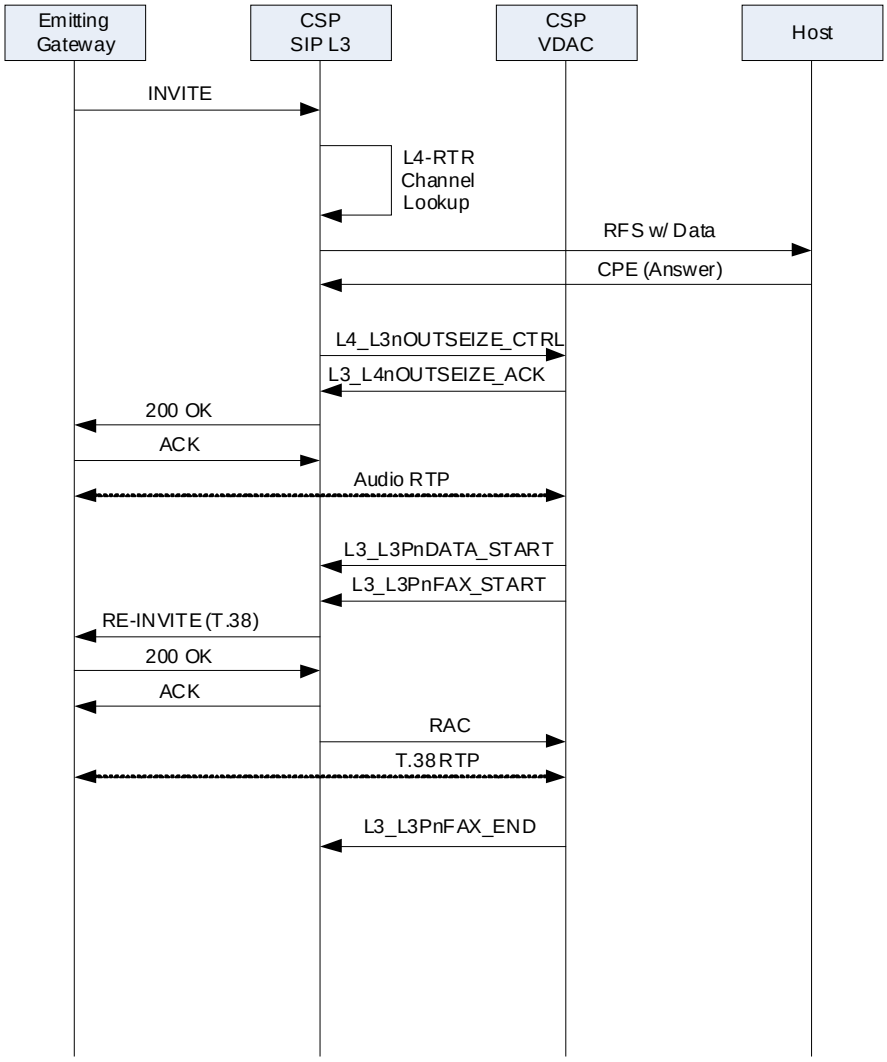
Refer to *Configuring SIP* in the *Converged Services Administrator User's Guide* to configure with SwitchKit CSA.

Call Flows

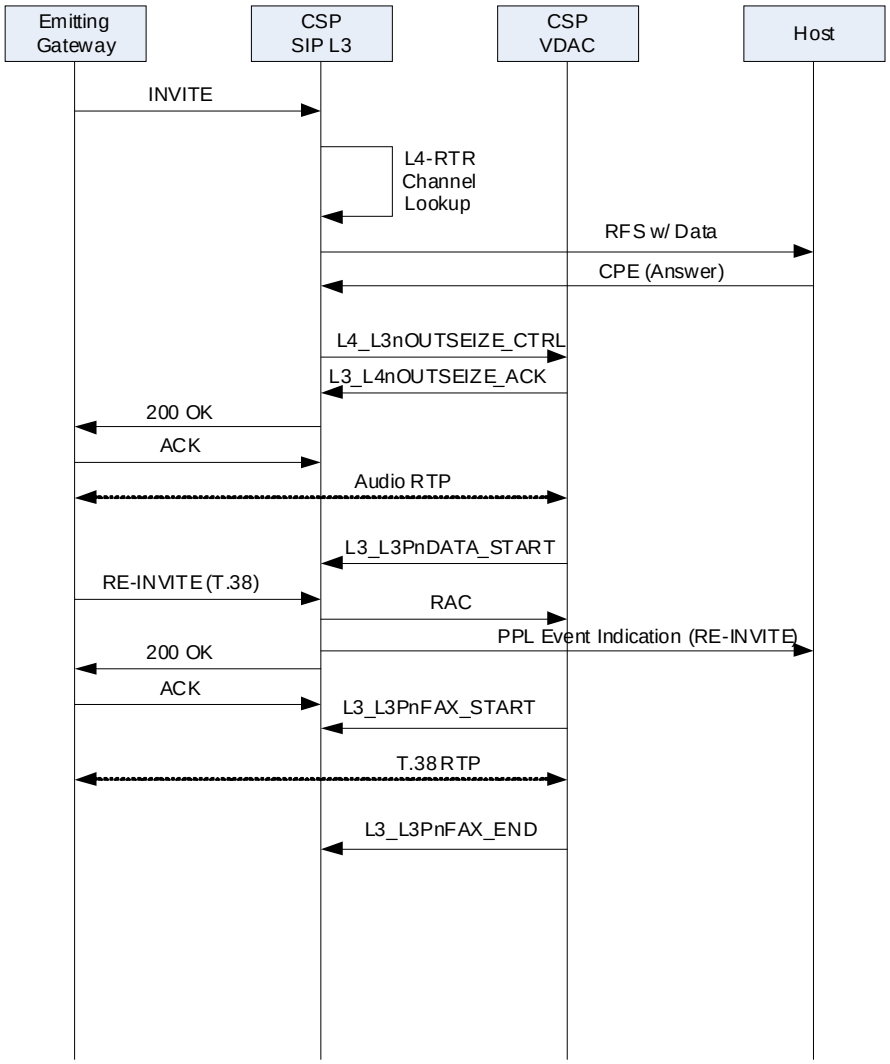
The T.38 fax call flow scenarios, include the following aspects of the call sequence:

- Detection of the fax transmission
- Usage of the T.38 session description attributes
- Session termination

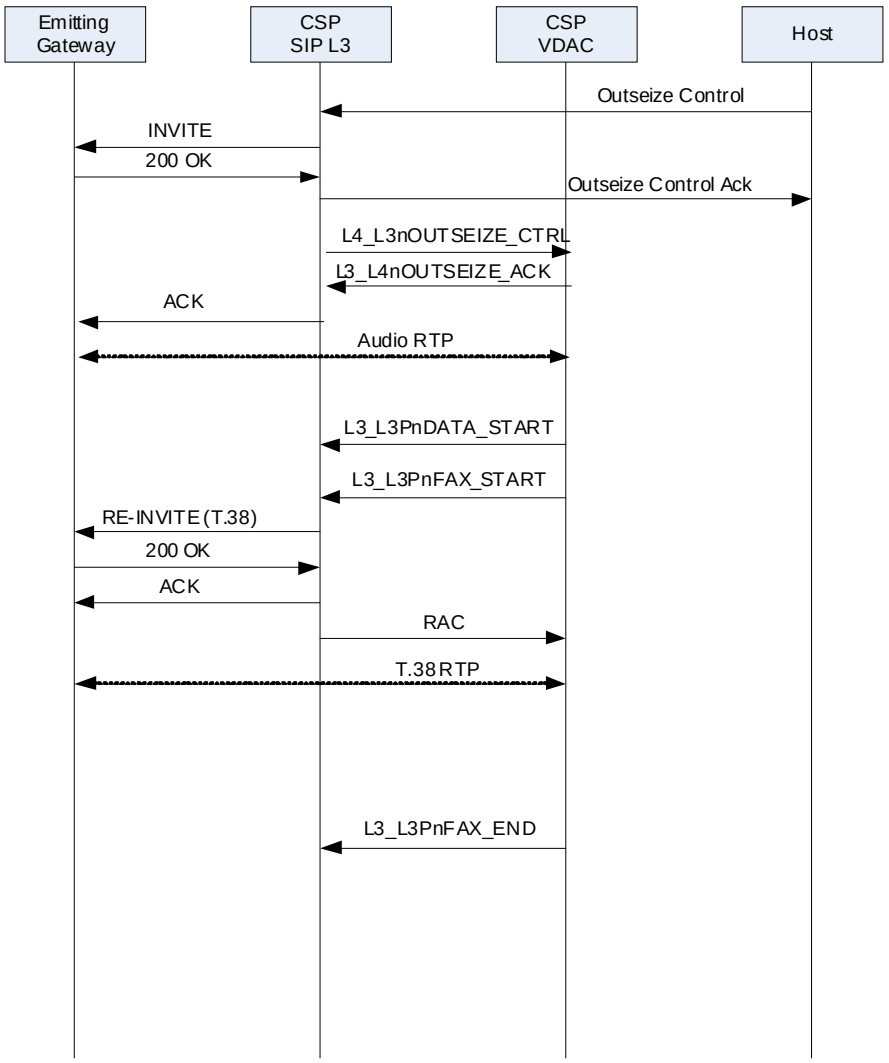
Scenario 1: CSP is the terminating gateway and sends a T.38 fax re-INVITE



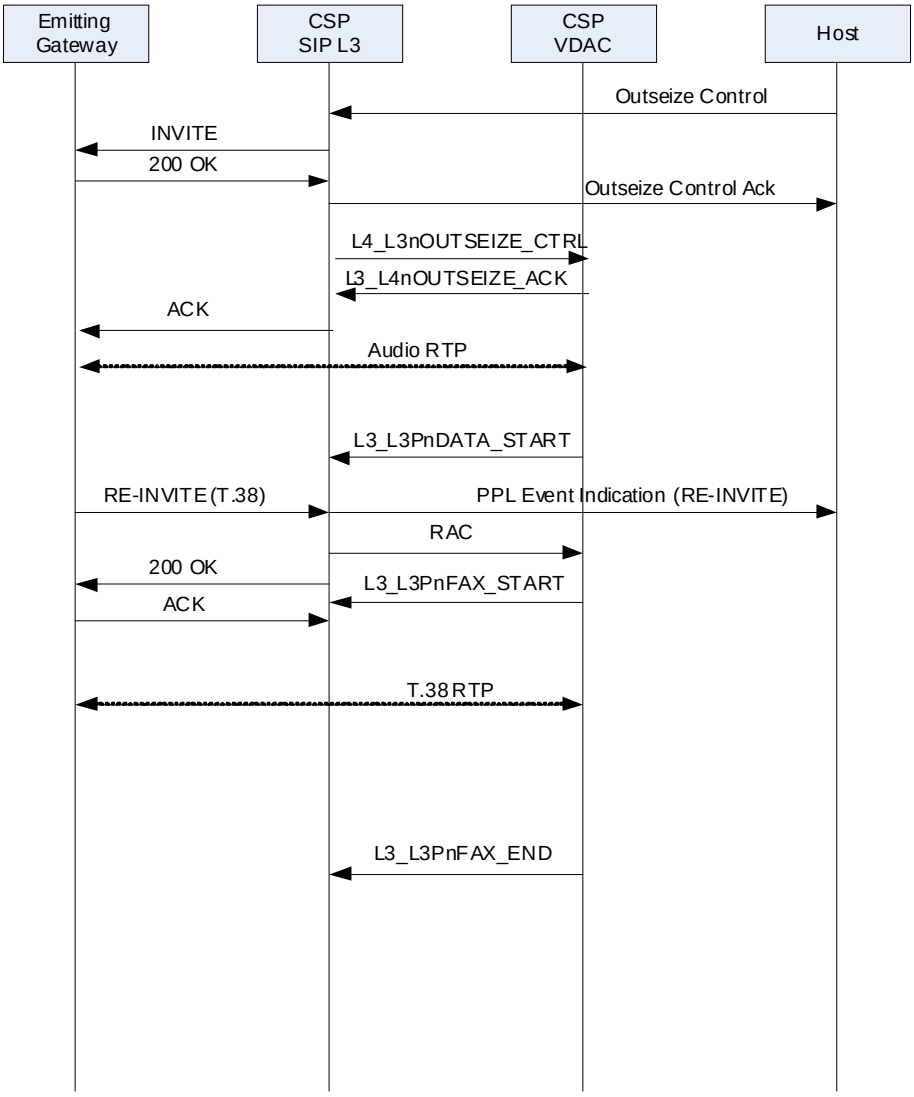
Scenario 2: CSP is the terminating gateway and receives a T.38 fax re-INVITE



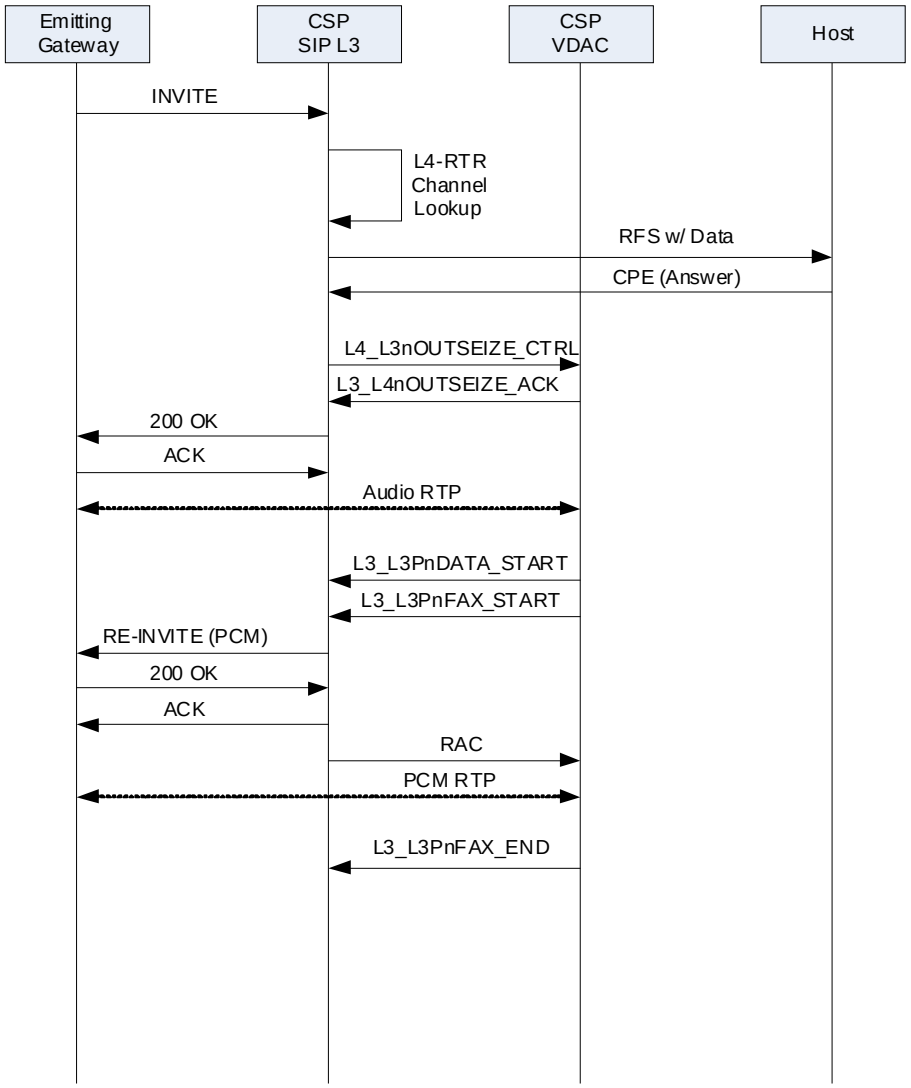
Scenario 3: CSP is the originating gateway and sends the T.38 fax re-INVITE



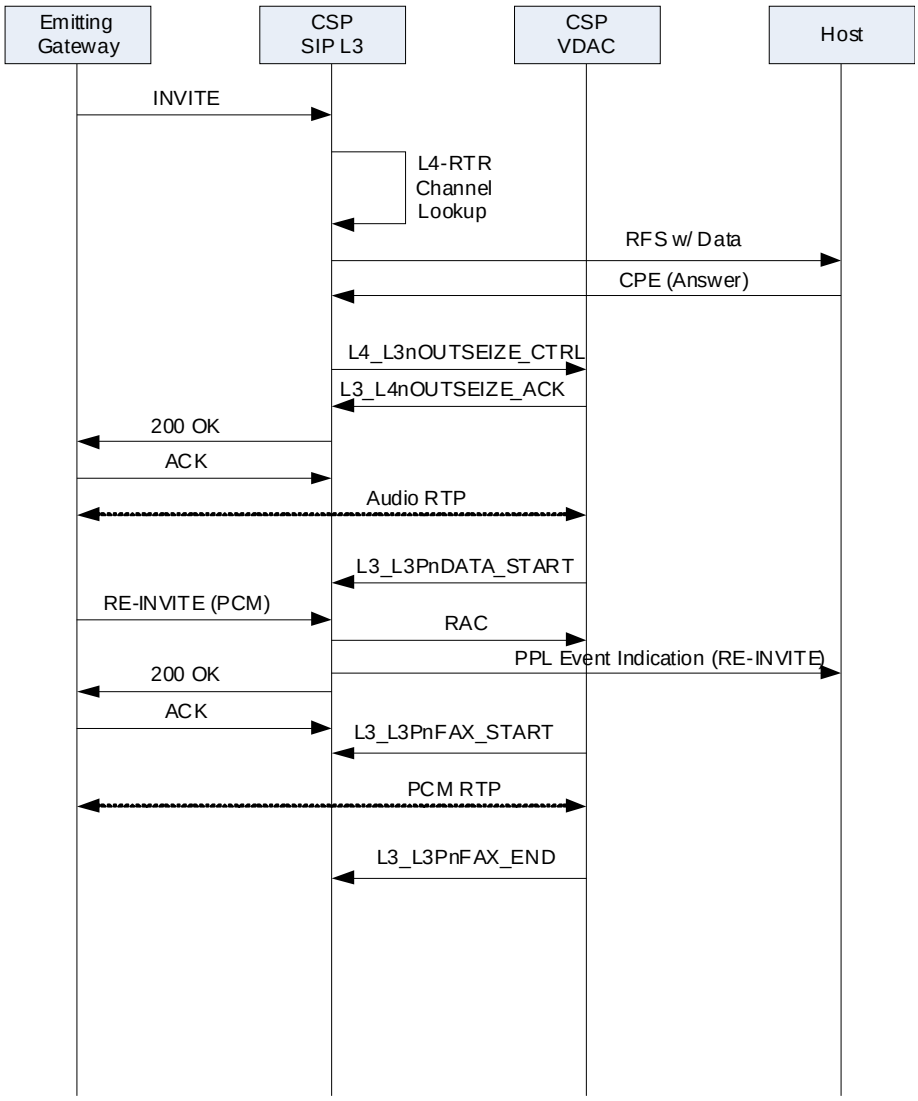
Scenario 4: CSP is the originating gateway and receives the T.38 fax re-INVITE



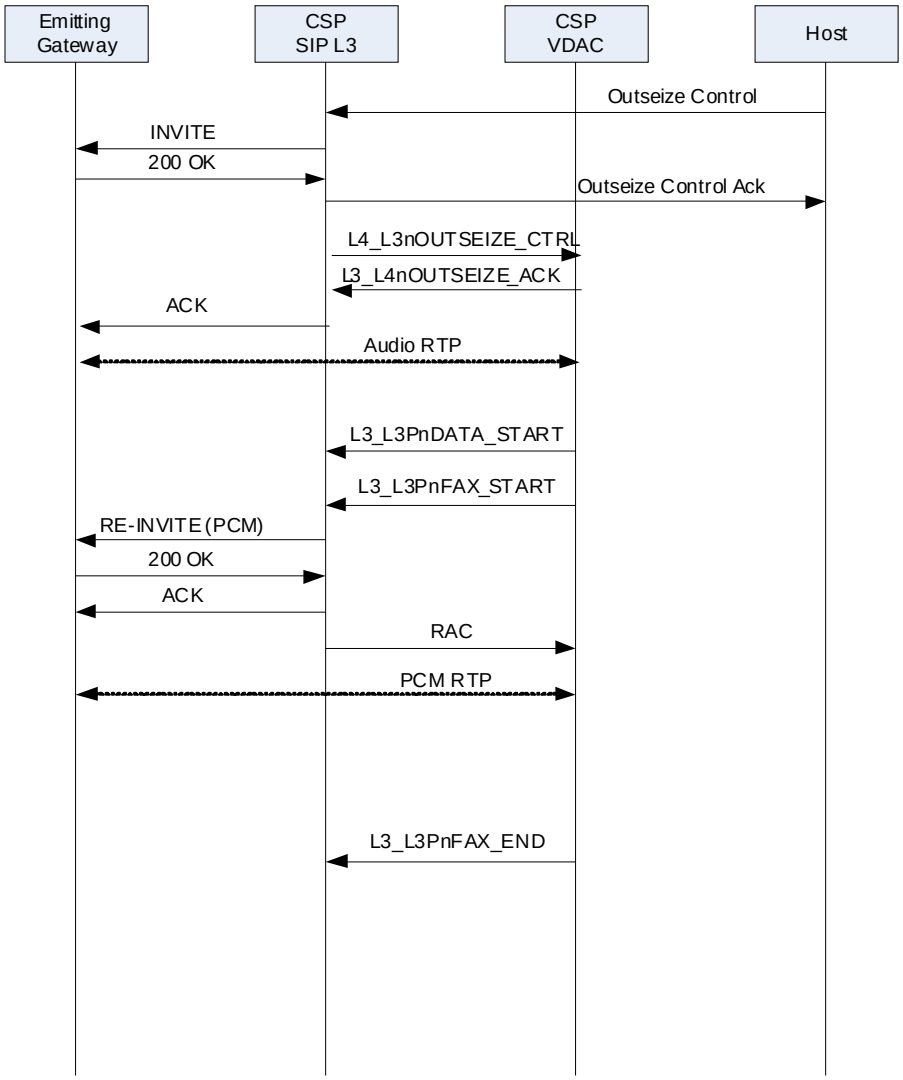
Scenario 5: CSP is the terminating gateway and sends a Bypass fax re-INVITE



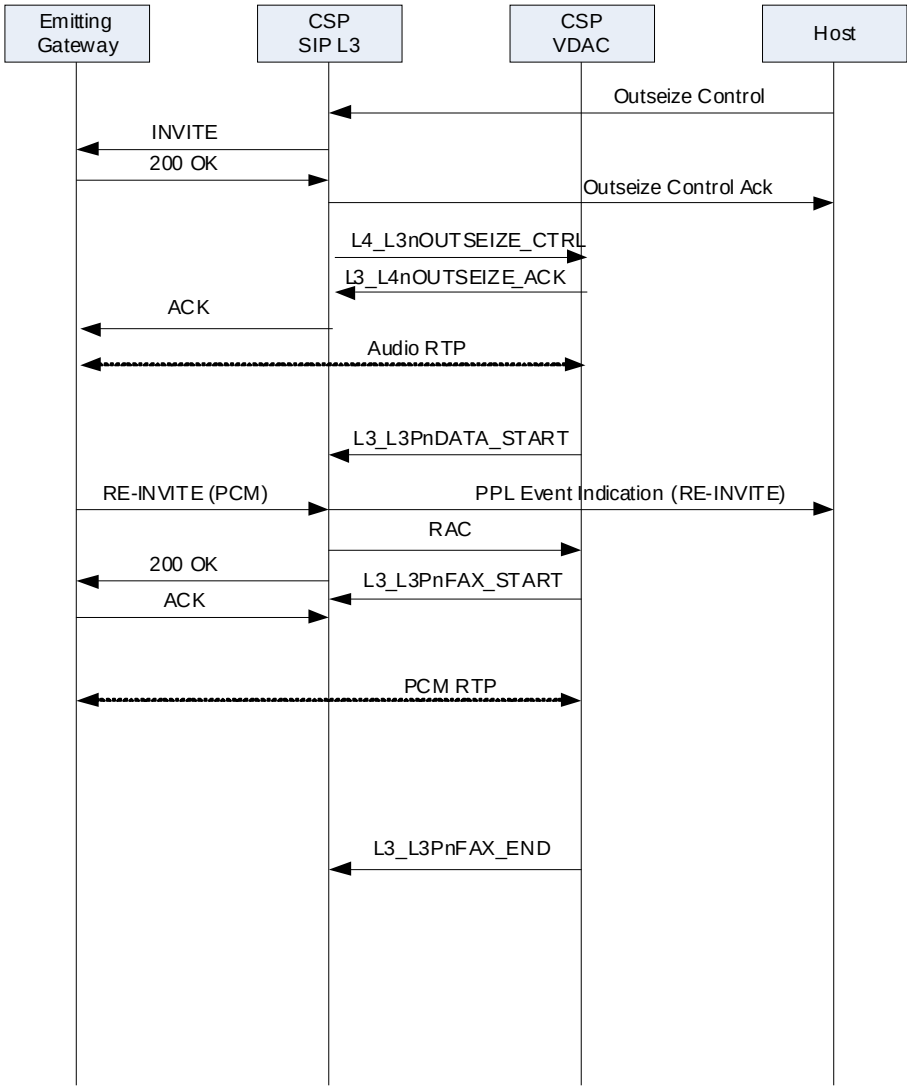
Scenario 6: CSP is the terminating gateway and receives a PCM fax re-INVITE



Scenario 7: CSP is the originating gateway and sends the PCM fax re-INVITE



Scenario 8: CSP is the originating gateway and receives the PCM fax re-INVITE



API Message Changes

This feature results in changes to the following API messages:

VoIP Protocol Configure (0x00EE)
VoIP Protocol Query (0x00EF)

The 0x011C TLV is added to both messages.

Important! This TLV is mandatory for enabling T.38 fax capability on the CSP.

:	TLVs Mandatory 0x011C SIP Stack T.38 Fax Support

Route Control (0x00E8)

The optional 0x27AF TLV is added to this message.

Important! This TLV is optional for fax calls when using L4 managed route control.

SIP

Extended ICBs

0x0033 NPDI Universal ICB

0x27AF NPDI SIP Fax Mode TLV
Value: 0x00 Disable; 0x01 T.38 Fax; 0x02 Fax Bypass

Information Control Block (ICB) Change

This feature supports the following change to the 0x0033 NPDI Universal ICB in the *Outseize Control* (0x002C) message.

0x0033 NPDI Universal ICB

The TLV 0x27AF is added to this ICB.

Important! This TLV is mandatory for fax calls when using host managed Outseize Control and Route Control.

	SIP 0x27AF NPDI SIP Fax Mod

Tag Length Values (TLVs) (New)

This feature supports the following new TLVs.

0x011C SIP Stack T.38 Fax Support

Used in:

VoIP Protocol Configure (0x00EE)
VoIP Protocol Query (0x00EF)

Byte	Description	
0,1	Tag	0x011C SIP Stack T.38 Fax Support
2,3	Length	0x0001
4	Value[0]	00=Disabled 01=Enabled

0x27AF NPDI SIP Fax Mode

Use this TLV in the following messages:

- *Outseize Control* (0x002C) and *Route Control* (0x00E8)
  Host managed outseize control and route control. Use as a mandatory TLV in this message.
- *PPL Table Download* (0x00D6)
  Use as a L4-Router TLV in this message.
- *Route Control* (0x00E8)
  L4 managed route control. Use as an optional TLV in this message.

Byte	Description	
0,1	Tag	0x27AF NPDI SIP Fax Mode
2,3	Length	0x0001
4	Value[0]	00=Disabled 01= Initiate T.38 re-INVITE 02= Initiate Bypass (PCM) re-INVITE

Core SIP Functionality

This section describes the core functionality of the SIP stack including the following:

- *Authentication (5-128)*
- *Host Control Method of SIP 180 Provisional Response Generation (5-166)*
- *RFC 2833 (DTMF Digits) Dynamic Payload Negotiation (5-168)*
- *Session Timers (5-170)*
- *Delayed Media (5-171)*
- *RE-INVITE Message (5-172)*
- *Codec List in SIP-Initiated Offer (5-173)*
- *Send and Receive SIP Signals Using the Same Port (5-177)*
- *SIP Notifications of Options (5-191)*
- *Disabling the SIP Domain Name System (DNS) Server (5-195)*

Authentication

Overview

Dialogic implemented SIP Authentication which includes protective measures to prevent an active attacker from modifying and replaying SIP requests and responses.

Pertinent Specification

RFC 2543

Description

SIP authentication enables the CSP to validate the legitimacy of the subscribers. It is like any login/password based scheme in that it ensures that only valid users can make calls through the CSP.

The same cryptographic measures that are used to ensure the authenticity of the SIP message also serve to authenticate the originator of the message. SIP extends the HTTP WWW-Authenticate and Authorization header fields and their Proxy counterparts to include cryptographically strong signatures.

The enhancements include support for both authentication and authorization using digest and basic authentication methods.

There are two methods for incoming call Authentication.

- Authenticating an inbound SIP INVITE using CSP managed data encryption/decryption:
- Authenticating an inbound SIP INVITE using host managed data encryption/decryption:

Authenticating an inbound SIP INVITE using CSP managed data encryption/decryption:

This method allows the CSP to manage the data encryption/decryption and requires the host to supply the authentication parameters to the CSP using a *PPL Event Request* (0x0044) message upon reception of a *SIP Request for Service with Data* (0x002D) message.

When the *SIP Request for Service with Data* (0x002D) message is reported to the host, the host may choose to authenticate the call by sending a *PPL Event Request* (0x0044) message to the SIP Component. The host will supply the following:

- NPDI SIP Authenticate Scheme TLV (0x2937)
- NPDI SIP Authentication Realm TLV (0x2938)
- NPDI SIP Authenticate Username TLV (0x2939)

- NPDI SIP Authenticate Password TLV (0x293A)

When the CSP receives the *PPL Event Request* message, the SIP stack will send the 401 Unauthorized response and include the WWW-Authenticate header using the information from the above TLVs. When the INVITE with credentials arrives, the CSP decrypts and validates the credentials and send the PPL Event Request ACK to the host.

Authenticating an inbound SIP INVITE using host managed data encryption/decryption:

This method requires the host to manage the data encryption/ decryption upon reception of the *Request for Service with Data* message.

An alternate method to authenticate an inbound call is to release the call with authentication information. The host may release the inbound call using the *Release Channel with Data* (0x0036) message and include the SIP response code in the NPDI SIP Response Code TLV (0x2915) and the ASCII value of the WWW-Authenticate header using the SIP Authenticate Header TLV (0x2943). It is the host's responsibility to properly encrypt the header as necessary. The channel is released when this message is sent to the CSP, and a SIP authentication message will be sent by the CSP to challenge the INVITE. When a new INVITE with credentials is received, the Request or Service with Data will report the value of the Authorization header in the SIP Authorization Header TLV (0x2941). It is the host's responsibility to decrypt the contents of the Authorization header and validate the credentials.

The CSP automatically responds to SIP Authentication challenges when the credentials are supplied in the *Route Control* (0x00E8) message or *Outseize Control* (0x002C) message.

When included in the Route Control or Outseize Control message, the following allow the CSP SIP stack to properly respond to an authorization challenge.

- NPDI SIP Authorization Username TLV (0x293B)
- NPDI SIP Authorization Password TLV (0x293C)
- NPDI SIP Proxy Authorization Username TLV (0x293D)
- NPDI SIP Proxy Authorization Password TLV (0x293E)
- NPDI SIP Authentication Timeout TLV (0x293F)

If the outbound CSP INVITE is challenged, the CSP will automatically send a new INVITE request with credentials based on the host provided authorization information.

API Messages Used

- *Outseize Control* (0x002C) message
- *PPL Event Request* (0x0044) message
- *Release Channel with Data* (0x0036) message
- *Route Control* (0x00E8) message

Configuring

Modify the VoIP Protocol Configure message (0x00EE) to include the SIP Information Mask TLV (0x027F) if the Authorization header or Proxy-Authorization headers are to be reported (set bits 1 and 2). The reporting of the Authorization header is mandatory if the inbound call is authenticated by using the *Release Channel with Data* method.

Switchkit Considerations

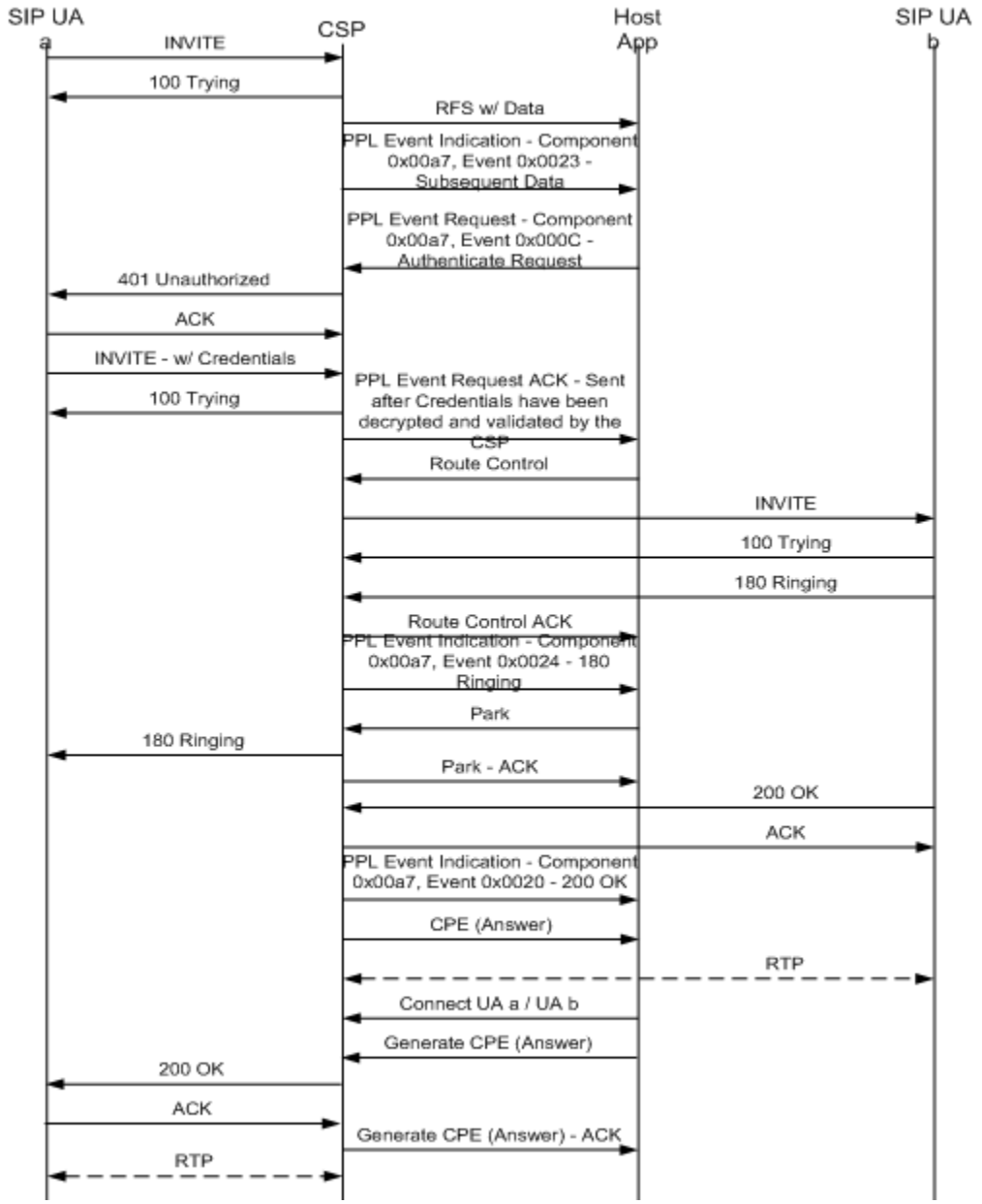
As the authentication process may take several seconds to complete, the acknowledgement to the API messages listed may not be returned by the CSP for several seconds. It is possible for Switch Kit LLC to timeout waiting for a response to these API messages prior to those responses being reported. Use the SK_AdjustMessageTimeout message to change the default time-out for responses.

Call Flows

This section contains four call flows and corresponding message traces.

401 Authenticate for Inbound Call

The following call flow show an inbound INVITE message that get authenticated using a PPL Event Request.



Host API and Signaling Trace

```
1 -RECEIVED From 192.168.1.51:52678 at 23596
INVITE sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK27519f66
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
To: <sip:8623000@192.168.1.102>
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
CSeq: 101 INVITE
User-Agent: CSCO/7
Contact: <sip:8623001@192.168.1.51:5060>
Expires: 180
Content-Type: application/sdp
Content-Length: 191
Accept: application/sdp

v=0
o=Cisco-SIPUA 20722 17063 IN IP4 192.168.1.51
s=SIP Call
c=IN IP4 192.168.1.51
t=0 0
m=audio 30670 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000


2 -SENT To 192.168.1.51:5060 at 23596
SIP/2.0 100 Trying
To: <sip:8623000@192.168.1.102>;tag=11635c2c
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK27519f66
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [01 33 00 2d 00 03 00 00 01 0d 03 00 00 02 00 33 01 03
   00 33
     01 1f 00 18 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 38 36 32 33 30 30 30 00 29 1b 00 0e 31 39 32
     2e 31 36 38 2e 31 2e 31 30 32 00 29 1c 00 04 00 00 13
     c4 29 23 00 08 38 36 32 33 30 30 31 00 29 25 00 0e 31
     39 32 2e 31 36 38 2e 31 2e 31 30 32 00 29 26 00 04 00
```

```
     00 13 c4 29 2d 00 08 38 36 32 33 30 30 31 00 29 2f 00
     0d 31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 30 00 04
     00 00 13 c4 29 33 00 01 01 27 18 00 09 02 00 00 00 07
     86 23 00 10 27 17 00 07 02 00 07 86 23 00 00 27 94 00
     04 c0 a8 01 33 27 95 00 04 00 00 77 ce 27 b0 00 02 00
     02 27 b1 00 02 00 01 29 16 00 01 01 29 54 00 08 38 36
     32 33 30 30 30 00 29 55 00 0e 31 39 32 2e 31 36 38 2e
     31 2e 31 30 32 00 29 56 00 04 00 00 13 c4 29 50 00 31
     30 30 30 33 36 62 33 63 2d 32 35 31 35 30 30 31 35 2d
     32 34 32 66 35 31 65 37 2d 34 65 30 61 63 30 62 32 40
     31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 53 00 02 01
      00]

H->X
    [00 0c 00 2d 00 03 00 00 01 0d 03 00 00 02]

X->H
    [00 59 00 43 00 05 00 00 01 0d 03 00 00 02 00 a7 00 23
   01 03
     00 33 00 43 00 04 29 53 00 02 02 01 29 51 00 22 30 30
     30 33 36 62 33 63 32 35 31 35 30 30 31 32 30 36 65 39
     37 34 33 63 2d 31 39 65 30 38 31 32 35 00 29 52 00 09
      31 31 36 33 35 63 32 63 00 2a 0e 00 04 c0 a8 01 33]

H->X
    [00 05 00 43 00 05 00]

H->X
    [00 4a 00 44 00 00 ff 00 01 0d 03 00 00 02 00 a7 00 0c
     01 03 00 33 00 34 00 04 29 37 00 01 01 29 38 00 13 65
     78 63 65 6c 73 77 69 74 63 68 69 6e 67 2e 63 6f 6d 00
     29 39 00 08 38 36 32 33 30 30 31 00 29 3a 00 06 31 32
      33 34 35 00]


3 -SENT To 192.168.1.51:5060 at 23597
SIP/2.0 401 Unauthorized
To: <sip:8623000@192.168.1.102>;tag=11635c2c
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
CSeq: 101 INVITE
WWW-Authenticate: Digest realm="excelswitching.com",
   nonce="42cac6967970048b000",
   opaque="asop19431163asdfj"
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK27519f66
User-Agent: Excel_CSP/83.10.57
Content-Length: 0
```

```
4 -RECEIVED From 192.168.1.51:52686 at 23597
ACK sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK27519f66
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
To: <sip:8623000@192.168.1.102>;tag=11635c2c
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
CSeq: 101 ACK
Content-Length: 0


5 -RECEIVED From 192.168.1.51:52678 at 23597
INVITE sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK34ee27fc
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
To: <sip:8623000@192.168.1.102>
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
CSeq: 102 INVITE
User-Agent: CSCO/7
Contact: <sip:8623001@192.168.1.51:5060>
Authorization: Digest
   username="8623001",realm="excelswitching.com",uri="sip
   :192.168.1.102",response="4c183f1b93f8c63b7052eed3cdfe
   daf8",nonce="42cac6967970048b000",opaque="asop19431163
   asdfj",algorithm=md5
Expires: 180
Content-Type: application/sdp
Content-Length: 191

v=0
o=Cisco-SIPUA 20722 17063 IN IP4 192.168.1.51
s=SIP Call
c=IN IP4 192.168.1.51
t=0 0
m=audio 30670 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000


6 -SENT To 192.168.1.51:5060 at 23597
SIP/2.0 100 Trying
To: <sip:8623000@192.168.1.102>;tag=11635c2c
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
```

```
CSeq: 102 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK34ee27fc
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [00 07 00 44 00 00 00 00 10]

H->X
    [00 73 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 44 00 05 27 7e 00
     03 08 00 00 29 19 00 08 38 36 32 33 30 30 30 00 29 1b
     00 0e 31 39 32 2e 31 36 38 2e 31 2e 32 32 30 00 29 23
     00 08 38 36 32 33 30 30 31 00 29 25 00 0d 31 39 32 2e
      31 36 38 2e 31 2e 35 31 00]


7 -SENT To 192.168.1.220:5060 at 23597
INVITE sip:8623000@192.168.1.220:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.220:5060>
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=195199955c2
   d
Call-ID: EXCEL-CSP0.79f.23597.450@192.168.1.102
Contact: 8623001<sip:8623001@192.168.1.102:5060>
User-Agent: Excel_CSP/83.10.57
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 15804 RTP/AVP 0


8 -RECEIVED From 192.168.1.220:5060 at 23597
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=195199955c2d
```

```
To: 8623000
   <sip:8623000@192.168.1.220:5060>;tag=345281520
Contact: <sip:8623000@192.168.1.220:5060>
Call-ID: EXCEL-CSP0.79f.23597.450@192.168.1.102
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


9 -RECEIVED From 192.168.1.220:5060 at 23597
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=195199955c2d
To: 8623000
   <sip:8623000@192.168.1.220:5060>;tag=345281520
Contact: <sip:8623000@192.168.1.220:5060>
Call-ID: EXCEL-CSP0.79f.23597.450@192.168.1.102
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


X->H
    [00 46 00 e8 00 00 00 00 10 02 02 1e 09 00 01 00 39 00
   03 00
     00 03 03 00 33 00 2d 00 01 29 50 00 27 45 58 43 45 4c
     2d 43 53 50 30 2e 37 39 66 2e 32 33 35 39 37 2e 34 35
      30 40 31 39 32 2e 31 36 38 2e 31 2e 31 30 32 00]

X->H
    [00 11 00 43 00 06 00 00 01 0d 03 00 00 03 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 06 00]

H->X
    [00 11 00 bf 00 00 ff 00 02 0d 03 00 00 02 0d 03 00 00
     02]


10-SENT To 192.168.1.51:5060 at 23597
SIP/2.0 180 Ringing
To: <sip:8623000@192.168.1.102>;tag=11635c2c
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
CSeq: 102 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
```

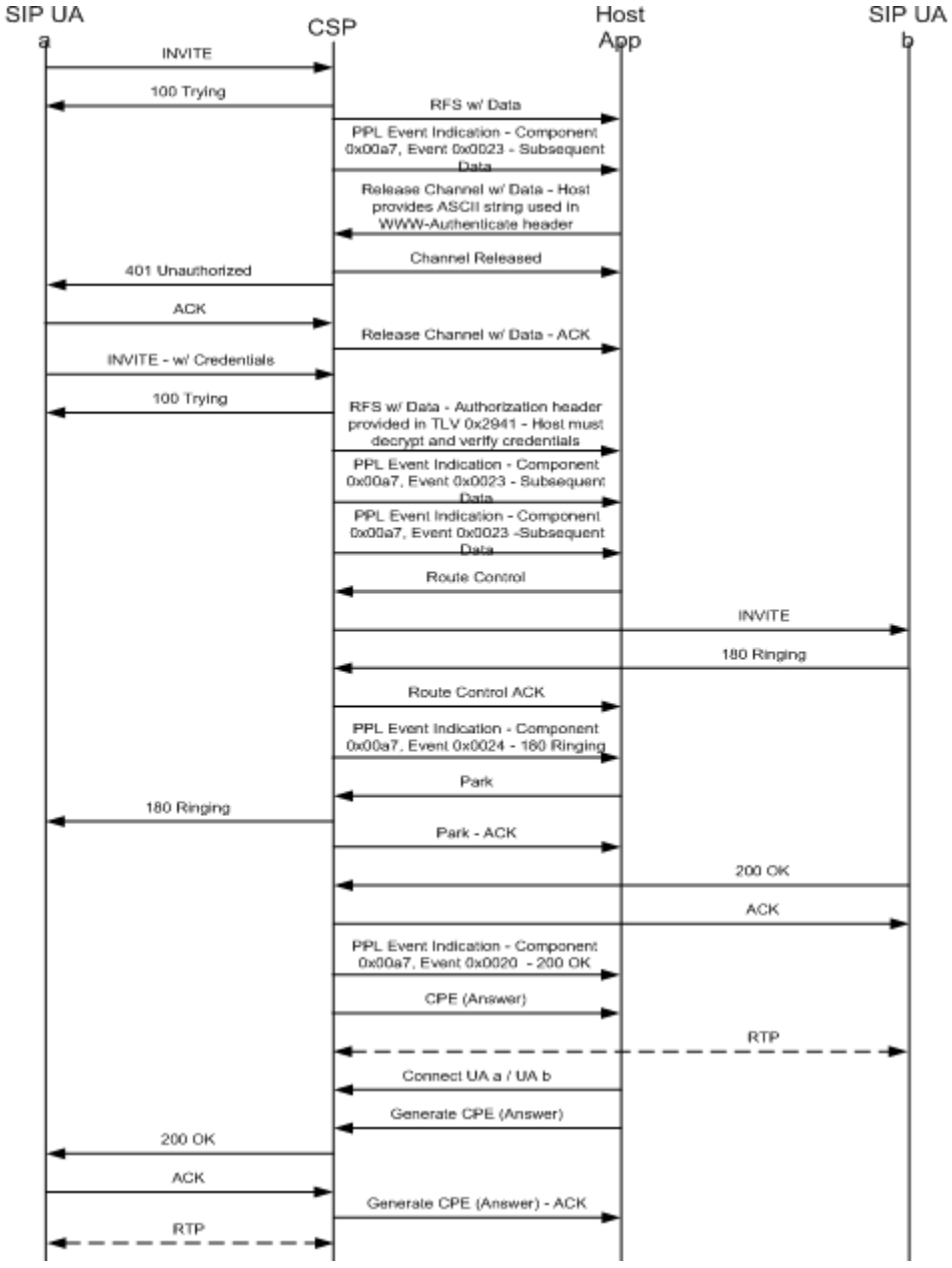
```
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK34ee27fc
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [00 07 00 bf 00 00 00 00 10]


11-RECEIVED From 192.168.1.220:5060 at 23599
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=195199955c2d
To: 8623000
   <sip:8623000@192.168.1.220:5060>;tag=345281520
Contact: <sip:8623000@192.168.1.220:5060>
Call-ID: EXCEL-CSP0.79f.23597.450@192.168.1.102
CSeq: 1 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 300

v=0
o=8623000 287016562 287018125 IN IP4 192.168.1.220
s=X-Lite
c=IN IP4 192.168.1.220
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15


12-SENT To 192.168.1.220:5060 at 23599
ACK sip:8623000@192.168.1.220:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.220:5060>;tag=345281520
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=195199955c2
   d
Call-ID: EXCEL-CSP0.79f.23597.450@192.168.1.102
CSeq: 1 ACK
Content-Length: 0
```

```
X->H
    [00 2e 00 43 00 07 00 00 01 0d 03 00 00 03 00 a7 00 20
   01 03
     00 33 00 18 00 03 27 94 00 04 c0 a8 01 dc 27 95 00 04
      00 00 1f 40 27 b0 00 02 00 02]

H->X
    [00 05 00 43 00 07 00]

X->H
    [00 0d 00 2e 00 01 00 00 01 0d 03 00 00 03 20]

H->X
    [00 05 00 2e 00 01 00]

H->X
    [00 11 00 00 00 00 ff 00 02 0d 03 00 00 02 0d 03 00 00
     03]

X->H
    [00 07 00 00 00 00 00 00 10]

H->X
    [00 0d 00 ba 00 00 ff 00 01 0d 03 00 00 02 01]

X->H
    [00 07 00 ba 00 00 00 00 10]


13-SENT To 192.168.1.51:5060 at 23599
SIP/2.0 200 OK
To: <sip:8623000@192.168.1.102>;tag=11635c2c
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
CSeq: 102 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK34ee27fc
User-Agent: Excel_CSP/83.10.57
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
```

```
m=audio 15772 RTP/AVP 0


14-RECEIVED From 192.168.1.51:52678 at 23599
ACK sip:8623000@192.168.1.102:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK693d8d04
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001206e974
   3c-19e08125
To: <sip:8623000@192.168.1.102>;tag=11635c2c
Call-ID: 00036b3c-25150015-242f51e7-4e0ac0b2@192.168.1.51
CSeq: 102 ACK
User-Agent: CSCO/7
Authorization: Digest
   username="8623001",realm="excelswitching.com",uri="sip
   :192.168.1.102",response="4c183f1b93f8c63b7052eed3cdfe
   daf8",nonce="42cac6967970048b000",opaque="asop19431163
   asdfj",algorithm=md5
Content-Length: 0
```

401 Authenticate Inbound Invite by Releasing Data

The following call flow shows an inbound INVITE message that gets authenticated using the Release Channel with Data message.



Host API/Signaling Trace

```
1 -RECEIVED From 192.168.1.51:52678 at 26587
INVITE sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK20deb67f
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
To: <sip:8623000@192.168.1.102>
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 101 INVITE
User-Agent: CSCO/7
Contact: <sip:8623001@192.168.1.51:5060>
Expires: 180
Content-Type: application/sdp
Content-Length: 191
Accept: application/sdp

v=0
o=Cisco-SIPUA 26417 26343 IN IP4 192.168.1.51
s=SIP Call
c=IN IP4 192.168.1.51
t=0 0
m=audio 30688 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000


2 -SENT To 192.168.1.51:5060 at 26587
SIP/2.0 100 Trying
To: <sip:8623000@192.168.1.102>;tag=688567db
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK20deb67f
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [01 33 00 2d 00 0f 00 00 01 0d 03 00 08 03 00 33 01 03
   00 33
     01 1f 00 18 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 38 36 32 33 30 30 30 00 29 1b 00 0e 31 39 32
     2e 31 36 38 2e 31 2e 31 30 32 00 29 1c 00 04 00 00 13
     c4 29 23 00 08 38 36 32 33 30 30 31 00 29 25 00 0e 31
     39 32 2e 31 36 38 2e 31 2e 31 30 32 00 29 26 00 04 00
```

```
     00 13 c4 29 2d 00 08 38 36 32 33 30 30 31 00 29 2f 00
     0d 31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 30 00 04
     00 00 13 c4 29 33 00 01 01 27 18 00 09 02 00 00 00 07
     86 23 00 10 27 17 00 07 02 00 07 86 23 00 00 27 94 00
     04 c0 a8 01 33 27 95 00 04 00 00 77 e0 27 b0 00 02 00
     02 27 b1 00 02 00 01 29 16 00 01 01 29 54 00 08 38 36
     32 33 30 30 30 00 29 55 00 0e 31 39 32 2e 31 36 38 2e
     31 2e 31 30 32 00 29 56 00 04 00 00 13 c4 29 50 00 31
     30 30 30 33 36 62 33 63 2d 32 35 31 35 30 30 31 65 2d
     36 66 35 33 39 34 62 38 2d 36 36 34 33 36 34 63 61 40
     31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 53 00 02 01
      00]

H->X
    [00 0c 00 2d 00 0f 00 00 01 0d 03 00 08 03]

X->H
    [00 59 00 43 00 1b 00 00 01 0d 03 00 08 03 00 a7 00 23
   01 03
     00 33 00 43 00 04 29 53 00 02 02 01 29 51 00 22 30 30
     30 33 36 62 33 63 32 35 31 35 30 30 31 62 30 63 30 36
     63 62 31 36 2d 33 31 62 64 66 39 33 61 00 29 52 00 09
      36 38 38 35 36 37 64 62 00 2a 0e 00 04 c0 a8 01 33]

H->X
    [00 05 00 43 00 1b 00]

H->X
    [00 7f 00 36 00 00 ff 00 02 0d 03 00 08 03 0d 03 00 08
     03 33 01 03 00 33 00 67 00 02 29 15 00 02 01 91 29 43
     00 5b 44 69 67 65 73 74 20 72 65 61 6c 6d 3d 22 65 78
     63 65 6c 73 77 69 74 63 68 69 6e 67 2e 63 6f 6d 22 2c
     20 6e 6f 6e 63 65 3d 22 34 32 63 61 63 36 39 36 37 39
     37 30 30 34 38 62 30 30 30 22 2c 20 6f 70 61 71 75 65
     3d 22 61 73 6f 70 31 39 34 33 31 31 36 33 61 73 64 66
      6a 22 00]


3 -SENT To 192.168.1.51:5060 at 26588
SIP/2.0 401 Unauthorized
To: <sip:8623000@192.168.1.102>;tag=688567db
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 101 INVITE
WWW-Authenticate: Digest realm="excelswitching.com",
   nonce="42cac6967970048b000",
   opaque="asop19431163asdfj"
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK20deb67f
```

```
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


4 -RECEIVED From 192.168.1.51:52697 at 26588
ACK sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK20deb67f
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
To: <sip:8623000@192.168.1.102>;tag=688567db
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 101 ACK
Content-Length: 0


X->H
    [00 07 00 36 00 00 00 00 10]

X->H
    [00 0c 00 49 00 0d 00 00 01 0d 03 00 08 03]

H->X
    [00 05 00 49 00 0d 00]


5 -RECEIVED From 192.168.1.51:52678 at 26588
INVITE sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK4c16f6c3
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
To: <sip:8623000@192.168.1.102>
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 102 INVITE
User-Agent: CSCO/7
Contact: <sip:8623001@192.168.1.51:5060>
Authorization: Digest
   username="8623001",realm="excelswitching.com",uri="sip
   :192.168.1.102",response="4c183f1b93f8c63b7052eed3cdfe
   daf8",nonce="42cac6967970048b000",opaque="asop19431163
   asdfj",algorithm=md5
Expires: 180
Content-Type: application/sdp
Content-Length: 191

v=0
o=Cisco-SIPUA 26417 26343 IN IP4 192.168.1.51
s=SIP Call
c=IN IP4 192.168.1.51
t=0 0
```

```
m=audio 30688 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000


6 -SENT To 192.168.1.51:5060 at 26588
SIP/2.0 100 Trying
To: <sip:8623000@192.168.1.102>;tag=688567dc
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 102 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK4c16f6c3
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [00 d8 00 2d 00 10 00 00 01 0d 03 00 00 10 00 33 01 03
   00 33
     00 c4 00 14 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 38 36 32 33 30 30 30 00 29 1b 00 0e 31 39 32
     2e 31 36 38 2e 31 2e 31 30 32 00 29 1c 00 04 00 00 13
     c4 29 23 00 08 38 36 32 33 30 30 31 00 29 25 00 0e 31
     39 32 2e 31 36 38 2e 31 2e 31 30 32 00 29 26 00 04 00
     00 13 c4 29 2d 00 08 38 36 32 33 30 30 31 00 29 2f 00
     0d 31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 30 00 04
     00 00 13 c4 29 33 00 01 01 27 18 00 09 02 00 00 00 07
     86 23 00 10 27 17 00 07 02 00 07 86 23 00 00 27 94 00
     04 c0 a8 01 33 27 95 00 04 00 00 77 e0 27 b0 00 02 00
     02 27 b1 00 02 00 01 29 16 00 01 01 29 53 00 02 01 00]

H->X
    [00 0c 00 2d 00 10 00 00 01 0d 03 00 00 10]

X->H
    [01 3b 00 43 00 1c 00 00 01 0d 03 00 00 10 00 a7 00 23
   01 03
     00 33 01 25 00 06 29 41 00 be 44 69 67 65 73 74 20 75
     73 65 72 6e 61 6d 65 3d 22 38 36 32 33 30 30 31 22 2c
     72 65 61 6c 6d 3d 22 65 78 63 65 6c 73 77 69 74 63 68
     69 6e 67 2e 63 6f 6d 22 2c 75 72 69 3d 22 73 69 70 3a
     31 39 32 2e 31 36 38 2e 31 2e 31 30 32 22 2c 72 65 73
     70 6f 6e 73 65 3d 22 34 63 31 38 33 66 31 62 39 33 66
     38 63 36 33 62 37 30 35 32 65 65 64 33 63 64 66 65 64
     61 66 38 22 2c 6e 6f 6e 63 65 3d 22 34 32 63 61 63 36
     39 36 37 39 37 30 30 34 38 62 30 30 30 22 2c 6f 70 61
```

```
     71 75 65 3d 22 61 73 6f 70 31 39 34 33 31 31 36 33 61
     73 64 66 6a 22 2c 61 6c 67 6f 72 69 74 68 6d 3d 6d 64
     35 00 29 54 00 08 38 36 32 33 30 30 30 00 29 55 00 0e
     31 39 32 2e 31 36 38 2e 31 2e 31 30 32 00 29 56 00 04
     00 00 13 c4 29 50 00 31 30 30 30 33 36 62 33 63 2d 32
     35 31 35 30 30 31 65 2d 36 66 35 33 39 34 62 38 2d 36
     36 34 33 36 34 63 61 40 31 39 32 2e 31 36 38 2e 31 2e
      35 31 00 29 53 00 02 02 00]

H->X
    [00 05 00 43 00 1c 00]

X->H
    [00 59 00 43 00 1d 00 00 01 0d 03 00 00 10 00 a7 00 23
   01 03
     00 33 00 43 00 04 29 53 00 02 03 01 29 51 00 22 30 30
     30 33 36 62 33 63 32 35 31 35 30 30 31 62 30 63 30 36
     63 62 31 36 2d 33 31 62 64 66 39 33 61 00 29 52 00 09
      36 38 38 35 36 37 64 63 00 2a 0e 00 04 c0 a8 01 33]

H->X
    [00 05 00 43 00 1d 00]

H->X
    [00 73 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 44 00 05 27 7e 00
     03 08 00 00 29 19 00 08 38 36 32 33 30 30 30 00 29 1b
     00 0e 31 39 32 2e 31 36 38 2e 31 2e 32 32 30 00 29 23
     00 08 38 36 32 33 30 30 31 00 29 25 00 0d 31 39 32 2e
      31 36 38 2e 31 2e 35 31 00]


7 -SENT To 192.168.1.220:5060 at 26589
INVITE sip:8623000@192.168.1.220:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.220:5060>
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=1551590867d
   d
Call-ID: EXCEL-CSP0.60f.26589.960@192.168.1.102
Contact: 8623001<sip:8623001@192.168.1.102:5060>
User-Agent: Excel_CSP/83.10.57
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 105
```

```
v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 14204 RTP/AVP 0


8 -RECEIVED From 192.168.1.220:5060 at 26589
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1551590867dd
To: 8623000
   <sip:8623000@192.168.1.220:5060>;tag=224531354
Contact: <sip:8623000@192.168.1.220:5060>
Call-ID: EXCEL-CSP0.60f.26589.960@192.168.1.102
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


9 -RECEIVED From 192.168.1.220:5060 at 26590
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1551590867dd
To: 8623000
   <sip:8623000@192.168.1.220:5060>;tag=224531354
Contact: <sip:8623000@192.168.1.220:5060>
Call-ID: EXCEL-CSP0.60f.26589.960@192.168.1.102
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


X->H
    [00 46 00 e8 00 00 00 00 10 02 02 1e 09 00 01 00 39 00
   03 00
     00 11 03 00 33 00 2d 00 01 29 50 00 27 45 58 43 45 4c
     2d 43 53 50 30 2e 36 30 66 2e 32 36 35 38 39 2e 39 36
      30 40 31 39 32 2e 31 36 38 2e 31 2e 31 30 32 00]

X->H
    [00 11 00 43 00 1e 00 00 01 0d 03 00 00 11 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 1e 00]

H->X
```

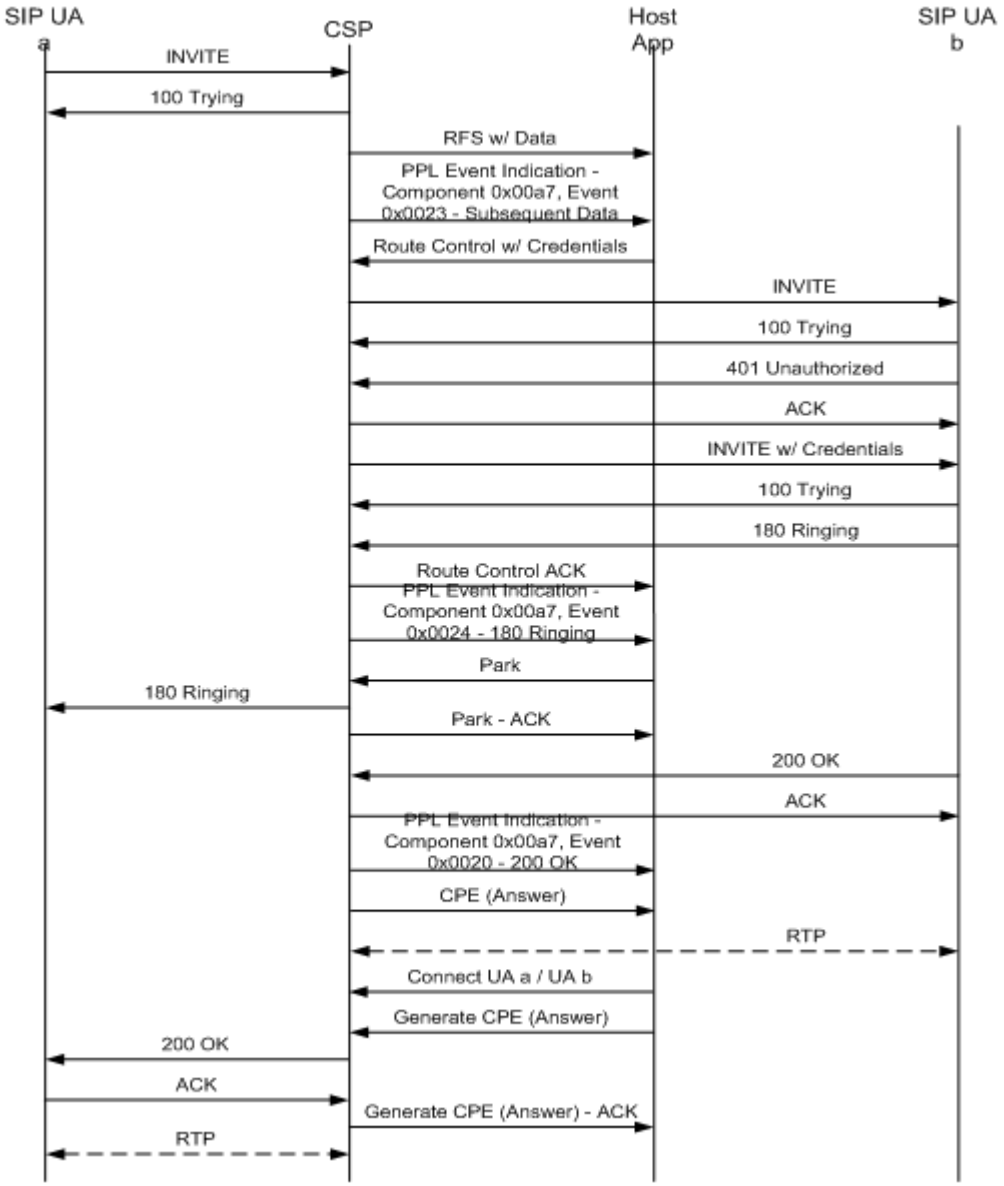
```
    [00 11 00 bf 00 00 ff 00 02 0d 03 00 00 10 0d 03 00 00
     10]


10-SENT To 192.168.1.51:5060 at 26590
SIP/2.0 180 Ringing
To: <sip:8623000@192.168.1.102>;tag=688567dc
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 102 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK4c16f6c3
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [00 07 00 bf 00 00 00 00 10]


11-RECEIVED From 192.168.1.220:5060 at 26593
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.1.102
From: 8623001
   <sip:8623001@192.168.1.51:5060>;tag=1551590867dd
To: 8623000
   <sip:8623000@192.168.1.220:5060>;tag=224531354
Contact: <sip:8623000@192.168.1.220:5060>
Call-ID: EXCEL-CSP0.60f.26589.960@192.168.1.102
CSeq: 1 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 300

v=0
o=8623000 290009000 290012046 IN IP4 192.168.1.220
s=X-Lite
c=IN IP4 192.168.1.220
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

```
12-SENT To 192.168.1.220:5060 at 26593
ACK sip:8623000@192.168.1.220:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.220:5060>;tag=224531354
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=1551590867d
   d
Call-ID: EXCEL-CSP0.60f.26589.960@192.168.1.102
CSeq: 1 ACK
Content-Length: 0


X->H
    [00 2e 00 43 00 1f 00 00 01 0d 03 00 00 11 00 a7 00 20
   01 03
     00 33 00 18 00 03 27 94 00 04 c0 a8 01 dc 27 95 00 04
      00 00 1f 40 27 b0 00 02 00 02]

H->X
    [00 05 00 43 00 1f 00]

X->H
    [00 0d 00 2e 00 04 00 00 01 0d 03 00 00 11 20]

H->X
    [00 05 00 2e 00 04 00]

H->X
    [00 11 00 00 00 00 ff 00 02 0d 03 00 00 10 0d 03 00 00
     11]

X->H
    [00 07 00 00 00 00 00 00 10]

H->X
    [00 0d 00 ba 00 00 ff 00 01 0d 03 00 00 10 01]

X->H
    [00 07 00 ba 00 00 00 00 10]


13-SENT To 192.168.1.51:5060 at 26593
SIP/2.0 200 OK
To: <sip:8623000@192.168.1.102>;tag=688567dc
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 102 INVITE
```

```
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK4c16f6c3
User-Agent: Excel_CSP/83.10.57
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 14172 RTP/AVP 0

14-RECEIVED From 192.168.1.51:52678 at 26593
ACK sip:8623000@192.168.1.102:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK64b1d90a
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515001b0c06cb
   16-31bdf93a
To: <sip:8623000@192.168.1.102>;tag=688567dc
Call-ID: 00036b3c-2515001e-6f5394b8-664364ca@192.168.1.51
CSeq: 102 ACK
User-Agent: CSCO/7
Authorization: Digest
   username="8623001",realm="excelswitching.com",uri="sip
   :192.168.1.102",response="4c183f1b93f8c63b7052eed3cdfe
   daf8",nonce="42cac6967970048b000",opaque="asop19431163
   asdfj",algorithm=md5
Content-Length: 0
```

401 Authenticate for Outbound Invite

The following call flow shows an outbound INVITE message receiving a 401 Authenticate response.



Host API/Signaling Trace

```
1 -RECEIVED From 192.168.1.51:52678 at 14674
INVITE sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK658c0840
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c25150028421f63
   3a-68d591de
To: <sip:8623000@192.168.1.102>
Call-ID: 00036b3c-2515002a-3af1874b-4da5b6ed@192.168.1.51
CSeq: 101 INVITE
User-Agent: CSCO/7
Contact: <sip:8623001@192.168.1.51:5060>
Expires: 180
Content-Type: application/sdp
Content-Length: 190
Accept: application/sdp

v=0
o=Cisco-SIPUA 5235 22144 IN IP4 192.168.1.51
s=SIP Call
c=IN IP4 192.168.1.51
t=0 0
m=audio 30712 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000


2 -SENT To 192.168.1.51:5060 at 14674
SIP/2.0 100 Trying
To: <sip:8623000@192.168.1.102>;tag=42303952
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c25150028421f63
   3a-68d591de
Call-ID: 00036b3c-2515002a-3af1874b-4da5b6ed@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK658c0840
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [01 33 00 2d 00 0b 00 00 01 0d 03 00 00 11 00 33 01 03
   00 33
     01 1f 00 18 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 38 36 32 33 30 30 30 00 29 1b 00 0e 31 39 32
     2e 31 36 38 2e 31 2e 31 30 32 00 29 1c 00 04 00 00 13
     c4 29 23 00 08 38 36 32 33 30 30 31 00 29 25 00 0e 31
     39 32 2e 31 36 38 2e 31 2e 31 30 32 00 29 26 00 04 00
```

```
     00 13 c4 29 2d 00 08 38 36 32 33 30 30 31 00 29 2f 00
     0d 31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 30 00 04
     00 00 13 c4 29 33 00 01 01 27 18 00 09 02 00 00 00 07
     86 23 00 10 27 17 00 07 02 00 07 86 23 00 00 27 94 00
     04 c0 a8 01 33 27 95 00 04 00 00 77 f8 27 b0 00 02 00
     02 27 b1 00 02 00 01 29 16 00 01 01 29 54 00 08 38 36
     32 33 30 30 30 00 29 55 00 0e 31 39 32 2e 31 36 38 2e
     31 2e 31 30 32 00 29 56 00 04 00 00 13 c4 29 50 00 31
     30 30 30 33 36 62 33 63 2d 32 35 31 35 30 30 32 61 2d
     33 61 66 31 38 37 34 62 2d 34 64 61 35 62 36 65 64 40
     31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 53 00 02 01
      00]

H->X
    [00 0c 00 2d 00 0b 00 00 01 0d 03 00 00 11]

X->H
    [00 59 00 43 00 11 00 00 01 0d 03 00 00 11 00 a7 00 23
   01 03
     00 33 00 43 00 04 29 53 00 02 02 01 29 51 00 22 30 30
     30 33 36 62 33 63 32 35 31 35 30 30 32 38 34 32 31 66
     36 33 33 61 2d 36 38 64 35 39 31 64 65 00 29 52 00 09
      34 32 33 30 33 39 35 32 00 2a 0e 00 04 c0 a8 01 33]

H->X
    [00 05 00 43 00 11 00]

H->X
    [00 89 00 e8 00 00 00 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 5a 00 05 27 7e 00
     03 08 00 00 29 19 00 08 38 36 32 33 30 30 30 00 29 1b
     00 0e 31 39 32 2e 31 36 38 2e 31 2e 32 30 31 00 29 23
     00 08 38 36 32 33 30 30 31 00 29 25 00 0d 31 39 32 2e
     31 36 38 2e 31 2e 35 31 00 29 3b 00 08 38 36 32 33 30
      30 31 00 29 3c 00 06 31 32 33 34 35 00]


3 -SENT To 192.168.1.201:5060 at 14675
INVITE sip:8623000@192.168.1.201:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.201:5060>
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
Contact: 8623001<sip:8623001@192.168.1.102:5060>
User-Agent: Excel_CSP/83.10.57
Supported: timer
Session-Expires: 1800
```

```
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 14236 RTP/AVP 0


4 -RECEIVED From 192.168.1.201:1047 at 14675
SIP/2.0 100 Trying
To: 8623000<sip:8623000@192.168.1.201:5060>;tag=9668e0
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
CSeq: 1 INVITE
Contact: 8623000<sip:8623000@192.168.1.201:5060>
Via: SIP/2.0/UDP 192.168.1.102
User-Agent: Excel_CSP/82.30.134
Content-Length: 0


5 -RECEIVED From 192.168.1.201:1047 at 14675
SIP/2.0 401 Unauthorized
To: 8623000<sip:8623000@192.168.1.201:5060>;tag=9668e0
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
CSeq: 1 INVITE
WWW-Authenticate: Digest realm="excelswitching.com",
   nonce="42cbde587025c4000", opaque="asop79668asdfj"
Via: SIP/2.0/UDP 192.168.1.102
User-Agent: Excel_CSP/82.30.134
Content-Length: 0


6 -SENT To 192.168.1.201:5060 at 14675
ACK sip:8623000@192.168.1.201:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
CSeq: 1 ACK
To: 8623000<sip:8623000@192.168.1.201:5060>;tag=9668e0
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
```

```
Content-Length: 0


7 -SENT To 192.168.1.201:5060 at 14675
INVITE sip:8623000@192.168.1.201:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.201:5060>
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
CSeq: 2 INVITE
User-Agent: Excel_CSP/83.10.57
Contact: 8623001<sip:8623001@192.168.1.102:5060>
Authorization: Digest realm="excelswitching.com",
   nonce="42cbde587025c4000", opaque="asop79668asdfj",
   username="8623001",
   response="d26f5e9f9b4adfb4e513e73e2cfd389b",
   uri="sip:8623000@192.168.1.201:5060"
Supported: timer
Session-Expires: 1800
Min-SE: 300
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 14236 RTP/AVP 0


8 -RECEIVED From 192.168.1.201:1047 at 14675
SIP/2.0 100 Trying
To: 8623000<sip:8623000@192.168.1.201:5060>;tag=9668e0
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
CSeq: 2 INVITE
Contact: 8623000<sip:8623000@192.168.1.201:5060>
Via: SIP/2.0/UDP 192.168.1.102
User-Agent: Excel_CSP/82.30.134
Content-Length: 0


9 -RECEIVED From 192.168.1.201:1047 at 14675
SIP/2.0 180 Ringing
To: 8623000<sip:8623000@192.168.1.201:5060>;tag=9668e0
```

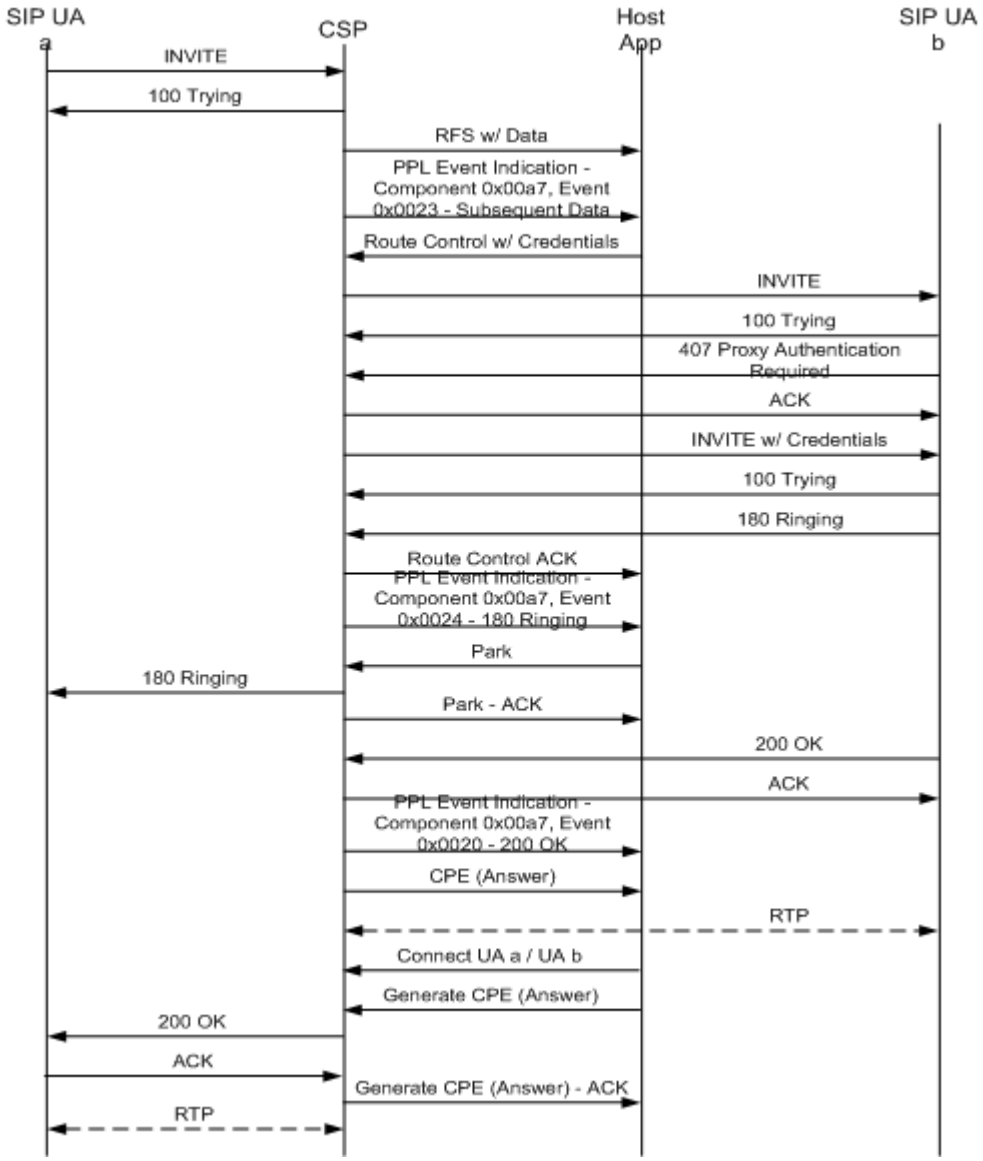
```
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
CSeq: 2 INVITE
Contact: 8623000<sip:8623000@192.168.1.201:5060>
Via: SIP/2.0/UDP 192.168.1.102
User-Agent: Excel_CSP/82.30.134
Content-Length: 0


X->H
    [00 46 00 e8 00 00 00 00 10 02 02 1e 09 00 01 00 39 00
   03 00
     00 12 03 00 33 00 2d 00 01 29 50 00 27 45 58 43 45 4c
     2d 43 53 50 30 2e 36 31 37 2e 31 34 36 37 35 2e 33 36
      30 40 31 39 32 2e 31 36 38 2e 31 2e 31 30 32 00]

X->H
    [00 11 00 43 00 12 00 00 01 0d 03 00 00 12 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 12 00]

H->X
    [00 11 00 bf 00 00 00 00 02 0d 03 00 00 11 0d 03 00 00
     11]


10-SENT To 192.168.1.51:5060 at 14675
SIP/2.0 180 Ringing
To: <sip:8623000@192.168.1.102>;tag=42303952
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c25150028421f63
   3a-68d591de
Call-ID: 00036b3c-2515002a-3af1874b-4da5b6ed@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK658c0840
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [00 07 00 bf 00 00 00 00 10]


11-RECEIVED From 192.168.1.201:1047 at 14677
SIP/2.0 200 OK
To: 8623000<sip:8623000@192.168.1.201:5060>;tag=9668e0
```

```
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
CSeq: 2 INVITE
Contact: 8623000<sip:8623000@192.168.1.201:5060>
Require: timer
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 192.168.1.102
User-Agent: Excel_CSP/82.30.134
Content-Type: application/sdp
Content-Length: 104

v=0
o=sip 0 0 IN IP4 192.168.1.201
s=SIP_Call
c=IN IP4 192.168.1.232
t=0 0
m=audio 8028 RTP/AVP 0


12-SENT To 192.168.1.201:5060 at 14678
ACK sip:8623000@192.168.1.201:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.201:5060>;tag=9668e0
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15599668395
   3
Call-ID: EXCEL-CSP0.617.14675.360@192.168.1.102
CSeq: 2 ACK
Content-Length: 0


X->H
    [00 2e 00 43 00 13 00 00 01 0d 03 00 00 12 00 a7 00 20
   01 03
     00 33 00 18 00 03 27 94 00 04 c0 a8 01 e8 27 95 00 04
      00 00 1f 5c 27 b0 00 02 00 02]

H->X
    [00 05 00 43 00 13 00]

X->H
    [00 0d 00 2e 00 03 00 00 01 0d 03 00 00 12 20]

H->X
    [00 05 00 2e 00 03 00]

H->X
```

```
    [00 11 00 00 00 00 00 00 02 0d 03 00 00 11 0d 03 00 00
     12]

X->H
    [00 07 00 00 00 00 00 00 10]

H->X
    [00 0d 00 ba 00 00 00 00 01 0d 03 00 00 11 01]

X->H
    [00 07 00 ba 00 00 00 00 10]


13-SENT To 192.168.1.51:5060 at 14678
SIP/2.0 200 OK
To: <sip:8623000@192.168.1.102>;tag=42303952
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c25150028421f63
   3a-68d591de
Call-ID: 00036b3c-2515002a-3af1874b-4da5b6ed@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK658c0840
User-Agent: Excel_CSP/83.10.57
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 14204 RTP/AVP 0

14-RECEIVED From 192.168.1.51:52678 at 14678
ACK sip:8623000@192.168.1.102:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK42290e92
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c25150028421f63
   3a-68d591de
To: <sip:8623000@192.168.1.102>;tag=42303952
Call-ID: 00036b3c-2515002a-3af1874b-4da5b6ed@192.168.1.51
CSeq: 101 ACK
User-Agent: CSCO/7
Content-Length: 0
```

407 Proxy Authenticate for Outbound INVITE

The following call flows shows the outbound INVITE message receiving a 407 Proxy Authenticate response.



Host API/Signaling Trace

```
1 -RECEIVED From 192.168.1.51:52678 at 13883
INVITE sip:8623000@192.168.1.102 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK78bfb5a0
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515002729a42d
   d3-51a1a17a
To: <sip:8623000@192.168.1.102>
Call-ID: 00036b3c-25150029-7ac0682f-2f71a3f1@192.168.1.51
CSeq: 101 INVITE
User-Agent: CSCO/7
Contact: <sip:8623001@192.168.1.51:5060>
Expires: 180
Content-Type: application/sdp
Content-Length: 190
Accept: application/sdp

v=0
o=Cisco-SIPUA 12170 6068 IN IP4 192.168.1.51
s=SIP Call
c=IN IP4 192.168.1.51
t=0 0
m=audio 30710 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000


2 -SENT To 192.168.1.51:5060 at 13883
SIP/2.0 100 Trying
To: <sip:8623000@192.168.1.102>;tag=4788363b
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515002729a42d
   d3-51a1a17a
Call-ID: 00036b3c-25150029-7ac0682f-2f71a3f1@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK78bfb5a0
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [01 33 00 2d 00 0a 00 00 01 0d 03 00 08 03 00 33 01 03
   00 33
     01 1f 00 18 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 08 38 36 32 33 30 30 30 00 29 1b 00 0e 31 39 32
     2e 31 36 38 2e 31 2e 31 30 32 00 29 1c 00 04 00 00 13
     c4 29 23 00 08 38 36 32 33 30 30 31 00 29 25 00 0e 31
     39 32 2e 31 36 38 2e 31 2e 31 30 32 00 29 26 00 04 00
```

```
     00 13 c4 29 2d 00 08 38 36 32 33 30 30 31 00 29 2f 00
     0d 31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 30 00 04
     00 00 13 c4 29 33 00 01 01 27 18 00 09 02 00 00 00 07
     86 23 00 10 27 17 00 07 02 00 07 86 23 00 00 27 94 00
     04 c0 a8 01 33 27 95 00 04 00 00 77 f6 27 b0 00 02 00
     02 27 b1 00 02 00 01 29 16 00 01 01 29 54 00 08 38 36
     32 33 30 30 30 00 29 55 00 0e 31 39 32 2e 31 36 38 2e
     31 2e 31 30 32 00 29 56 00 04 00 00 13 c4 29 50 00 31
     30 30 30 33 36 62 33 63 2d 32 35 31 35 30 30 32 39 2d
     37 61 63 30 36 38 32 66 2d 32 66 37 31 61 33 66 31 40
     31 39 32 2e 31 36 38 2e 31 2e 35 31 00 29 53 00 02 01
      00]

H->X
    [00 0c 00 2d 00 0a 00 00 01 0d 03 00 08 03]

X->H
    [00 59 00 43 00 0e 00 00 01 0d 03 00 08 03 00 a7 00 23
   01 03
     00 33 00 43 00 04 29 53 00 02 02 01 29 51 00 22 30 30
     30 33 36 62 33 63 32 35 31 35 30 30 32 37 32 39 61 34
     32 64 64 33 2d 35 31 61 31 61 31 37 61 00 29 52 00 09
      34 37 38 38 33 36 33 62 00 2a 0e 00 04 c0 a8 01 33]

H->X
    [00 05 00 43 00 0e 00]

H->X
    [00 89 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 5a 00 05 27 7e 00
     03 08 00 00 29 19 00 08 38 36 32 33 30 30 30 00 29 1b
     00 0e 31 39 32 2e 31 36 38 2e 31 2e 30 32 30 00 29 23
     00 08 38 36 32 33 30 30 31 00 29 25 00 0d 31 39 32 2e
     31 36 38 2e 31 2e 35 31 00 29 3d 00 08 38 36 32 33 30
      30 31 00 29 3e 00 06 31 32 33 34 35 00]


3 -SENT To 192.168.1.20:5060 at 13884
INVITE sip:8623000@192.168.1.020:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.020:5060>
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15434937363
   c
Call-ID: EXCEL-CSP0.607.13884.820@192.168.1.102
Contact: 8623001<sip:8623001@192.168.1.102:5060>
User-Agent: Excel_CSP/83.10.57
Supported: timer
Session-Expires: 1800
```

```
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 14172 RTP/AVP 0


4 -RECEIVED From 192.168.1.20:5060 at 13884
SIP/2.0 407 Proxy Authentication Required
Via: SIP/2.0/UDP 192.168.1.102
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15434937363
   c
To:
   8623000<sip:8623000@192.168.1.020:5060>;tag=as40973a8e
Call-ID: EXCEL-CSP0.607.13884.820@192.168.1.102
CSeq: 1 INVITE
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER
Contact: <sip:8623000@192.168.1.20>
Proxy-Authenticate: Digest realm="asterisk",
   nonce="4fb1add8"
Content-Length: 0


5 -SENT To 192.168.1.20:5060 at 13884
ACK sip:8623000@192.168.1.020:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
Call-ID: EXCEL-CSP0.607.13884.820@192.168.1.102
CSeq: 1 ACK
To:
   8623000<sip:8623000@192.168.1.020:5060>;tag=as40973a8e
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15434937363
   c
Content-Length: 0


6 -SENT To 192.168.1.20:5060 at 13884
INVITE sip:8623000@192.168.1.020:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To: 8623000<sip:8623000@192.168.1.020:5060>
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15434937363
   c
```

```
Call-ID: EXCEL-CSP0.607.13884.820@192.168.1.102
CSeq: 2 INVITE
User-Agent: Excel_CSP/83.10.57
Contact: 8623001<sip:8623001@192.168.1.102:5060>
Proxy-Authorization: Digest realm="asterisk",
   nonce="4fb1add8", username="8623001",
   response="6a9a8a79460d5834f228069eb1399dae",
   uri="sip:8623000@192.168.1.020:5060"
Supported: timer
Session-Expires: 1800
Min-SE: 300
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.131
t=0 0
m=audio 14172 RTP/AVP 0


7 -RECEIVED From 192.168.1.20:5060 at 13884
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.102
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15434937363
   c
To: 8623000<sip:8623000@192.168.1.020:5060>
Call-ID: EXCEL-CSP0.607.13884.820@192.168.1.102
CSeq: 2 INVITE
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER
Contact: <sip:8623000@192.168.1.20>
Content-Length: 0


8 -RECEIVED From 192.168.1.20:5060 at 13884
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.102
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15434937363
   c
To:
   8623000<sip:8623000@192.168.1.020:5060>;tag=as1a9537b4
Call-ID: EXCEL-CSP0.607.13884.820@192.168.1.102
CSeq: 2 INVITE
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER
Contact: <sip:8623000@192.168.1.20>
Content-Length: 0
```

```
X->H
    [00 46 00 e8 00 00 00 00 10 02 02 1e 09 00 01 00 39 00
   03 00
     00 10 03 00 33 00 2d 00 01 29 50 00 27 45 58 43 45 4c
     2d 43 53 50 30 2e 36 30 37 2e 31 33 38 38 34 2e 38 32
      30 40 31 39 32 2e 31 36 38 2e 31 2e 31 30 32 00]

X->H
    [00 11 00 43 00 0f 00 00 01 0d 03 00 00 10 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 0f 00]

H->X
    [00 11 00 bf 00 00 ff 00 02 0d 03 00 08 03 0d 03 00 08
     03]


9 -SENT To 192.168.1.51:5060 at 13885
SIP/2.0 180 Ringing
To: <sip:8623000@192.168.1.102>;tag=4788363b
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515002729a42d
   d3-51a1a17a
Call-ID: 00036b3c-25150029-7ac0682f-2f71a3f1@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK78bfb5a0
User-Agent: Excel_CSP/83.10.57
Content-Length: 0


X->H
    [00 07 00 bf 00 00 00 00 10]


10-RECEIVED From 192.168.1.20:5060 at 13888
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.102
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15434937363
   c
To:
   8623000<sip:8623000@192.168.1.020:5060>;tag=as1a9537b4
Call-ID: EXCEL-CSP0.607.13884.820@192.168.1.102
CSeq: 2 INVITE
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER
```

```
Contact: <sip:8623000@192.168.1.20>
Content-Type: application/sdp
Content-Length: 207

v=0
o=root 16400 16400 IN IP4 192.168.1.20
s=session
c=IN IP4 192.168.1.20
t=0 0
m=audio 12026 RTP/AVP 3 0 8
a=rtpmap:3 GSM/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=silenceSupp:off - - - -


11-SENT To 192.168.1.20:5060 at 13888
ACK sip:8623000@192.168.1.20:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.102
To:
   8623000<sip:8623000@192.168.1.020:5060>;tag=as1a9537b4
From:
   8623001<sip:8623001@192.168.1.51:5060>;tag=15434937363
   c
Call-ID: EXCEL-CSP0.607.13884.820@192.168.1.102
CSeq: 2 ACK
Content-Length: 0


X->H
    [00 2e 00 43 00 10 00 00 01 0d 03 00 00 10 00 a7 00 20
   01 03
     00 33 00 18 00 03 27 94 00 04 c0 a8 01 14 27 95 00 04
      00 00 2e fa 27 b0 00 02 00 02]

H->X
    [00 05 00 43 00 10 00]

X->H
    [00 0d 00 2e 00 02 00 00 01 0d 03 00 00 10 20]

H->X
    [00 05 00 2e 00 02 00]

H->X
    [00 11 00 00 00 00 ff 00 02 0d 03 00 08 03 0d 03 00 00
      10]

X->H
    [00 07 00 00 00 00 00 00 10]
```

```
H->X
    [00 0d 00 ba 00 00 ff 00 01 0d 03 00 08 03 01]

X->H
    [00 07 00 ba 00 00 00 00 10]


12-SENT To 192.168.1.51:5060 at 13888
SIP/2.0 200 OK
To: <sip:8623000@192.168.1.102>;tag=4788363b
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515002729a42d
   d3-51a1a17a
Call-ID: 00036b3c-25150029-7ac0682f-2f71a3f1@192.168.1.51
CSeq: 101 INVITE
Contact: 8623000<sip:8623000@192.168.1.102:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK78bfb5a0
User-Agent: Excel_CSP/83.10.57
Content-Type: application/sdp
Content-Length: 105

v=0
o=sip 0 0 IN IP4 192.168.1.102
s=SIP_Call
c=IN IP4 192.168.1.132
t=0 0
m=audio 15800 RTP/AVP 0


13-RECEIVED From 192.168.1.51:52678 at 13888
ACK sip:8623000@192.168.1.102:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.51:5060;branch=z9hG4bK63f2911e
From: "8623001"
   <sip:8623001@192.168.1.102>;tag=00036b3c2515002729a42d
   d3-51a1a17a
To: <sip:8623000@192.168.1.102>;tag=4788363b
Call-ID: 00036b3c-25150029-7ac0682f-2f71a3f1@192.168.1.51
CSeq: 101 ACK
User-Agent: CSCO/7
Content-Length: 0
```

Host Control Method of SIP 180 Provisional Response Generation

Overview

Prior to this feature, the CSP delivered a SIP 180 Ringing Response to the calling SIP endpoint as a result of the host application sending a *Park Channel* or *Connect* message. At its discretion, the host application could initiate the 180 Response by sending a *Park Channel* message to the calling endpoint's span/channel at any time prior to issuing the *Connect* message. If the *Park Channel* message was not sent and the host application issued the *Connect* message, the CSP automatically generated the 180 Response after receiving alerting or an answer from the called party. In either case, the CSP always sent a 180 Ringing response to the SIP calling endpoint.

Description

With the Host Control Method of SIP 180 Provisional Response Generation feature, developers can use the Alerting Propagation Mode TLV (0x0117) in the 0x1E PPL Generic ICB in the following messages. Sending this TLV prevents the CSP from automatically generating the 180 Response to the calling SIP endpoint.

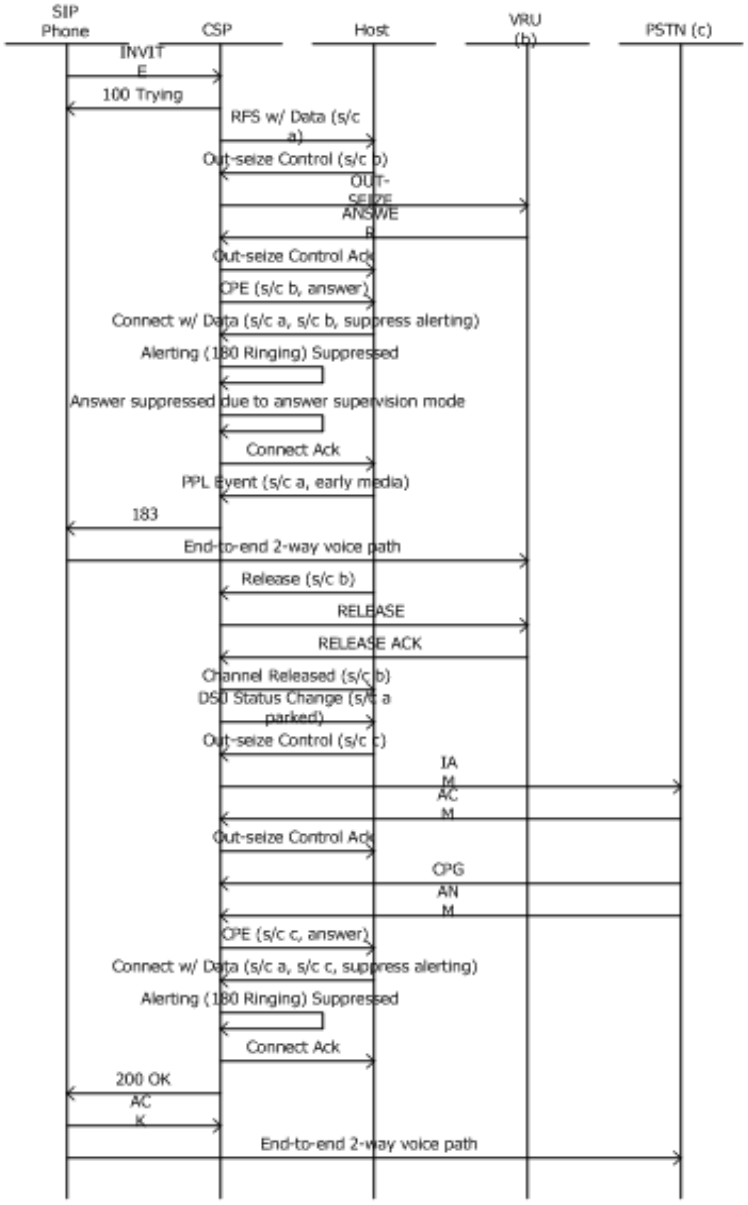
- PPL Event Request for Answer in L4 CH component
- *Connect with Data* (0x0005) message

API Messages Used

- *PPL Event Request* (0x0044)
- *Connect with Data* (0x0005)

Call Flow

The following call flow shows the SIP 180 Provisional Response Generation feature in an environment with a Voice Recognition Unit (VRU). Note that the SIP 180 response is suppressed for this call.


SIP Phone
CSP
Host
VRU (b)
PSTN (c)
INVITE
100 Trying
RFS w/ Data (s/c a)
Out-seize Control (s/c b)
OUT-SEIZE
ANSWER
Out-seize Control Ack
CPE (s/c b, answer)
Connect w/ Data (s/c a, s/c b, suppress alerting)
Alerting (180 Ringing) Suppressed
Answer suppressed due to answer supervision mode
Connect Ack
PPL Event (s/c a, early media)
183
End-to-end 2-way voice path
Release (s/c b)
RELEASE
RELEASE ACK
Channel Released (s/c b)
DSO Status Change (s/c a parked)
Out-seize Control (s/c c)
IAM
ACM
Out-seize Control Ack
CPG
ANM
CPE (s/c c, answer)
Connect w/ Data (s/c a, s/c c, suppress alerting)
Alerting (180 Ringing) Suppressed
Connect Ack
200 OK
ACK
End-to-end 2-way voice path

RFC 2833 (DTMF Digits) Dynamic Payload Negotiation

Overview

The SIP software supports RFC 2833 dynamic payload type negotiation between originating and terminating SIP gateways on a per call basis.

Pertinent Specification

RFC 2833

Description

The following applies to this feature:

- RFC 2833 is disabled by default. You should enable it before running calls.
- The SIP software supports RFC 2833 dynamic payload type negotiation in Session Description Protocol (SDP) within the SIP software.
- For outbound SIP Calls, the SIP software provides the RFC 2833 dynamic payload type in SDP as provided by the host application. If this information is not in the *Outseize Control* or *Route Control* messages, the SDP provides whatever was configured on the IP Network Interface Series 2 card.
- The host application can also provide the dynamic payload type for VoIP clear channel calls. It does not matter if the SIP software or the host application controls the calls.
- For in-bound SIP calls, the SIP software will propagate the RFC 2833 dynamic payload type to the IP Network Interface Series 2 card on a per connection basis.
- The SIP software can receive SIP RE-INVITE messages from the network that change the RFC 2833 dynamic payload type. The SIP software can not initiate a SIP RE-INVITE message to change RFC 2833 dynamic payload types.
- You can still configure the RFC 2833 Dynamic Payload Type when you configure the SIP software with the *Resource Attribute Configure message* (0x00E3).

API Messages

- *Route Control* (0x00E8)
- *Outseize Control* (0x002C)

Configuring

Follow the sections below to implement this feature in your host application.

Configuring the Default 2833 Payload Type on IPN Series 2

Use the following TLV to configure the default RFC 2833 Payload type on the Network Interface Series 2 cards.

Include the RFC 2833 Dynamic Payload Type TLV (0x01F1) in the *Resource Attribute Configure* message to configure the default payload type on a per module basis.

Changing the Default RFC 2833 Codec Value for Outbound Calls on IPN Series 2

Use the RFC 2833 Dynamic Payload Type TLV in the 0x001E PPL Generic ICB in either the *Route Control* or *Outseize Control* message to change the default RFC 2833 payload type.

Enabling or Disabling Digit Relay Enable

Use the RFC 2833 Enable TLV (0x01E2) in the 0x001E PPL Generic ICB in either the *Outseize Control*, *Route Control*, or *Resource Attribute Configure* message to enable or disable RFC 2833 support.

Querying Default RFC 2833 Payload Type

Use the VoIP Terminal Capabilities (0x01EA) TLV or the 0x01F1 RFC 2833 Dynamic Payload Type TLV in the *Resource Attribute Query* message to query the default RFC 2833 payload type configured for each module on the IP Network Interface Series 2 card.

Byte 12 in the VoIP Terminal Capabilities TLV indicates if the default value is supported and configurable.

Session Timers

Overview

SIP Session Timers are an extension of SIP RFC 2543 which allows a periodic refreshing of SIP sessions using the RE-INVITE message. The refreshing allows both user agents and proxies to determine if the SIP session is still active.

In addition, the SIP software supports response code 422. *Table B-2, Responses (B-7)* in the SIP Support and Compliance Appendix.

Pertinent Specification

RFC 2543/4028

Configuring

Use the following TLVs in the *VoIP Protocol Configure* (0x00EE) message to implement this feature:

- 0x027C Min-SE Interval
- 0x027B Session Interval

Delayed Media

Overview

The CSP supports delayed media to allow interworking with third party call control applications (3PCC).

Description

- Delayed media is implemented for inbound calls. That is, the CSP can accept a delayed media call but does not generate one.
- The CSP accepts INVITE messages with SDP or without SDP or with held SDP.
- The CSP accepts SDP in the positive ACK message.
- Delayed media is not supported for bearer-free calls. It is supported for bearer calls only.
- Connect instructing coupling and connection mode is not supported. In delayed media, the destination IP address is not known at the time of coupling, therefore the CSP cannot outseize an IPN-2 channel without the destination IP address and at the same time set the connection mode. The call will purge with reason 0x1A in this scenario.

RE-INVITE Message

Overview

The INVITE method is used to establish media sessions between User Agents. The RE-INVITE message permits the CSP to change parameters of an existing or pending call. For example, the RE-INVITE message is used with Session Timers to allow the CSP to regularly "ping" the far end to ensure that they are active.

Important! It is not possible to change the codecs (COder-DECoders) at runtime using the RE-INVITE message.

Pertinent Specification

RFC 3261

Description

The SIP software supports the following:

- RE-INVITE messages that change the port to which media should be sent.
- RE-INVITE messages that change the connection address.
- Media stream on hold (connection address is zero).
- Initial INVITE messages on hold.
- Initial INVITE messages with no Session Description Protocol (SDP).

Codec List in SIP-Initiated Offer

Overview

The CSP can send a list of up to five supported codecs in an offer to an endpoint when establishing a media session. The receiving endpoint selects one codec from that list and reports the selection back in the answer message to the CSP.

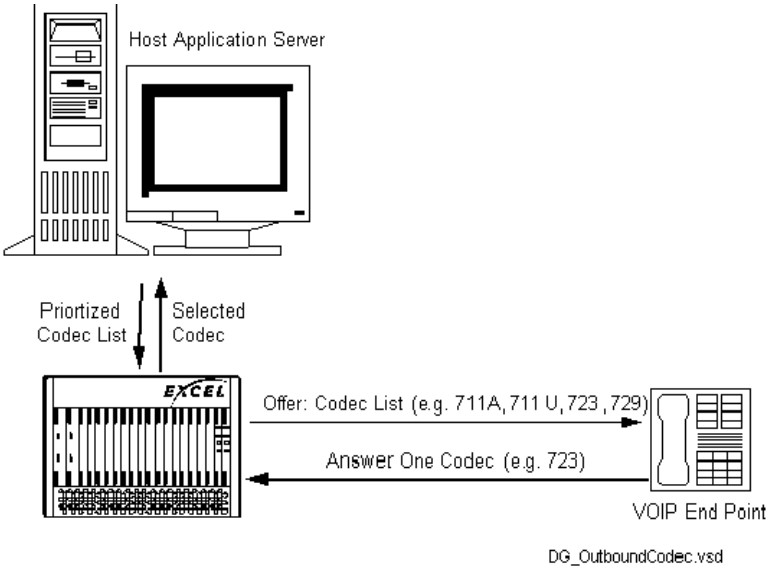
The following applies to this feature:

- The codec list is applicable for calls that use SIP or H.323 signaling and is not for clear-channel VoIP calls. For H.323 signaling, refer to *Codec List in H.323-Initiated Offers (9-9)*.
- The IP Network Interface card must use Profile 2.
- The VDAC-ONE card must use Profile 0.

Without this feature, the call might not get established because the CSP could include a codec type that the endpoint will not accept.

The figure below provides an overview of this process.

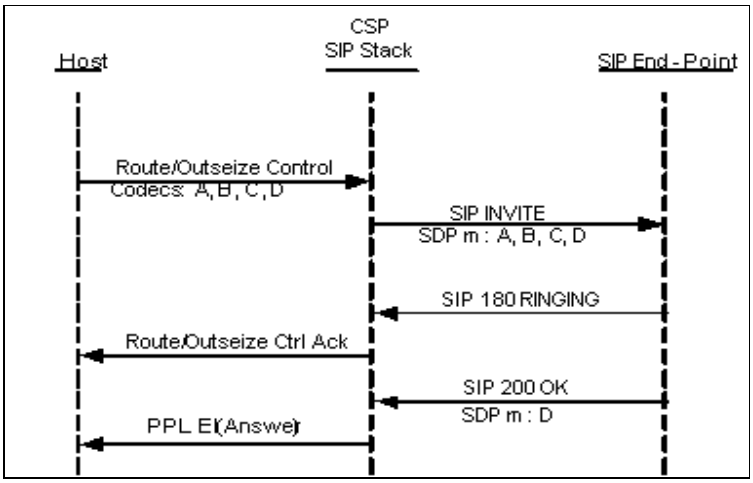
Figure 5-7 Codec List in Offer



Outseize Control message

The host application provides the codec list and sends the list in the *Outseize Control* message as shown in the next figure:

Figure 5-8 Host Provides Codec List



TLVs Used

Use the following TLVs in the *Outseize Control* message to include the codec types:

- 0x29FF - Local End-Point Media Info
- 0x2A0E - Media Connection Address
- 0x2A01 - Per Media Stream Information
- 0x2A03 - Media Type
- 0x2A07 - Media Port
- 0x2A02 - Per Codec Info

The trace below shows how to nest these TLVs in the *Outseize Control* message. Note that the *Outseize Control* message cannot exceed 260 bytes.

Important! DO NOT use the following TLVs for these codec types:

- 0x0100 - RTP Payload Type
- 0x0101 - RTP Payload Size
- 0x27B0 - RTP Payload Type
- 0x27B1 - RTP Payload Size

Outseize Control Message Trace

The following is a message trace for the *Outseize Control* message followed by the corresponding SIP INVITE message.

```
00 6c 00 2C 00 00 $node_id
00 ' address method
01 ' num of address elements
0d ' address type = span/channel
03 ' data length
00 01 01 ' span/channel
01
03 ' icb type = extended
00 33 ' icb subtype
00 5a ' icb data len
00 03 ' num tlvs
29 19 00 05 35 30 38 37 00                ' Called party number
27 7E 00 03 08 01 00                      ' L3 protocol type
29 ff 00 44                               ' Local End-Point Media Info
     2a 0e 00 04 0a 0a 01 25              ' Media connection address
     2a 01 00 38                          ' Per media stream information
           2a 03 00 01 00                 ' Media Type
           2a 07 00 04 00 00 5b 90        ' Media Port
           2a 02 00 13                    ' Per Codec Info
                2a 08 00 02 00 02         ' Codec Payload Type
                2a 09 00 01 02            ' Codec Payload Desc
                2a 0b 00 04 00 00 1f 40   ' Codec clock rate
           2a 02 00 06                    ' Per codec info
                2a 08 00 02 00 01         ' Codec Payload Type
           2a 02 00 06                    ' Per codec info
                2a 08 00 02 00 12         ' Codec Payload Type
```

SIP INVITE

```
INVITE sip:5087@10.10.1.203:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.32
To: 5087<sip:5087@10.10.1.203:5060>
From:
   00000000<sip:00000000@10.10.1.32:5060>;tag=19353246318
   f
Call-ID: EXCEL-CSP255.78f.12687.680@10.10.1.32
Contact: 00000000<sip:00000000@10.10.1.32:5060>
User-Agent: Excel_CSP/83.1.115
Supported: timer
Session-Expires: 40
Min-SE: 39
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 142

v=0
o=sip 946686677 946686677 IN IP4 10.10.1.32
s=SIP_Call
c=IN IP4 10.10.1.37
t=0 0
m=audio 23440 RTP/AVP 0 8 18
a=rtpmap:0 PCMU/8000
```

Send and Receive SIP Signals Using the Same Port

Prior to this feature

Prior to this feature, the CSP received inbound SIP signals on a specific User Datagram Protocol (UDP) port - by default it was port 5060. The receiving port could be changed at run time.

All outbound SIP signals were sent out on a UDP port allocated by the CSP when you configured SIP. This port number could be any valid (and unused) port number above 1023. The port used for outbound SIP signaling from the CSP remained constant until SIP was deconfigured.

With this feature

With this feature, the CSP now sends and receives SIP signals using the same port as configured by the host. By default, the CSP uses port 5060 for all SIP traffic.

Important Points Regarding this Feature

Note the following important points before implementing this feature:

Important! Dialogic recommends that the port not be changed while calls are in progress because doing so could cause the following:

- After a port change, any outstanding SIP responses to the previous port will be ignored.
- If the remote UA has not received a target refresh request from the CSP since the port change, then any request from a remote UA will be ignored subsequent to the port change. These requests include, but are not limited to, BYE, RE-INVITE, OPTIONS, REFER and NOTIFY.

Other Points

- As soon as the host changes the port number, the CSP expects inbound SIP traffic in the new port. All outbound SIP traffic hereafter contains the new port number in the Contact Header.
- When the port number is changed, the previous port is closed. Any SIP messages sent to the previous port number will not be received by the CSP.
- Via and Contact headers are updated for outbound calls.

Changing Port Number

You can change the port number for inbound/the EXS API or SwitchKit CSA. The following describes each procedure.

Using API

Use the following TLV in the *VoIP Protocol Configure* (0x00EE) message to change the port number for inbound/outbound SIP signaling traffic.

The range of values is 0x400-0xFFFF (1024-65535) provided the port is not in use. The default is port 0x13C4 (5060).

0x0262 Use SIP Local Port

Byte	Description	
0, 1	Tag	0x0262
2, 3	Length	0x0002
4, 5	Value[0]	Port Number

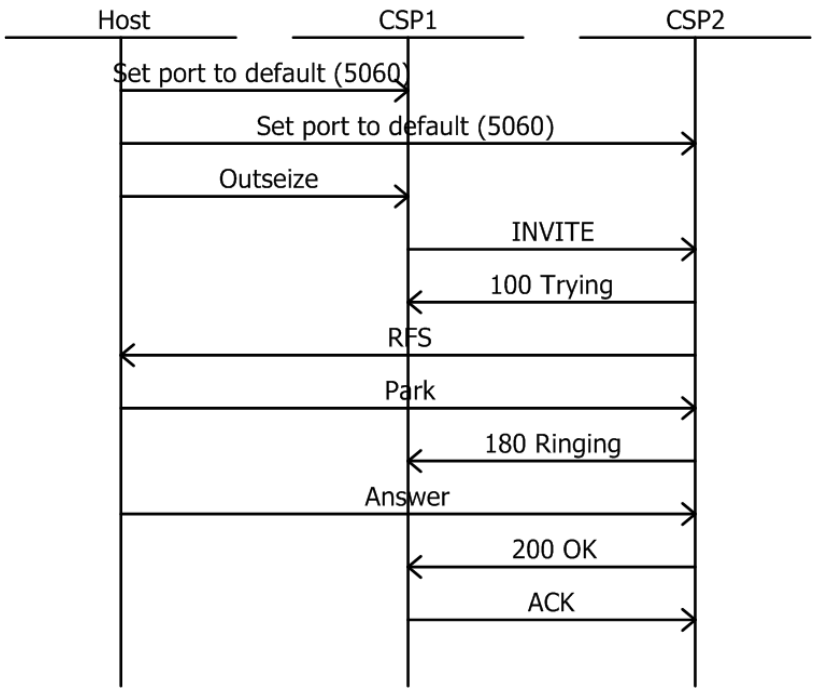
Using CSA

Refer to *Configuring SIP* in the *Converged Services Administrator User's Guide* to configure with SwitchKit CSA.

Call Flows and Traces

Default

The port is set to the default value in the call flow and corresponding trace below.




```
CSP1:

H->X
    [00 15 00 ee 00 00 01 00 00 00 00 02 01 c8 00 01 04 02
     62 00 02 13 c4]

X->H
    [00 07 00 ee 00 00 01 00 10]

H->X
    [00 83 00 2c 00 01 01 00 01 0d 03 00 64 1f 02 03 00 1e
     00 0f 00 02 01 16 00 02 00 00 01 1a 00 03 00 00 00 03
     00 33 00 5d 00 09 27 7e 00 03 08 00 00 29 19 00 06 32
     32 32 32 32 00 29 1b 00 0c 31 30 2e 31 30 2e 31 2e 32
     35 32 00 29 1c 00 04 00 00 13 c4 29 23 00 06 38 38 38
     38 38 00 29 25 00 0c 31 30 2e 31 30 2e 31 2e 32 35 34
     00 29 26 00 04 00 00 13 c4 27 92 00 04 0a 0a 01 b6 27
     93 00 04 00 00 10 7c]

X->H
    [00 07 00 2c 00 01 01 00 10]

X->H
```

```
    [00 11 00 43 00 15 01 00 01 0d 03 00 64 1f 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 15 01]

X->H
    [00 44 00 43 00 16 01 00 01 0d 03 00 64 1f 00 a7 00 20
   01 03
     00 33 00 2e 00 01 29 ff 00 28 2a 0e 00 04 0a 0a 01 ac
     2a 01 00 1c 2a 03 00 01 00 2a 07 00 04 00 00 27 9c 2a
     13 00 01 00 2a 02 00 06 2a 08 00 02 00 02]

H->X
    [00 05 00 43 00 16 01]
2
 Printing all SIP messages

1 -SENT To 10.10.1.252:5060 at 3857
INVITE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254
To: 22222<sip:22222@10.10.1.252:5060>
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
Contact: 88888<sip:88888@10.10.1.254:5060>
User-Agent: Excel_CSP/83.1.13
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.254
s=SIP_Call
c=IN IP4 10.10.1.182
t=0 0
m=audio 4220 RTP/AVP 0

2 -RECEIVED From 10.10.1.252:5060 at 3857
SIP/2.0 100 Trying
To: 22222<sip:22222@10.10.1.252:5060>;tag=5790bba
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.1.13
Content-Length: 0
```

```
3 -RECEIVED From 10.10.1.252:5060 at 3857
SIP/2.0 180 Ringing
To: 22222<sip:22222@10.10.1.252:5060>;tag=5790bba
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.1.13
Content-Length: 0


4 -RECEIVED From 10.10.1.252:5060 at 3857
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=5790bba
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Require: timer
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.1.13
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1139999646 1139999646 IN IP4 10.10.1.252
s=SIP_Call
c=IN IP4 10.10.1.172
t=0 0
m=audio 10140 RTP/AVP 0
a=sendrecv

5 -SENT To 10.10.1.252:5060 at 3857
ACK sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254
To: 22222<sip:22222@10.10.1.252:5060>;tag=5790bba
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
CSeq: 1 ACK
Content-Length: 0


**********************************************************

CSP2:
```

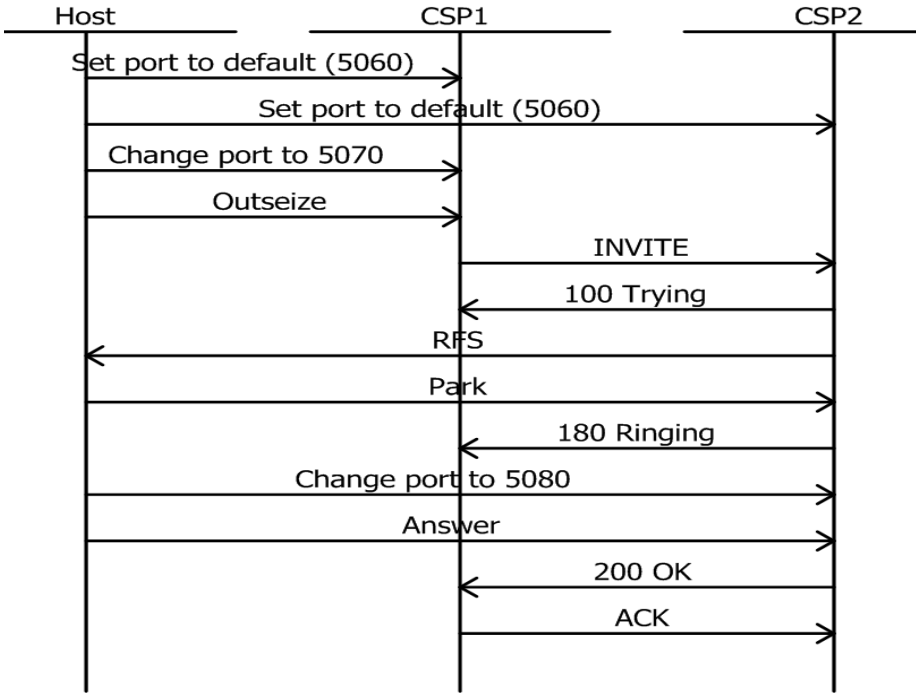
```
H->X
    [00 15 00 ee 00 00 02 00 00 00 00 02 01 c8 00 01 04 02
     62 00 02 13 c4]

X->H
    [00 07 00 ee 00 00 02 00 10]

X->H
    [00 da 00 2d 00 07 02 00 01 0d 03 00 c8 07 00 33 01 03
   00 33
     00 c6 00 11 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 06 32 32 32 32 32 00 29 1b 00 0c 31 30 2e 31 30
     2e 31 2e 32 35 32 00 29 1c 00 04 00 00 13 c4 29 23 00
     06 38 38 38 38 38 00 29 25 00 0c 31 30 2e 31 30 2e 31
     2e 32 35 34 00 29 26 00 04 00 00 13 c4 29 28 00 06 38
     38 38 38 38 00 29 2d 00 06 38 38 38 38 38 00 29 2f 00
     0c 31 30 2e 31 30 2e 31 2e 32 35 34 00 29 30 00 04 00
     00 13 c4 29 33 00 01 01 27 18 00 08 02 00 00 00 05 88
     88 80 27 17 00 06 02 00 05 22 22 20 29 ff 00 23 2a 0e
     00 04 0a 0a 01 b6 2a 01 00 17 2a 03 00 01 00 2a 07 00
     04 00 00 10 7c 2a 02 00 06 2a 08 00 02 00 02 29 16 00
     01 01]

H->X
    [00 0c 00 2d 00 07 02 00 01 0d 03 00 c8 07]

H->X
    [00 0c 00 2d 00 07 02 00 01 0d 03 00 c8 07]

H->X
    [00 11 00 bf 00 00 02 00 02 0d 03 00 c8 07 0d 03 00 c8
     07]

X->H
    [00 07 00 bf 00 00 02 00 10]

H->X
    [00 0d 00 ba 00 00 02 00 01 0d 03 00 c8 07 01]

X->H
    [00 07 00 ba 00 00 02 00 10]
2
 Printing all SIP messages

1 -RECEIVED From 10.10.1.254:5060 at 3002
INVITE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254
To: 22222<sip:22222@10.10.1.252:5060>
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
```

```
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
Contact: 88888<sip:88888@10.10.1.254:5060>
User-Agent: Excel_CSP/83.1.13
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.254
s=SIP_Call
c=IN IP4 10.10.1.182
t=0 0
m=audio 4220 RTP/AVP 0

2 -SENT To 10.10.1.254:5060 at 3002
SIP/2.0 100 Trying
To: 22222<sip:22222@10.10.1.252:5060>;tag=5790bba
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.1.13
Content-Length: 0


3 -SENT To 10.10.1.254:5060 at 3002
SIP/2.0 180 Ringing
To: 22222<sip:22222@10.10.1.252:5060>;tag=5790bba
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.1.13
Content-Length: 0


4 -SENT To 10.10.1.254:5060 at 3002
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=5790bba
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Require: timer
Supported: timer
```

```
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.1.13
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1139999646 1139999646 IN IP4 10.10.1.252
s=SIP_Call
c=IN IP4 10.10.1.172
t=0 0
m=audio 10140 RTP/AVP 0
a=sendrecv

5 -RECEIVED From 10.10.1.254:5060 at 3002
ACK sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254
To: 22222<sip:22222@10.10.1.252:5060>;tag=5790bba
From: 88888<sip:88888@10.10.1.254:5060>;tag=41275790f11
Call-ID: EXCEL-CSP1.101f.3857.140@10.10.1.254
CSeq: 1 ACK
Content-Length: 0
```

Change port

The port is changed to 5070 in the call flow and trace below.




```
CSP1:

H->X
    [00 15 00 ee 00 00 01 00 00 00 00 02 01 c8 00 01 04 02
     62 00 02 13 c4]

X->H
    [00 07 00 ee 00 00 01 00 10]

H->X
    [00 15 00 ee 00 00 01 00 00 00 00 02 01 c8 00 01 04 02
     62 00 02 13 ce]

X->H
    [00 07 00 ee 00 00 01 00 10]

H->X
    [00 83 00 2c 00 01 01 00 01 0d 03 00 64 1f 02 03 00 1e
     00 0f 00 02 01 16 00 02 00 00 01 1a 00 03 00 00 00 03
     00 33 00 5d 00 09 27 7e 00 03 08 00 00 29 19 00 06 32
     32 32 32 32 00 29 1b 00 0c 31 30 2e 31 30 2e 31 2e 32
     35 32 00 29 1c 00 04 00 00 13 c4 29 23 00 06 38 38 38
     38 38 00 29 25 00 0c 31 30 2e 31 30 2e 31 2e 32 35 34
     00 29 26 00 04 00 00 13 c4 27 92 00 04 0a 0a 01 b6 27
     93 00 04 00 00 10 7c]

X->H
```

```
    [00 07 00 2c 00 01 01 00 10]

X->H
    [00 11 00 43 00 13 01 00 01 0d 03 00 64 1f 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 13 01]

H->X
    [00 05 00 43 00 13 01]

X->H
    [00 44 00 43 00 14 01 00 01 0d 03 00 64 1f 00 a7 00 20
   01 03
     00 33 00 2e 00 01 29 ff 00 28 2a 0e 00 04 0a 0a 01 ab
     2a 01 00 1c 2a 03 00 01 00 2a 07 00 04 00 00 3d 9c 2a
     13 00 01 00 2a 02 00 06 2a 08 00 02 00 02]

H->X
    [00 05 00 43 00 14 01]
2
 Printing all SIP messages

1 -SENT To 10.10.1.252:5060 at 3463
INVITE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254:5070
To: 22222<sip:22222@10.10.1.252:5060>
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
Contact: 88888<sip:88888@10.10.1.254:5070>
User-Agent: Excel_CSP/83.1.13
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.254
s=SIP_Call
c=IN IP4 10.10.1.182
t=0 0
m=audio 4220 RTP/AVP 0

2 -RECEIVED From 10.10.1.252:5060 at 3463
SIP/2.0 100 Trying
To: 22222<sip:22222@10.10.1.252:5060>;tag=1073a30
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
```

```
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254:5070
User-Agent: Excel_CSP/83.1.13
Content-Length: 0


3 -RECEIVED From 10.10.1.252:5060 at 3464
SIP/2.0 180 Ringing
To: 22222<sip:22222@10.10.1.252:5060>;tag=1073a30
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254:5070
User-Agent: Excel_CSP/83.1.13
Content-Length: 0


4 -RECEIVED From 10.10.1.252:5080 at 3466
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=1073a30
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5080>
Require: timer
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.254:5070
User-Agent: Excel_CSP/83.1.13
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1139999254 1139999254 IN IP4 10.10.1.252
s=SIP_Call
c=IN IP4 10.10.1.171
t=0 0
m=audio 15772 RTP/AVP 0
a=sendrecv

5 -SENT To 10.10.1.252:5080 at 3466
ACK sip:22222@10.10.1.252:5080 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254:5070
To: 22222<sip:22222@10.10.1.252:5060>;tag=1073a30
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
CSeq: 1 ACK
```

```
Content-Length: 0



**********************************************************

CSP2:

H->X
    [00 15 00 ee 00 00 02 00 00 00 00 02 01 c8 00 01 04 02
     62 00 02 13 c4]

X->H
    [00 07 00 ee 00 00 02 00 10]

X->H
    [00 da 00 2d 00 06 02 00 01 0d 03 00 c8 06 00 33 01 03
   00 33
     00 c6 00 11 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 06 32 32 32 32 32 00 29 1b 00 0c 31 30 2e 31 30
     2e 31 2e 32 35 32 00 29 1c 00 04 00 00 13 c4 29 23 00
     06 38 38 38 38 38 00 29 25 00 0c 31 30 2e 31 30 2e 31
     2e 32 35 34 00 29 26 00 04 00 00 13 c4 29 28 00 06 38
     38 38 38 38 00 29 2d 00 06 38 38 38 38 38 00 29 2f 00
     0c 31 30 2e 31 30 2e 31 2e 32 35 34 00 29 30 00 04 00
     00 13 ce 29 33 00 01 01 27 18 00 08 02 00 00 00 05 88
     88 80 27 17 00 06 02 00 05 22 22 20 29 ff 00 23 2a 0e
     00 04 0a 0a 01 b6 2a 01 00 17 2a 03 00 01 00 2a 07 00
     04 00 00 10 7c 2a 02 00 06 2a 08 00 02 00 02 29 16 00
     01 01]

H->X
    [00 0c 00 2d 00 06 02 00 01 0d 03 00 c8 06]

H->X
    [00 0c 00 2d 00 06 02 00 01 0d 03 00 c8 06]

H->X
    [00 11 00 bf 00 00 02 00 02 0d 03 00 c8 06 0d 03 00 c8
     06]

X->H
    [00 07 00 bf 00 00 02 00 10]

H->X
    [00 15 00 ee 00 00 02 00 00 00 00 02 01 c8 00 01 04 02
     62 00 02 13 d8]

X->H
    [00 07 00 ee 00 00 02 00 10]
```

```
H->X
    [00 0d 00 ba 00 00 02 00 01 0d 03 00 c8 06 01]

X->H
    [00 07 00 ba 00 00 02 00 10]
2
 Printing all SIP messages

1 -RECEIVED From 10.10.1.254:5070 at 2608
INVITE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254:5070
To: 22222<sip:22222@10.10.1.252:5060>
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
Contact: 88888<sip:88888@10.10.1.254:5070>
User-Agent: Excel_CSP/83.1.13
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.254
s=SIP_Call
c=IN IP4 10.10.1.182
t=0 0
m=audio 4220 RTP/AVP 0

2 -SENT To 10.10.1.254:5070 at 2608
SIP/2.0 100 Trying
To: 22222<sip:22222@10.10.1.252:5060>;tag=1073a30
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254:5070
User-Agent: Excel_CSP/83.1.13
Content-Length: 0


3 -SENT To 10.10.1.254:5070 at 2609
SIP/2.0 180 Ringing
To: 22222<sip:22222@10.10.1.252:5060>;tag=1073a30
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
```

```
Via: SIP/2.0/UDP 10.10.1.254:5070
User-Agent: Excel_CSP/83.1.13
Content-Length: 0


4 -SENT To 10.10.1.254:5070 at 2611
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=1073a30
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5080>
Require: timer
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.254:5070
User-Agent: Excel_CSP/83.1.13
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1139999254 1139999254 IN IP4 10.10.1.252
s=SIP_Call
c=IN IP4 10.10.1.171
t=0 0
m=audio 15772 RTP/AVP 0
a=sendrecv

5 -RECEIVED From 10.10.1.254:5070 at 2611
ACK sip:22222@10.10.1.252:5080 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254:5070
To: 22222<sip:22222@10.10.1.252:5060>;tag=1073a30
From: 88888<sip:88888@10.10.1.254:5060>;tag=41271073d87
Call-ID: EXCEL-CSP1.101f.3463.950@10.10.1.254
CSeq: 1 ACK
Content-Length: 0
```

SIP Notifications of Options

Overview

The SIP stack can now report the receipt of SIP OPTIONS to the host.

Specification Built From

RFC 3261

Description

The SIP method, OPTIONS, allows a User Agent (UA) to query another UA or a proxy regarding its capabilities. This feature allows a client to discover information about the supported methods, extension content types with or without a dialog being established.

When the CSP receives an OPTIONS message, it is reported to the host as a PPL Event Indication including the Request URI in the message.

API Call Control Messages

PPL Event Indication (0x0043) message:

The PPL Event Indication supports the following TLVs:

- *0x2946 - SIP Request URI Password (New)*
- *0x2947 - SIP Request URI Header (New)*
- *0x2954 - SIP Request URI User Name*
- *0x2955 - SIP Request URI Host Name*
- *0x2956 - SIP Request URI Port*
- *0x2958 - SIP Request URI Parameters*

PPL Information

PPL Event (0x002F) in SIP Component 0x00A7 reports the received OPTIONS message.

API and CSA Configuring and Querying

This feature can be configured either using the API or CSA.

API Configuring and Querying

To enable this feature, set bit 8 of the PPL Event Notification Mask TLV (0x0282) in the *VoIP Protocol Configure* (0x00EE) message.

CSA Configuring and Querying

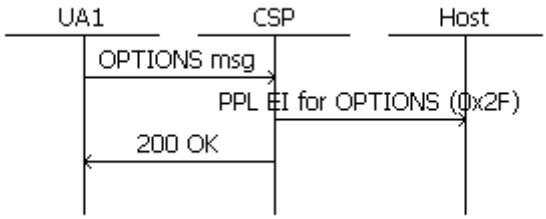
To configure and query the SIP stack for this feature, view the Configure SIP Advanced screen, Call Progress Notification, and select **Options message recvd**.

0x0282 PPL Event Notification Mask

Used in:
VoIP Protocol Configure message
VoIP Protocol Query message

Byte	Description	
0, 1	Tag	0x0282
2, 3	Length	0x0004
4-7	Value[0-3]	32 bit mask with each bit selects specific PPL Event Indication. Bit 8 - Options message received.

Call Flow



TLVs

0x2946 - SIP Request URI Password

Used in:
PPL Event Request message
PPL Event Indication message

Byte	Description	
0, 1	Tag	0x2946
2, 3	Length	Variable (maximum 20 bytes)
4-n	Value[0-n]	Null-terminated ASCII string

0x2947 - SIP Request URI Header

Used in:
PPL Event Request message
PPL Event Indication message

Byte	Description	
0, 1	Tag	0x2947
2, 3	Length	Variable (maximum of 650 bytes)
4-n	Value[0-n]	Null-terminated ASCII string

0x2954 - SIP Request URI User Name

Used in:
0x0033 NPDI Universal ICB in:
Route Control message
Request for Service with Data message

Byte	Description	
0, 1	Tag	0x2954
2, 3	Length	Variable (maximum of 40 bytes)
4-n	Value[0-n]	Null Terminated ASCII for user name in Request URI.

0x2955 - SIP Request URI Host Name

Used in:
0x0033 NPDI Universal ICB in:
Route Control message
Request for Service with Data message

Byte	Description	
0, 1	Tag	0x2955
2, 3	Length	Variable (maximum of 80 bytes)
4-n	Value[0-n]	Null Terminated ASCII for host name in Request URI.

0x2956 - SIP Request URI Port

Used in:
0x0033 NPDI Universal ICB in:
Route Control message

Byte	Description	
0, 1	Tag	0x2956
2, 3	Length	0x0004
4	Data[0] Data[1] Data[2] Data[3]	MSB Port in Request URI : : LSB Port in Request URI

0x2958 - SIP Request URI Parameters

Used in:
0x0033 NPDI Universal ICB in:
Request for Service with Data message
Route Control message
Outseize Control message

Byte	Description	
0, 1	Tag	0x2958
2, 3	Length	Variable (maximum of 650 bytes)
4-n	Value[0-n]	Null-terminated ASCII string

Important! All TLVs except the port number TLV ends with a NULL terminator, therefore the actual data supported is one less than the maximum length supported for the TLVs. Also note that the CSP nacks all SIP requests of a size greater that 1500 bytes with the 513 message, Too Large

Disabling the SIP Domain Name System (DNS) Server

Overview

Prior to this feature, you enabled and disabled the Domain Name System (DNS) Server at the card level. The DNS task and a UDP socket, used to contact the DNS Server, is created when the CSP Matrix Series 3 Card boots up. Disabling the DNS Server disables the DNS functionality in the CSP Matrix Series 3 Card.

Important! By default, the DNS Server is not configured (disabled).

Once the DNS Server is configured in the CSP SIP stack, there was no provision to disable the DNS Server lookup.

This feature allows the host application to disable the DNS Server lookup at any time. Disabling the DNS Server stops the CSP from sending outbound SIP calls requesting a DNS Server lookup.

API Messages

This feature requires the following API messages, TLVs and a new Response Status Value (NACK):

API Messages (Unmodified)

- *VoIP Protocol Configure* (0x00EE)
- *VoIP Protocol Query* (0x00EF)

TLVs (Unmodified)

- DNS IP Address (0x01D6)
- Default Domain Name (0x01D7)

Response Status Value NACK (New)

- DNS Server Disabled (0x1312)

Hex ID	Name	Description as necessary
13 12	DNS Server Disabled (NACK)	All Route/Outseize Control API messages are nacked with this response if the outbound SIP call requires a DNS Server lookup and the DNS is disabled.

Configuring with API

Disable the DNS Server

To disable the DNS Server, use the DNS IP Address TLV (0x01D6) to set the Primary, Secondary and Tertiary DNS IP addresses to zero. The host then sends the DNS IP Address TLV in the *VoIP Protocol Configure* (0x00EE) message to the CSP SIP stack to disable the DNS Server.

By disabling the DNS Server, all outbound SIP calls that require DNS lookup fail and be indicated by a NACK value of 0x1312. This NACK value applies to the *Outseize Control* (0x002C) and *Route Control* (0x00E8) messages.

When the host instructs the CSP Matrix Series 3 Card to disable the DNS Server, the following occurs:

- The Primary, Secondary and Tertiary DNS Server IP addresses are cleared (set to 0).
- The DNS Server is cleared.
- All DNS Server lookup requests queued up from SIP are NACKed with the failure status 0x01312.
- The most recent look-up cache will be cleared.

Enable the DNS Server

When host application configures (enables) the DNS Server, the destination address for the UDP socket must be configured. You must have the host application do the following:

1. Have the DNS IP Address TLV (0x01D6) set the Primary, Secondary and Tertiary DNS Server IP addresses within the *VoIP Protocol Configure* (0x00EE) message.
    - Initially, the destination address for the UDP socket is set to the primary DNS IP address and the DNS task state is changed to DNS_SOCK_STATE_PRIM.
    - If the primary DNS IP address cannot be reached and if the secondary DNS IP address is configured, then the secondary DNS is contacted. On changing the contact to the secondary DNS, the destination address for the UDP socket is changed accordingly and the DNS task state is changed to DNS_SOCK_STATE_SEC.

- If the secondary primary DNS cannot be reached and if the tertiary DNS address is configured, then the tertiary DNS is contacted. On changing to tertiary DNS, the destination address for the UDP socket is changed accordingly and the DNS task state is changed to DNS_SOCK_STATE_TER.
- The DNS request for look-up fails after three attempts.

2. Have the Default Domain Name TLV (0x01D7) set the domain name within the *VoIP Protocol Configure* (0x00EE) message. This configuration can be queried using the *VoIP Protocol Query* (0x00EF) message.

Query the DNS Server Configuration

The DNS Server configuration for SIP can be queried using the *VoIP Protocol Query* (0x00EF) message.

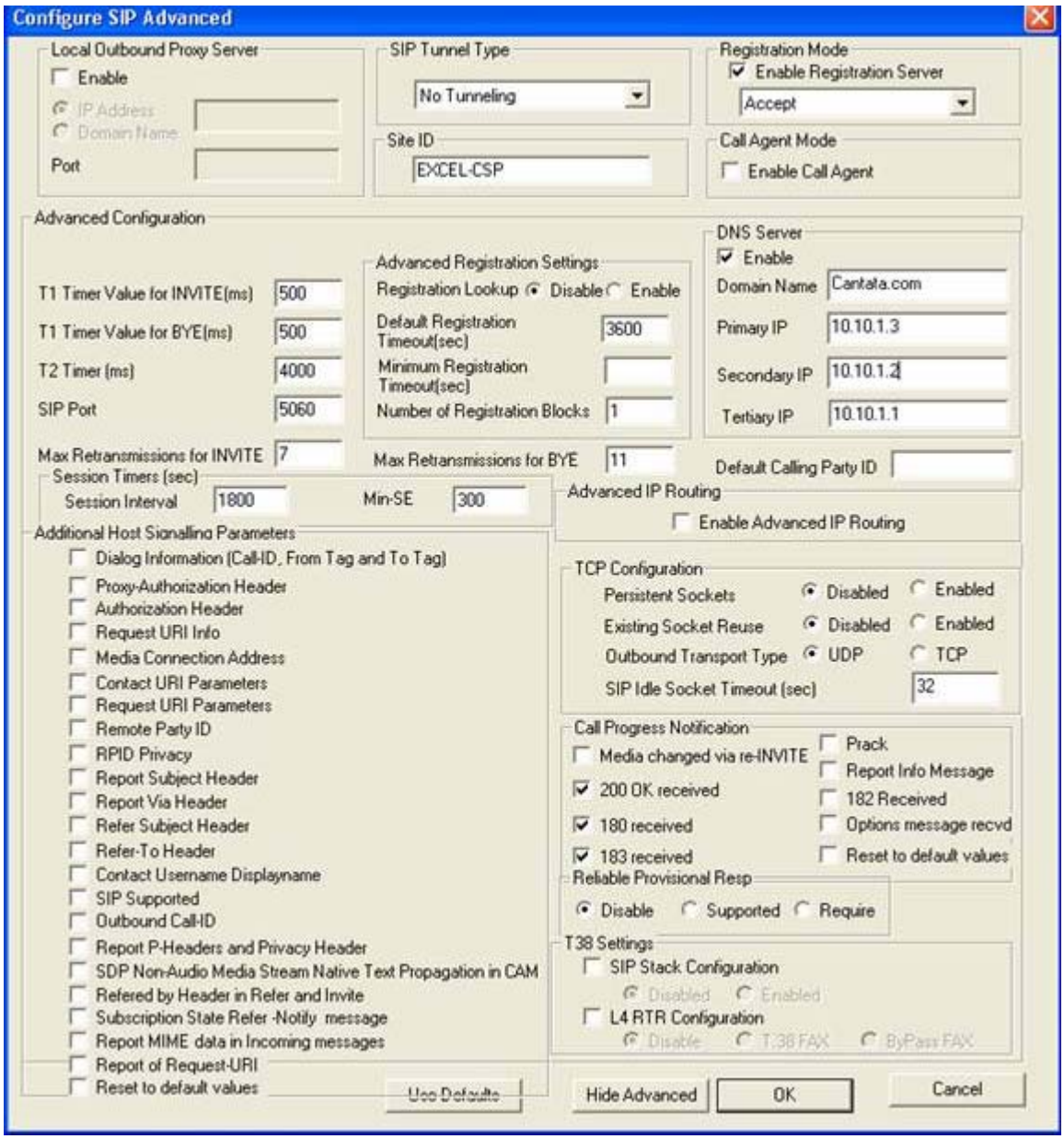
- If the DNS Server is disabled, the DNS IP Address TLV (0x01D6) and the Default Domain Name TLV (0x01D7) will be absent in the query response.
- If the DNS Server is enabled, the DNS IP Address TLV (0x01D6) is present in the query response and the Default Domain Name TLV (0x01D7) will be present if it was configured.
- If the Default Domain Name TLV (0x01D7) is returned in the query response it means the domain name was configured. If the TLV is absent in the query response it means that the domain name was not configured or the DNS is disabled.

Configuring with CSA

Enabling the DNS Server

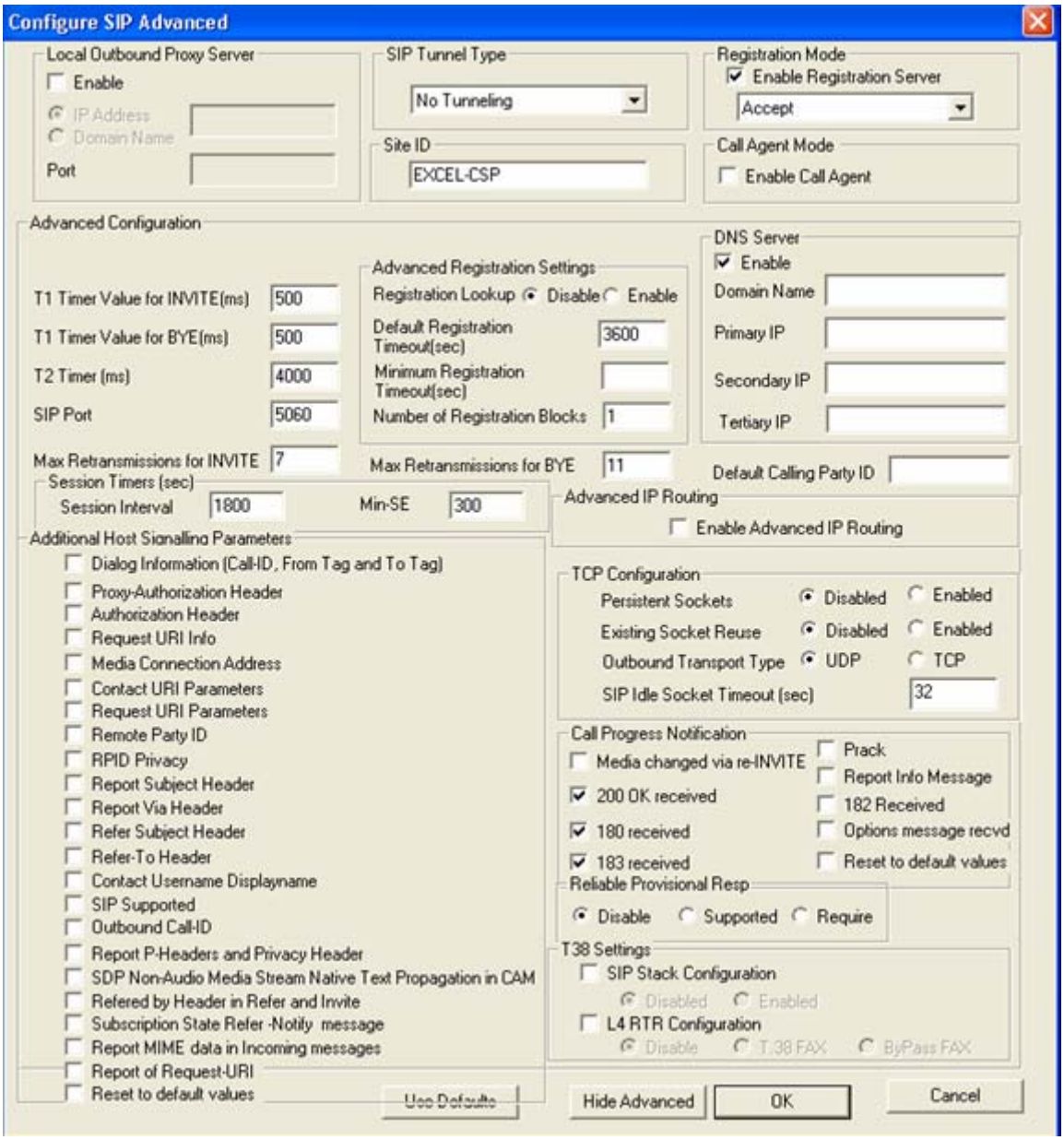
Enable the DNS by entering the Domain name, Primary IP address, Secondary IP address and Tertiary IP address.

The host application uses the DNS IP Address TLV (0x01D6) to set the DNS IP addresses, and the Default Domain Name TLV (0x01D7) to set the domain name within the *VoIP Protocol Configure* (0x00EE).


Configure SIP Advanced
Local Outbound Proxy Server
Enable
IP Address
Domain Name
Port
SIP Tunnel Type
No Tunneling
Site ID
EXCEL-CSP
Registration Mode
Enable Registration Server
Accept
Call Agent Mode
Enable Call Agent
Advanced Configuration
T1 Timer Value for INVITE(ms) 500
T1 Timer Value for BYE(ms) 500
T2 Timer (ms) 4000
SIP Port 5060
Max Retransmissions for INVITE 7
Advanced Registration Settings
Registration Lookup Disable Enable
Default Registration Timeout(sec) 3600
Minimum Registration Timeout(sec)
Number of Registration Blocks 1
Max Retransmissions for BYE 11
DNS Server
Enable
Domain Name Cantata.com
Primary IP 10.10.1.3
Secondary IP 10.10.1.2
Tertiary IP 10.10.1.1
Default Calling Party ID
Session Timers (sec)
Session Interval 1800
Min-SE 300
Advanced IP Routing
Enable Advanced IP Routing
Additional Host Signalling Parameters
Dialog Information (Call-ID, From Tag and To Tag)
Proxy-Authorization Header
Authorization Header
Request URI Info
Media Connection Address
Contact URI Parameters
Request URI Parameters
Remote Party ID
RPID Privacy
Report Subject Header
Report Via Header
Refer Subject Header
Refer-To Header
Contact Username Displayname
SIP Supported
Outbound Call-ID
Report P-Headers and Privacy Header
SDP Non-Audio Media Stream Native Text Propagation in CAM
Refered by Header in Refer and Invite
Subscription State Refer -Notify message
Report MIME data in Incoming messages
Report of Request-URI
Reset to default values
Use Defaults
TCP Configuration
Persistent Sockets Disabled Enabled
Existing Socket Reuse Disabled Enabled
Outbound Transport Type UDP TCP
SIP Idle Socket Timeout (sec) 32
Call Progress Notification
Media changed via re-INVITE
200 OK received
180 received
183 received
Prack
Report Info Message
182 Received
Options message recvd
Reset to default values
Reliable Provisional Resp
Disable Supported Require
T38 Settings
SIP Stack Configuration
Disabled Enabled
L4 RTR Configuration
Disable T.38 FAX ByPass FAX
Hide Advanced
OK
Cancel

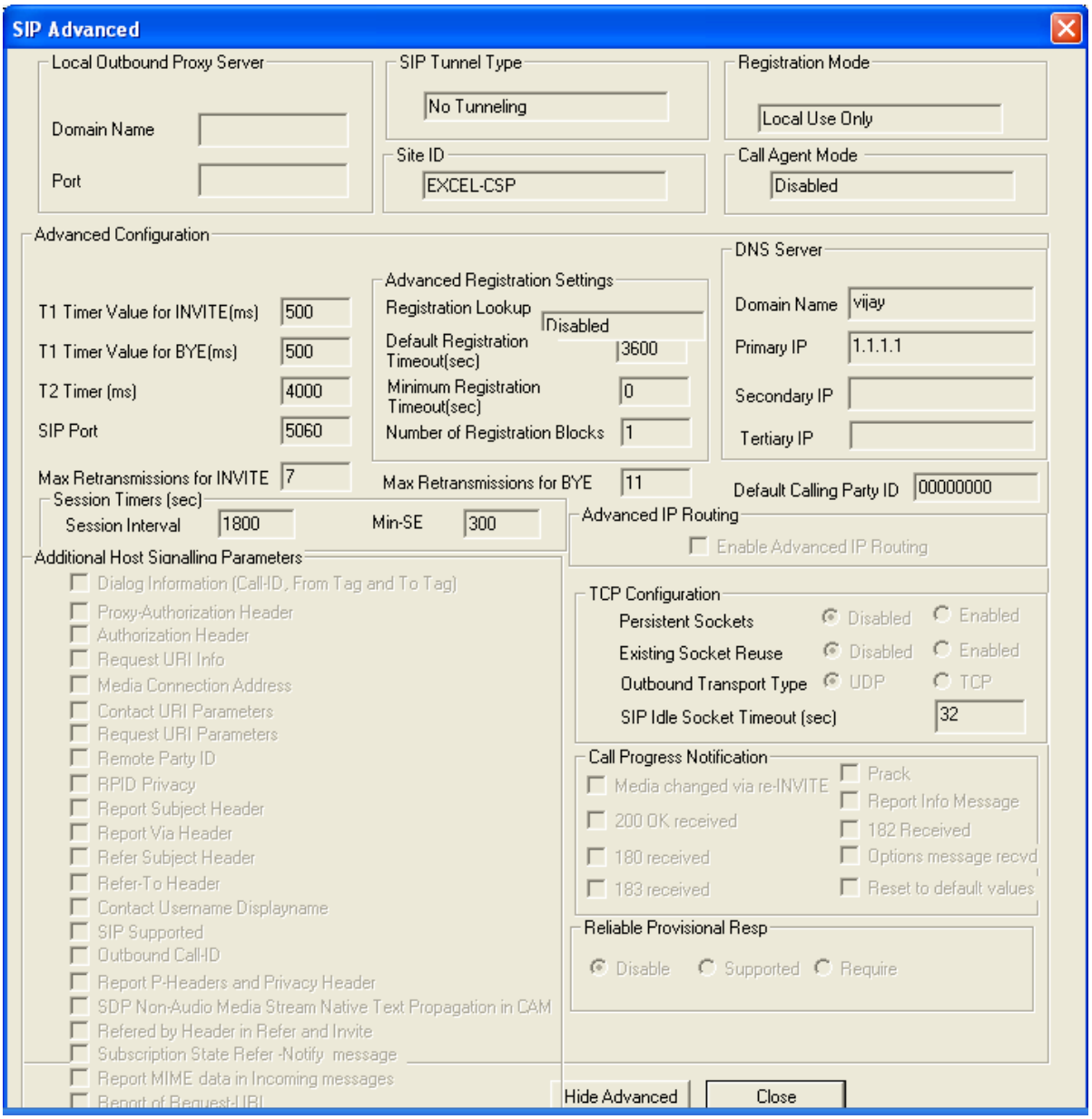
Disabling the DNS Server

To disable the DNS check the **Enable** box and leave the Domain name, Primary IP address, Secondary IP address and Tertiary IP boxes empty.



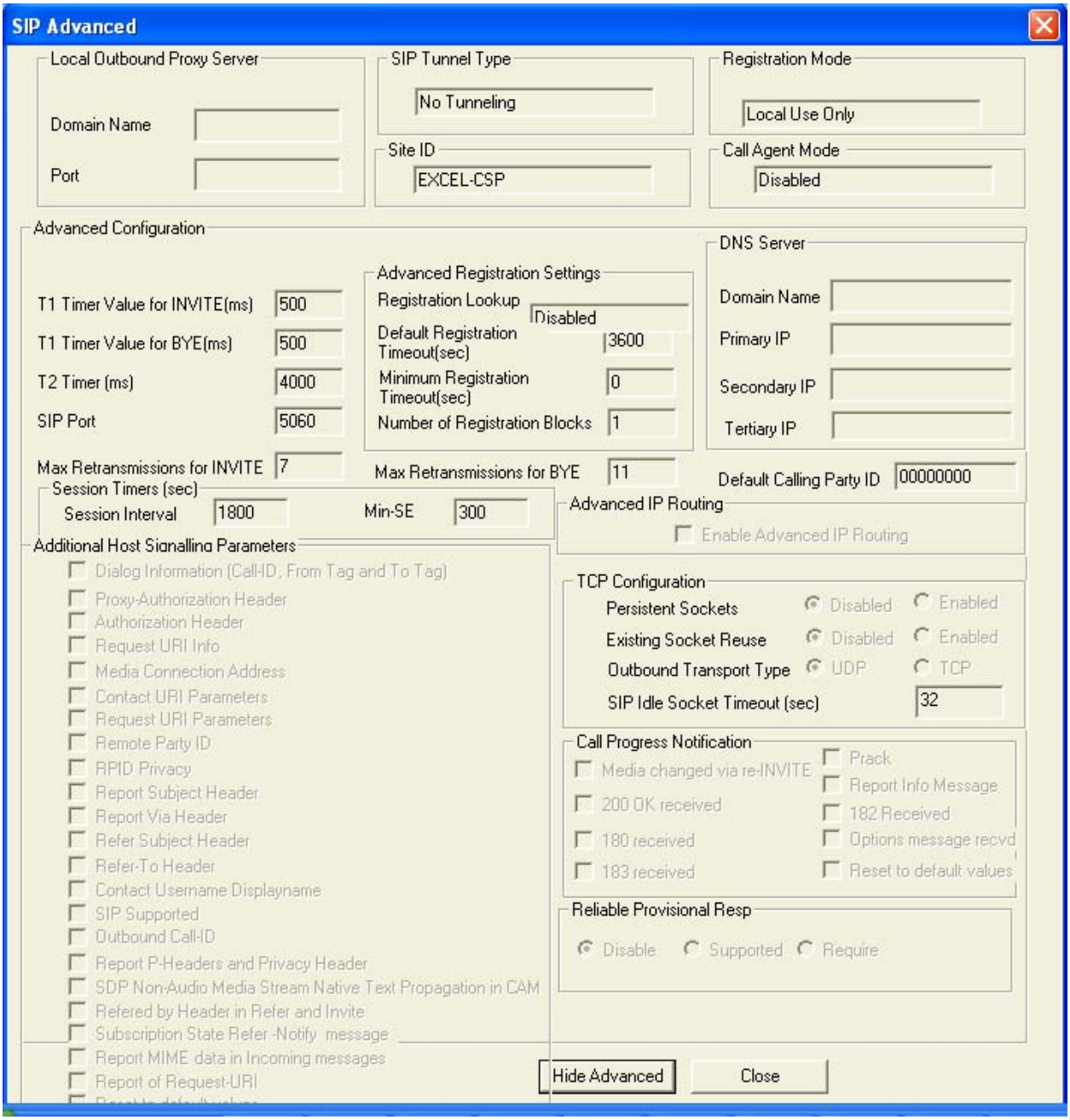
DNS Server with Monitor Mode Enabled

The following shows the DNS Server box when enabled in the Monitor Mode.



DNS Server with Monitor Mode Disabled

The following shows the DNS Server box when disabled in the Monitor Mode.



Advanced SIP Functionality

This section describes advanced SIP functionality including the following:

Call Transfer Functionality

- *REFER and NOTIFY Methods (5-203)*
- *SIP Referred By Mechanism (5-204)*
- *Host Generated Refer Message (5-225)*
- *Refer-To Header Parameter Access (5-231)*
- *SIP Subject Header in REFER (5-234)*
- *Report REFER Request URI in PPL Event Indication (5-236)*
- *Refer to Phone Number (5-246)*

- *SIP Population of Status in Outbound NOTIFY Message (5-248)*
- *SUBSCRIBE and NOTIFY Method for DTMF Detection (5-250)*
- *SIP Notify Subscription State (5-252)*
- *Programmable SIP URI Extensions (5-264)*
- *Support for Request URI Parameters in SIP INVITE Messages (5-269)*
- *Support SIP Max Forward in INVITE Message (5-272)*
- *PRACK Support (5-275)*
- *Support for SIP INFO Message (5-283)*
- *SIP Tunneling (5-286)*
- *SIP Support for MIME (5-300)*

Call Agent Mode

- *Outbound SIP Call with Call Agent Mode (5-308)*

REFER and NOTIFY Methods

REFER Method

Third-party call control enables a SIP entity to be in control of session signaling while the media is exchanged between other entities. In some situations, the controller will not want to continue monitoring (controlling) the signaling of the session. Instead, the controller will want the other entities to continue the session independently. At this point, the controller requires a mechanism to transfer SIP sessions to another entity.

To provide session transfer functionality, the REFER method was defined. One SIP entity instructs another to perform a certain action. For example, the REFER method instructs a server to send a specific request a certain URL.

Refer to PPL Event Indication IDs 0x0021 and 0x0022 in *Table 5-3, PPL Event Indications (5-18)*.

Pertinent Specification

RFC 3515

NOTIFY Method

The NOTIFY method is defined in SIP to provide asynchronous event notification. The CSP supports the NOTIFY method, which is typically used with the REFER method in consultative and non-consultative Call Transfer applications (IP Centrex or IP PBX).

> **Important!** The REFER and NOTIFY methods are supported for bearer-free and bearer calls.

Refer to the PPL Event Request ID 0x0020 in the *Table 5-2, PPL Event Requests (5-14)*.

This feature was implemented per the following IETF documents:

- ·draft-ietf-sip-cc-transfer-05.txt
- ·draft-ietf-sip-refer-02.txt
- ·draft-ietf-sip-replaces-00.txt

Pertinent Specification

RFC 3265

SIP Referred By Mechanism

Overview

This feature supports the insecure refer technique using the Referred-By mechanism. This mechanism supports Call Agent Mode (CAM) and non-CAM configurations. The Referred-By header is disabled by default.

There are applications of the REFER where it is desirable to provide the refer target with the information about the referrer. The refer target can use this information when deciding whether to admit the referenced request. This feature provides the refer target with the SIP URI of the referrer.

Pertinent Specification

RFC 3892

Description

This feature allows the host to instruct the CSP SIP stack to insert the Referred-By header in the outbound REFER request. By default the CSP SIP stack does not include the Referred-By header. The Referred-By header is then reported to the host by the CSP receiving the REFER request (Referee) within the PPL Event Indication to report receipt of REFER message.

The host can use this Referred-By header data in either the *Outseize Control* (0x002C) or *Route Control* (0x00E8) message to instruct the CSP SIP stack to include Referred-By header in the INVITE request to the Refer Target (another CSP). The Refer target reports the Referred-By header in the INVITE to the host with in the RFS.

API Messages Used

- *PPL Event Request (0x0044)*
- *PPL Event Indication (0x0043)*
- *Outseize Control (0x002C)*
- Route Control (0x00E8)
- *Request For Service with Data (0x002D)*

These messages support the following TLVs.

- *0x2929 - SIP Referred By Header URI TLV*
- *0x292A - SIP Referred By Header Parameters TLV*

API and CSA Configuring and Querying

This feature can be configured either using the API or CSA.

API Configuring and Querying

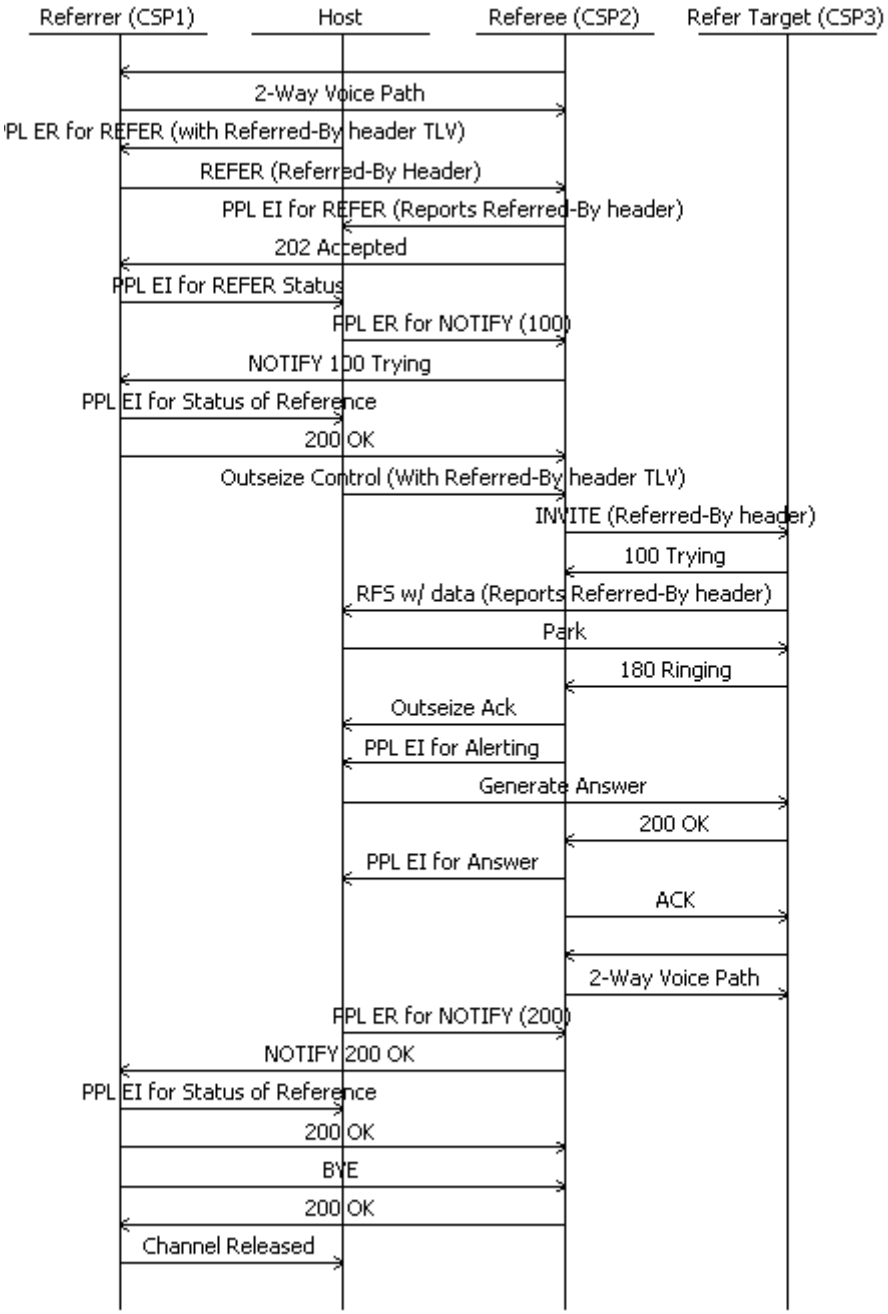
By default CSP does not report the Referred-By header, if it is present in the received REFER request or INVITE request message. To enable this feature, set bit 18 in the 0x027F - SIP Message Information Mask TLV and send it within the *VoIP Protocol Configure* (0x00EE) message.

See *0x027F - SIP Message Information Mask TLV*

CSA Configuring and Querying

To configure and query the SIP stack for this feature, view the Configure SIP Advanced screen, Additional Host Signaling Parameters, and select **Referred by Header in Refer and Invite**.

Call Flow - Insecure Refer The call flow below shows the insecure Refer scenario.



Notes

- The call flow assumes that Referrer and Referee are in answered state.
- All CSPs have the Referred-By header reporting enabled.
- The PPL ER for REFER to Referrer instructs CSP SIP stack to include Referred-By mechanism in the REFER request.
- The referee receiving the REFER request reports the Referred-by header to the host in the PPL Event Indication used to report the receipt of REFER message.
- The host uses this Referred-By header data received in the PPL Event Indication, within the Outseize Control message to instruct the CSP SIP stack to include the Referred-By header in the INVITE message.
- The Refer Target receiving the INVITE, reports the Referred-By header to host in the Request for Service message.

Message Trace

Below is the API and SIP trace for the call flow above but the Refer target in the trace is an Eyebeam softphone not a CSP.

```
CSP1:
H->X
    [00 83 00 2c 00 00 01 00 01 0d 03 00 64 1f 02 03 00 1e
   00 0f 00 02 01 16 00 02 00 00 01 1a 00 03 00 00 00 03
   00 33 00 5d 00 09 27 7e 00 03 08 00 00 29 19 00 06 32
   32 32 32 32 00 29 1b 00 0c 31 30 2e 31 30 2e 31 2e 32
   35 32 00 29 1c 00 04 00 00 13 c4 29 23 00 06 31 31 31
   31 31 00 29 25 00 0c 31 30 2e 31 30 2e 31 2e 32 35 30
   00 29 26 00 04 00 00 13 c4 27 92 00 04 0a 0a 01 bf 27
   93 00 04 00 00 10 7c]

X->H
    [00 07 00 2c 00 00 01 00 10]

X->H
    [00 20 00 43 00 09 01 00 01 0d 03 00 64 1f 00 a7 00 24
   01 03 00 33 00 0a 00 01 29 4a 00 04 00 00 00 00]

H->X
    [00 05 00 43 00 09 01]

H->X
    [00 05 00 43 00 09 01]

X->H
```

```
    [00 58 00 43 00 0a 01 00 01 0d 03 00 64 1f 00 a7 00 20
   01 03 00 33 00 42 00 03 29 2b 00 06 32 32 32 32 32 00
   29 2d 00 06 32 32 32 32 32 00 29 ff 00 28 2a 0e 00 04
   0a 0a 01 ab 2a 01 00 1c 2a 03 00 01 00 2a 07 00 04 00
   00 3d 7c 2a 13 00 01 00 2a 02 00 06 2a 08 00 02 00 02]

H->X
    [00 05 00 43 00 0a 01]

H->X
    [00 05 00 43 00 0a 01]

H->X
    [00 48 00 44 00 00 01 00 01 0d 03 00 64 1f 00 a7 00 27
   01 03 00 33 00 32 00 05 29 1e 00 06 32 32 32 32 32 00
   29 20 00 0c 31 30 2e 31 30 2e 31 2e 32 35 32 00 29 21
   00 04 00 00 13 c4 29 29 00 00 29 2a 00 06 70 61 72 61
   6d 00]

X->H
    [00 07 00 44 00 00 01 00 10]

X->H
    [00 1e 00 43 00 0b 01 00 01 0d 03 00 64 1f 00 a7 00 27
   01 03 00 33 00 08 00 01 29 15 00 02 00 ca]

H->X
    [00 05 00 43 00 0b 01]

H->X
    [00 05 00 43 00 0b 01]

X->H
    [00 25 00 43 00 0c 01 00 01 0d 03 00 64 1f 00 a7 00 28
   01 03 00 33 00 0f 00 02 29 15 00 02 00 64 29 48 00 03
   02 00 00]

H->X
    [00 05 00 43 00 0c 01]

H->X
    [00 05 00 43 00 0c 01]

X->H
    [00 30 00 43 00 0d 01 00 01 0d 03 00 64 1f 00 a7 00 28
   01 03 00 33 00 1a 00 02 29 15 00 02 00 c8 29 48 00 0e
   00 00 00 6e 6f 72 65 73 6f 75 72 63 65 00]

H->X
    [00 05 00 43 00 0d 01]

H->X
```

```
    [00 05 00 43 00 0d 01]

H->X
    [00 11 00 08 00 00 01 00 02 0d 03 00 64 1f 0d 03 00 64
     1f]

X->H
    [00 07 00 08 00 00 01 00 10]

X->H
    [00 57 00 69 00 02 01 00 01 0d 03 00 64 1f 02 02 1e 2a
   00 05 01 04 00 04 00 00 00 00 01 05 00 04 00 00 01 14
   01 11 00 04 00 00 01 15 01 10 00 04 00 00 ac 80 01 12
   00 04 00 00 ad 20 03 00 33 00 18 00 03 27 4e 00 02 00
   10 27 92 00 04 0a 0a 01 bf 27 93 00 04 00 00 10 7c]

H->X
    [00 05 00 69 00 02 01]

H->X
    [00 05 00 69 00 02 01]
2
 Printing all SIP messages

1 -RECEIVED From 10.10.1.2:5060 at 490

2 -SENT To 10.10.1.252:5060 at 494
INVITE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
To: 22222<sip:22222@10.10.1.252:5060>
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
Contact: 11111<sip:11111@10.10.1.250:5060>
User-Agent: Excel_CSP/83.10.15
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.250
s=SIP_Call
c=IN IP4 10.10.1.191
t=0 0
m=audio 4220 RTP/AVP 0

3 -RECEIVED From 10.10.1.252:5060 at 494
SIP/2.0 100 Trying
```

```
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

4 -RECEIVED From 10.10.1.252:5060 at 494
SIP/2.0 180 Ringing
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

5 -RECEIVED From 10.10.1.252:5060 at 495
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Require: timer
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1152893505 1152893505 IN IP4 10.10.1.252
s=SIP_Call
c=IN IP4 10.10.1.171
t=0 0
m=audio 15740 RTP/AVP 0
a=sendrecv

6 -SENT To 10.10.1.252:5060 at 495
ACK sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 1 ACK
```

```
Content-Length: 0

7 -SENT To 10.10.1.252:5060 at 497
REFER sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 2 REFER
Max-Forwards: 70
Contact: 11111<sip:11111@10.10.1.250:5060>
Refer-To: <sip:22222@10.10.1.252:5060>
Referred-By: <sip:11111@10.10.1.250:5060>;param
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

8 -RECEIVED From 10.10.1.252:5060 at 497
SIP/2.0 202 Accepted
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 2 REFER
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

9 -RECEIVED From 10.10.1.252:5060 at 497
NOTIFY sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
From: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 3 NOTIFY
Event: refer
Contact: 22222<sip:22222@10.10.1.252:5060>
Subscription-State: active
Content-Type: message/sipfrag;version=2.0
Content-Length: 20

SIP/2.0 100 Trying

10-SENT To 10.10.1.252:5060 at 497
SIP/2.0 200 OK
To: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
From: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 3 NOTIFY
Via: SIP/2.0/UDP 10.10.1.252
User-Agent: Excel_CSP/83.10.15
```

```
Content-Length: 0

11-RECEIVED From 10.10.1.2:5060 at 499

12-RECEIVED From 10.10.1.252:5060 at 500
NOTIFY sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
From: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 4 NOTIFY
Event: refer
Contact: 22222<sip:22222@10.10.1.252:5060>
Subscription-State: terminated;reason=noresource
Content-Type: message/sipfrag;version=2.0
Content-Length: 16

SIP/2.0 200 OK

13-SENT To 10.10.1.252:5060 at 500
SIP/2.0 200 OK
To: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
From: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 4 NOTIFY
Via: SIP/2.0/UDP 10.10.1.252
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

14-SENT To 10.10.1.252:5060 at 500
BYE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 3 BYE
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

15-RECEIVED From 10.10.1.252:5060 at 500
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 3 BYE
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Length: 0
```

```
CSP2:
X->H
     [01 6e 00 2d 00 02 02 00 01 0d 03 00 c8 02 00 33 01
   03 00 33 01 5a 00 1a 27 4e 00 02 00 05 27 7e 00 03 08
   00 00 29 19 00 06 32 32 32 32 32 00 29 1b 00 0c 31 30
   2e 31 30 2e 31 2e 32 35 32 00 29 1c 00 04 00 00 13 c4
   29 23 00 06 31 31 31 31 31 00 29 25 00 0c 31 30 2e 31
   30 2e 31 2e 32 35 30 00 29 26 00 04 00 00 13 c4 29 28
   00 06 31 31 31 31 31 00 29 2d 00 06 31 31 31 31 31 00
   29 2f 00 0c 31 30 2e 31 30 2e 31 2e 32 35 30 00 29 30
   00 04 00 00 13 c4 29 33 00 01 01 27 18 00 08 02 00 00
   00 05 11 11 10 27 17 00 06 02 00 05 22 22 20 29 16 00
   01 01 29 54 00 06 32 32 32 32 32 00 29 55 00 0c 31 30
   2e 31 30 2e 31 2e 32 35 32 00 29 56 00 04 00 00 13 c4
   29 50 00 24 45 58 43 45 4c 2d 43 53 50 31 2e 31 30 31
   66 2e 34 39 34 2e 31 38 30 40 31 30 2e 31 30 2e 31 2e
   32 35 30 00 29 51 00 0c 34 31 32 37 32 36 38 32 31 65
   65 00 29 52 00 08 32 36 38 32 31 65 62 00 2a 0e 00 04
   0a 0a 01 bf 29 9a 00 1a 29 9b 00 02 00 22 29 9c 00 10
   29 5e 00 0c 31 30 2e 31 30 2e 31 2e 32 35 30 00 29 4a
   00 04 00 00 00 01 29 ff 00 23 2a 0e 00 04 0a 0a 01 bf
   2a 01 00 17 2a 03 00 01 00 2a 07 00 04 00 00 10 7c 2a
   02 00 06 2a 08 00 02 00 02]

H->X
    [00 0c 00 2d 00 02 02 00 01 0d 03 00 c8 02]

H->X
    [00 0c 00 2d 00 02 02 00 01 0d 03 00 c8 02]

H->X
    [00 11 00 bf 00 00 02 00 02 0d 03 00 c8 02 0d 03 00 c8
     02]

X->H
    [00 07 00 bf 00 00 02 00 10]

H->X
    [00 0d 00 ba 00 00 02 00 01 0d 03 00 c8 02 01]

X->H
    [00 07 00 ba 00 00 02 00 10]

X->H
    [00 77 00 43 00 04 02 00 01 0d 03 00 c8 02 00 a7 00
   21 01 03 00 33 00 61 00 07 29 2b 00 06 31 31 31 31 31
   00 29 2d 00 06 31 31 31 31 31 00 29 19 00 06 32 32 32
   32 32 00 29 1b 00 0c 31 30 2e 31 30 2e 31 2e 32 35 32
   00 29 1c 00 04 00 00 13 c4 29 29 00 1b 73 69 70 3a 31
   31 31 31 31 40 31 30 2e 31 30 2e 31 2e 32 35 30 3a 35
   30 36 30 00 29 2a 00 06 70 61 72 61 6d 00]

H->X
    [00 05 00 43 00 04 02]
```

```
H->X
    [00 05 00 43 00 04 02]

H->X
    [00 1e 00 44 00 00 02 00 01 0d 03 00 c8 02 00 a7 00
  20 01 03 00 33 00 08 00 01 29 4b 00 02 00 64]

X->H
    [00 07 00 44 00 00 02 00 10]

H->X
    [00 a6 00 2c 00 01 02 00 01 0d 03 00 c8 1f 02 03 00
  1e 00 0f 00 02 01 16 00 02 00 00 01 1a 00 03 00 0d 00
  03 00 33 00 80 00 0b 27 7e 00 03 08 00 00 29 19 00 06
  32 30 35 39 37 00 29 1b 00 0a 31 30 2e 31 30 2e 31 2e
  32 00 29 1c 00 04 00 00 13 c4 29 23 00 06 32 32 32 32
  32 00 29 25 00 0c 31 30 2e 31 30 2e 31 2e 32 35 32 00
  29 26 00 04 00 00 13 c4 27 92 00 04 0a 0a 01 ac 27 93
  00 04 00 00 20 8c 29 29 00 17 31 31 31 31 31 40 31 30
  2e 31 30 2e 31 2e 32 35 30 3a 35 30 36 30 00 29 2a 00
  06 70 61 72 61 6d 00]

X->H
    [00 07 00 2c 00 01 02 00 10]

X->H
    [00 20 00 43 00 05 02 00 01 0d 03 00 c8 1f 00 a7 00
  24 01 03 00 33 00 0a 00 01 29 4a 00 04 00 00 00 00]

H->X
    [00 05 00 43 00 05 02]

H->X
    [00 05 00 43 00 05 02]

X->H
    [00 5b 00 43 00 06 02 00 01 0d 03 00 c8 1f 00 a7 00
  20 01 03 00 33 00 45 00 02 29 2d 00 06 32 30 35 39 37
  00 29 ff 00 35 2a 0e 00 04 0a 0a 01 02 2a 01 00 29 2a
  03 00 01 00 2a 07 00 04 00 00 20 72 2a 13 00 01 00 2a
  02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04
  00 00 1f 40]

H->X
    [00 05 00 43 00 06 02]

H->X
    [00 05 00 43 00 06 02]

H->X
```

```
    [00 1e 00 44 00 00 02 00 01 0d 03 00 c8 02 00 a7 00
  20 01 03 00 33 00 08 00 01 29 4b 00 02 00 c8]

X->H
    [00 07 00 44 00 00 02 00 10]

X->H
    [00 57 00 69 00 02 02 00 01 0d 03 00 c8 02 02 02 1e
  2a 00 05 01 04 00 04 00 00 00 00 01 05 00 04 00 00 01
  13 01 11 00 04 00 00 01 20 01 10 00 04 00 00 ab e0 01
  12 00 04 00 00 b4 00 03 00 33 00 18 00 03 27 4e 00 02
  00 10 27 92 00 04 0a 0a 01 ab 27 93 00 04 00 00 3d 7c]

H->X
    [00 05 00 69 00 02 02]

H->X
    [00 05 00 69 00 02 02]

H->X
    [00 11 00 08 00 00 02 00 02 0d 03 00 c8 1f 0d 03 00 c8
     1f]

X->H
    [00 07 00 08 00 00 02 00 10]

X->H
    [00 57 00 69 00 03 02 00 01 0d 03 00 c8 1f 02 02 1e
  2a 00 05 01 04 00 04 00 00 00 00 01 05 00 04 00 00 00
  6c 01 11 00 04 00 00 00 7f 01 10 00 04 00 00 43 80 01
  12 00 04 00 00 4f 60 03 00 33 00 18 00 03 27 4e 00 02
  00 10 27 92 00 04 0a 0a 01 ac 27 93 00 04 00 00 20 8c]

H->X
    [00 05 00 69 00 03 02]
2
 Printing all SIP messages

1 -RECEIVED From 10.10.1.250:5060 at 491
INVITE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
To: 22222<sip:22222@10.10.1.252:5060>
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
Contact: 11111<sip:11111@10.10.1.250:5060>
User-Agent: Excel_CSP/83.10.15
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
```

```
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.250
s=SIP_Call
c=IN IP4 10.10.1.191
t=0 0
m=audio 4220 RTP/AVP 0

2 -SENT To 10.10.1.250:5060 at 491
SIP/2.0 100 Trying
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

3 -SENT To 10.10.1.250:5060 at 492
SIP/2.0 180 Ringing
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

4 -SENT To 10.10.1.250:5060 at 492
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Require: timer
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1152893505 1152893505 IN IP4 10.10.1.252
s=SIP_Call
c=IN IP4 10.10.1.171
```

```
t=0 0
m=audio 15740 RTP/AVP 0
a=sendrecv

5 -RECEIVED From 10.10.1.250:5060 at 493
ACK sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 1 ACK
Content-Length: 0

 6 -RECEIVED From 10.10.1.250:5060 at 495
REFER sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 2 REFER
Max-Forwards: 70
Contact: 11111<sip:11111@10.10.1.250:5060>
Refer-To: <sip:22222@10.10.1.252:5060>
Referred-By: <sip:11111@10.10.1.250:5060>;param
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

 7 -SENT To 10.10.1.250:5060 at 495
SIP/2.0 202 Accepted
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 2 REFER
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

8 -SENT To 10.10.1.250:5060 at 495
NOTIFY sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
From: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 3 NOTIFY
Event: refer
Contact: 22222<sip:22222@10.10.1.252:5060>
Subscription-State: active
Content-Type: message/sipfrag;version=2.0
Content-Length: 20
```

```
SIP/2.0 100 Trying

9 -RECEIVED From 10.10.1.250:5060 at 495
SIP/2.0 200 OK
To: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
From: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 3 NOTIFY
Via: SIP/2.0/UDP 10.10.1.252
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

10-SENT To 10.10.1.2:5060 at 495
INVITE sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: 20597<sip:20597@10.10.1.2:5060>
From: 22222<sip:22222@10.10.1.252:5060>;tag=412786241ef
Call-ID: EXCEL-CSP2.101f.495.990@10.10.1.252
Contact: 22222<sip:22222@10.10.1.252:5060>
User-Agent: Excel_CSP/83.10.15
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Referred-By: <11111@10.10.1.250:5060>;param
Content-Type: application/sdp
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.252
s=SIP_Call
c=IN IP4 10.10.1.172
t=0 0
m=audio 8332 RTP/AVP 0

11-RECEIVED From 10.10.1.2:5060 at 496
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.10.1.252
Contact: <sip:20597@10.10.1.2:5060>
To: "20597"<sip:20597@10.10.1.2:5060>;tag=7204a018
From:
  "22222"<sip:22222@10.10.1.252:5060>;tag=412786241ef
Call-ID: EXCEL-CSP2.101f.495.990@10.10.1.252
CSeq: 1 INVITE
User-Agent: eyeBeam release 3010n stamp 19039
Content-Length: 0

12-RECEIVED From 10.10.1.2:5060 at 498
SIP/2.0 200 OK
```

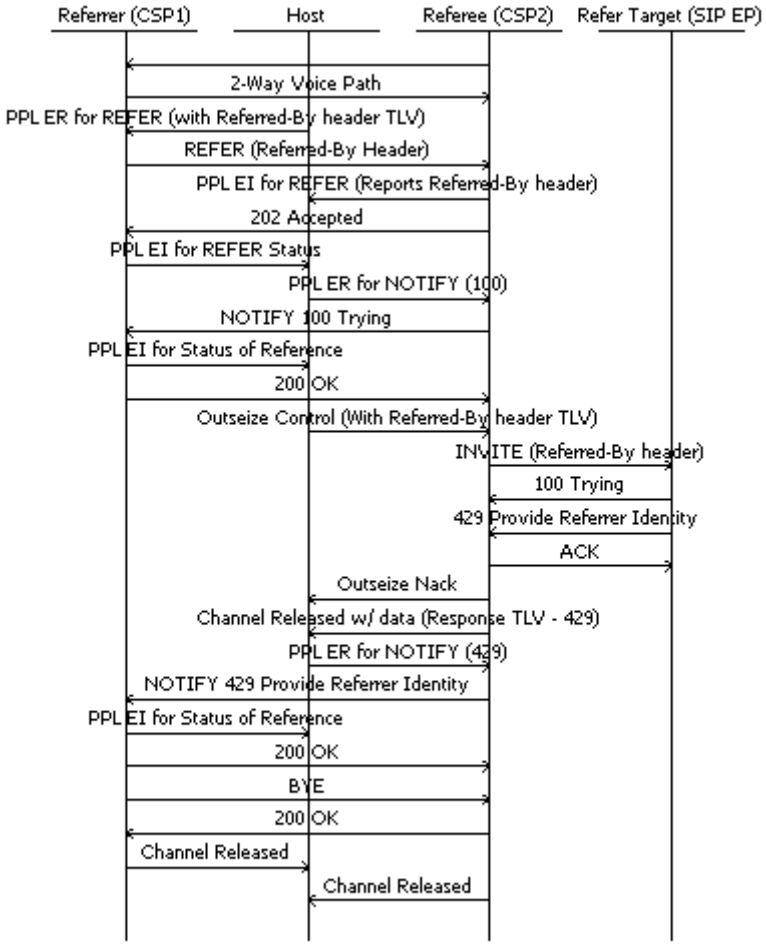
```
Via: SIP/2.0/UDP 10.10.1.252
Contact: <sip:20597@10.10.1.2:5060>
To: "20597"<sip:20597@10.10.1.2:5060>;tag=7204a018
From:
  "22222"<sip:22222@10.10.1.252:5060>;tag=412786241ef
Call-ID: EXCEL-CSP2.101f.495.990@10.10.1.252
CSeq: 1 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY,
  MESSAGE, SUBSCRIBE, INF
0
Content-Type: application/sdp
Supported: eventlist
User-Agent: eyeBeam release 3010n stamp 19039
Content-Length: 187

v=0
o=- 87437649 87437674 IN IP4 10.10.1.2
s=eyeBeam
c=IN IP4 10.10.1.2
t=0 0
m=audio 8306 RTP/AVP 0
a=alt:1 1 : 1DE6CC47 0000007B 10.10.1.2 8306
a=rtpmap:0 pcmu/8000
a=sendrecv

13-SENT To 10.10.1.2:5060 at 498
ACK sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: 20597<sip:20597@10.10.1.2:5060>;tag=7204a018
From: 22222<sip:22222@10.10.1.252:5060>;tag=412786241ef
Call-ID: EXCEL-CSP2.101f.495.990@10.10.1.252
CSeq: 1 ACK
Content-Length: 0

14-SENT To 10.10.1.250:5060 at 498
NOTIFY sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
From: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 4 NOTIFY
Event: refer
Contact: 22222<sip:22222@10.10.1.252:5060>
Subscription-State: terminated;reason=noresource
Content-Type: message/sipfrag;version=2.0
Content-Length: 16

SIP/2.0 200 OK

15-RECEIVED From 10.10.1.250:5060 at 498
```

```
SIP/2.0 200 OK
To: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
From: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 4 NOTIFY
Via: SIP/2.0/UDP 10.10.1.252
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

16-RECEIVED From 10.10.1.250:5060 at 498
BYE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.250
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 3 BYE
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

17-SENT To 10.10.1.250:5060 at 498
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=26821eb
From: 11111<sip:11111@10.10.1.250:5060>;tag=412726821ee
Call-ID: EXCEL-CSP1.101f.494.180@10.10.1.250
CSeq: 3 BYE
Via: SIP/2.0/UDP 10.10.1.250
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

18-SENT To 10.10.1.2:5060 at 500
BYE sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: "20597"<sip:20597@10.10.1.2:5060>;tag=7204a018
From:
  "22222"<sip:22222@10.10.1.252:5060>;tag=412786241ef
Call-ID: EXCEL-CSP2.101f.495.990@10.10.1.252
CSeq: 2 BYE
User-Agent: Excel_CSP/83.10.15
Content-Length: 0

19-RECEIVED From 10.10.1.2:5060 at 500
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.10.1.252
Contact: <sip:20597@10.10.1.2:5060>
To: "20597"<sip:20597@10.10.1.2:5060>;tag=7204a018
From:
  "22222"<sip:22222@10.10.1.252:5060>;tag=412786241ef
Call-ID: EXCEL-CSP2.101f.495.990@10.10.1.252
CSeq: 2 BYE
User-Agent: eyeBeam release 3010n stamp 19039
```

```
Content-Length: 0
```

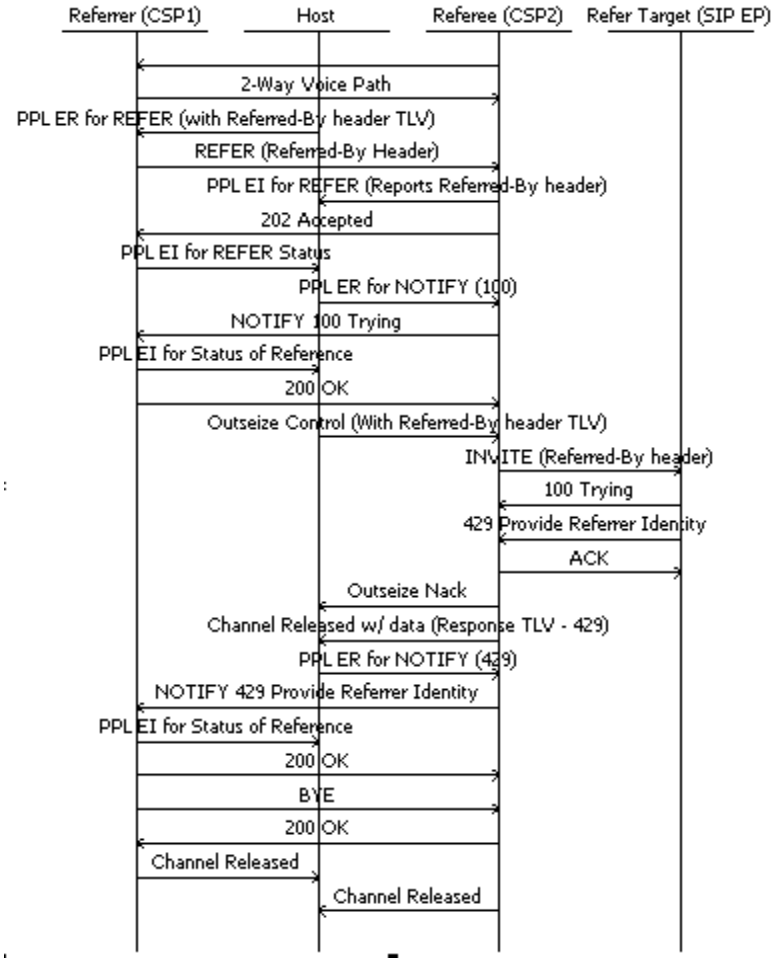
Call Flow - How to handle REFER with cid token (Secure REFER)

The call flow below provides an example how to inform the Referrer that the Referee does not support secure REFER. The Referred-By header in REFER request with “cid” token shall be reported to the host. The host is expected to parse and find out if this a secure (“cid” token present) or insecure REFER request. In case of secure REFER, the host can use PPL ER to generate NOTIFY with a message/sipfrag body indicating a final response of 501 “Not Implemented” to the Referrer. The host decides the final response to be used in the NOTIFY message body. Response 501 is an example:



Call Flow - How to handle 429 response for INVITE w/ Referred-By header

The call flow below shows an example of how to handle a 429 response for a INVITE w/ Referred-By header. If the INVITE message receives a 3xx-6xx response, the call is dropped and the channel is released. The channel-released API contains the response received for INVITE within the NPDI SIP Response Code TLV (0x2915). If the response is 429 “Provide Referrer Identity”, then the refer target requires a valid Referred-By token to accept the INVITE request (Secure REFER). This is informed to the Referrer using NOTIFY with a message/sipfrag body indicating a final response of 429.



New TLVs

0x027F - SIP Message Information Mask TLV

Used in:
VoIP Protocol Configure message
VoIP Protocol Query message

Byte	Description
0, 1	Tag 0x027F
2, 3	Length 0x0004
4-n	Value[0-3] This field is a 32-bit mask. Each bit selects specific SIP message fields and is listed from LSB to MSB. Bit 18 - Selects reporting of Referred-By header in REFER and INVITE request

0x2929 - SIP Referred By Header URI TLV

This TLV does the following:

- Instructs the CSP SIP stack to insert Referred-By header in the outbound REFER. In case no data is provided CSP SIP stack fills in the contact URI.
- Reports the URI in the Referred-By header, if present, in the received REFER or INVITE request.
- Writes the URI of the Referred-By header in the outbound INVITE request.

Used in:

Route Control
Outseize Control
Request For Service with Data
PPL Event Request
PPL Event Indication

Byte	Description
0, 1	Tag 0x2929
2, 3	Length Variable (Maximum is 100)
4-n	Value[0-3] Null Terminated ASCII string

0x292A - SIP Referred By Header Parameters TLV

This TLV does the following:

- Reports the Referred-By header parameters, if present, in the received REFER or INVITE request.
- Writes the parameters of the Referred-By header in the outbound REFER or INVITE request.

Used in:

Route Control
Outseize Control
Request For Service with Data
PPL Event Request
PPL Event Indication

Byte	Description	
0, 1	Tag	0x292A
2, 3	Length	Variable (Maximum is 100)
4-n	Value	Null Terminated ASCII string

Host Generated Refer Message

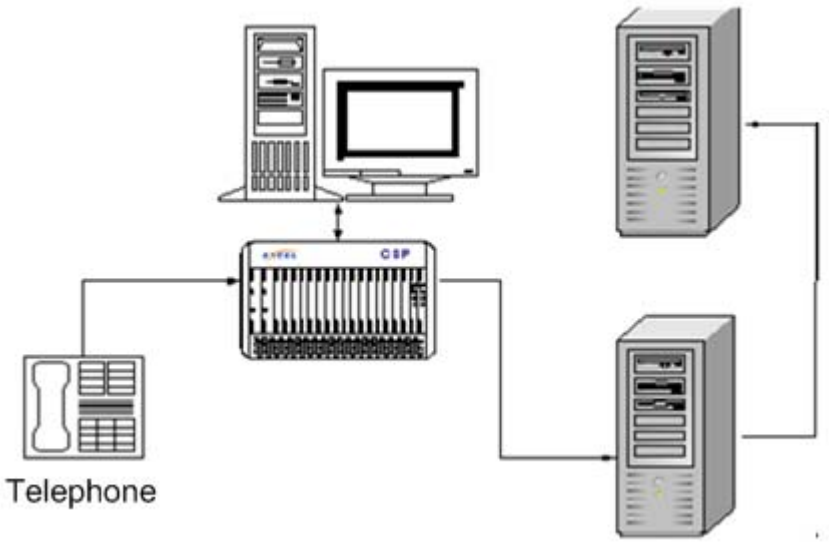
Overview

The SIP REFER message requests that the recipient contact a third party using the contact information in the request message. Prior to this Engineering Release, the CSP supported inbound REFER requests. With this release, the CSP now supports outbound REFER requests.

The CSP generates the outbound REFER message only when the host application triggers it.

The Figure 5-9 below shows the recipient contacting a third party after receiving the SIP REFER message from the CSP.

Figure 5-9 Recipient contacts third party



Generating Outbound REFER Request

The SIP software allows the CSP to generate an outbound REFER request when the host application sends a *PPL Event Request* (0x0044) message with Event 0x0027 to the SIP UA component (0x00A7).

The PPL Event Request must contain the target's URL in the following TLV which fills in Refer-To-Header field for the outbound request.

Mandatory

- NPDI SIP Refer to Username (0x291E)
- NPDI SIP Refer To Host Name (0x2920)

Optional

- NPDI SIP Refer To Port (0x2921)
- NPDI SIP Refer To Password (0x261F)

This feature also includes the following API changes.

- The existing SIP Response Code TLV (0x2915) is now supported in PPL Event Indications.
- PPL Event Request (0x0027) added to SIP UA component.
- PPL Event Indications (0x0027 and 0x0028) added to SIP UA component.

Call Flow

The following call flows show one successful host application generated REFER message and three unsuccessful scenarios. The explanation below the call flows explain each scenario. In all the four call flows, the Referrer and Referee are in the answered state.

Figure 5-10 Generated REFER Message Successful

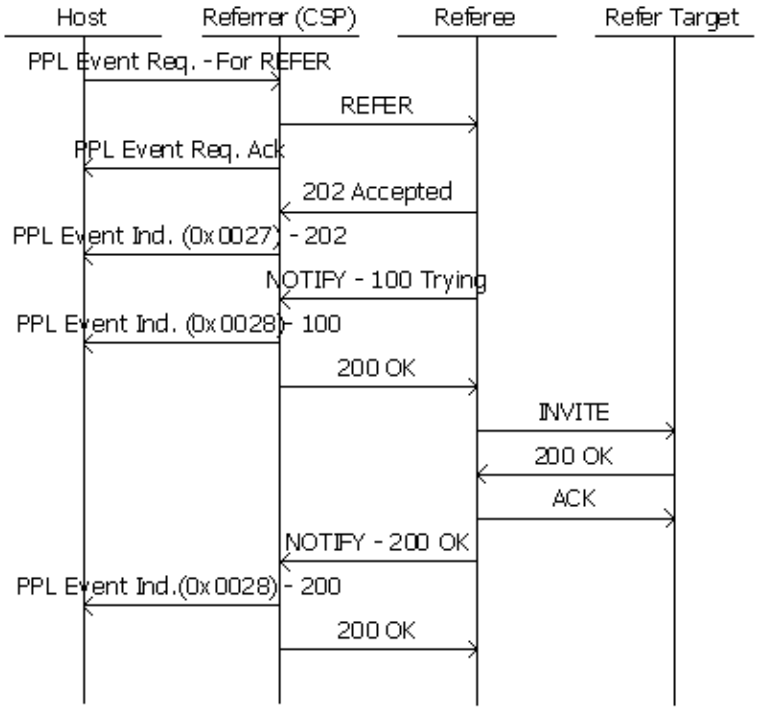


Figure 5-11 Generated REFER Message Failure Scenario 1

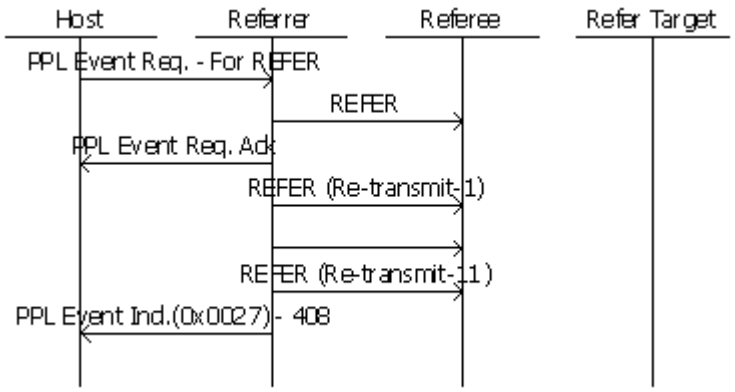


Figure 5-12 Generated REFER Message Failure Scenario 2

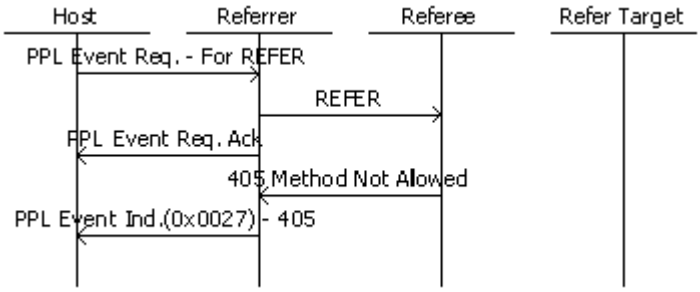
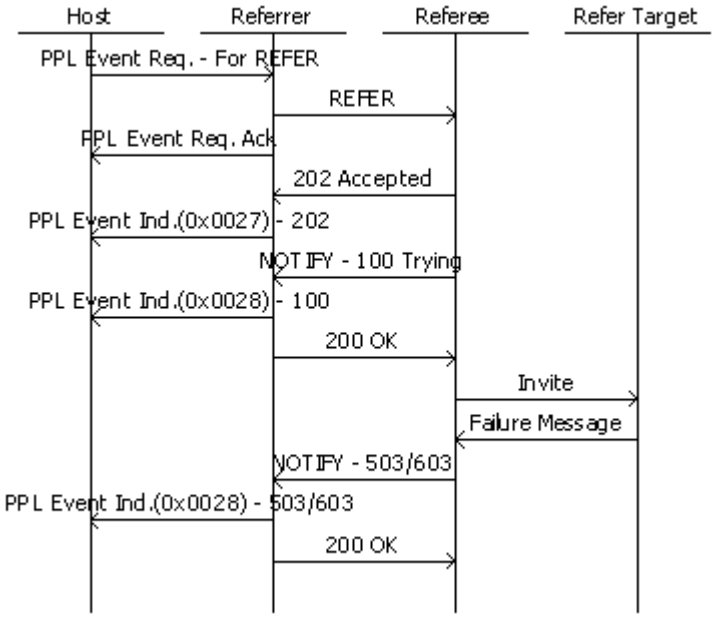


Figure 5-13 Reference Failure Scenario



Call Flow Steps

The following steps provide an overall explanation of the different call flow scenarios above.

1. The PPL Event Request message must have the Refer Target's address (SIP endpoint) in the following TLVs:

Mandatory TLVs:

- NPDI SIP Refer To Username TLV (0x291E).
- NPDI SIP Refer To Host Name TLV (0x2920)

Optional TLVs:

- NPDI SIP Refer To Password TLV (0x291F).
- NPDI SIP Refer To Port TLV (0x2921) - (Default value is 5060).

2. The CSP validates the data in the *PPL Event Request* message. If the validation is successful, a REFER request is constructed by the CSP and sent to the other end. The *PPL Event Request* message is ACKed by the CSP when the message is successfully sent from the referrer side.

3. The destination SIP UA (Referee) is supposed to send out a response when it receives the REFER request. If the REFER is accepted by the Referee, Referee responds with 202 response.

   For failures any appropriate 4xx-6xx class response is sent (In case the referee cannot accept the REFER, the referee responds with any appropriate 4xx-6xx status). When a response is received the referrer sends out a PPL Event Indication (0x0027) message with SIP Response Code TLV (0x2915) having the received response code (as shown in Figure 5-10, Figure 5-12, and Figure 5-13).
4. If the referee does not respond to the REFER request within 500 milliseconds (default value), the message is re-transmitted 11 times. If there are no responses even after 11 retries the referrer sends out a *PPL Event Indication* (0x0027) message with SIP Response Code TLV (0x2915) with response code 408-Request Timed out (as shown in Figure 5-11).
5. The host application has to interpret the SIP Response Code TLV (0x2915) data in PPL Event Indication 0x0027 as follows:

Response Values (TLV Data)	**Status of REFER**
202 (Accepted)	Success
408 (Request Timed Out)	Timed out
4xx-6xx	Failure

6. If the Referee (destination end-point) has accepted the REFER request, the Referee tries to contact the address in the Refer-To field. The status of this reference is passed on to the Referrer using the NOTIFY messages with the status in the body of the NOTIFY message. The Referrer receives the notification and informs the host application about the status of the reference using the PPL Event Indication 0x0028 - until the Refer-Notify subscription is terminated.
7. This *PPL Indication* message has the SIP Response Code TLV (0x2915), whose data part indicates the SIP response value (for example100 Trying, 200 OK, 603 Decline) received in the body of the Notify message. The host application has to interpret the SIP Response Code TLV (0x2915) data in PPL Event Indication message (0x0028) as follows:

Response Values (TLV Data)	Status of Reference
100 (Trying)	Trying
200 (OK)	Success
503 (Service Unavailable) 603 (Decline)	Failure

Refer-To Header Parameter Access

Overview

This feature allows access to the Refer-To parameters for both inbound and outbound calls where they are supplied in raw format.

Pertinent Specification

RFC 3515

Description

REFER is a SIP method defined by RFC 3261. The REFER method indicates that the recipient (identified by the Request -URI) should contact a third party using the contact information provided in the request.

The Refer-To is a request header field (request header) as defined by RFC 3515. It appears only in a REFER request. It provides a URL to reference.

The SIP stack in the CSP has the capability for inbound and outbound refer requests.

In addition to the Refer-To Parameters, the username, host name, and passwords are reported in a *PPL Event Indication* message for a Consultative REFER.

Prior to this feature, the SIP stack already supported the reporting of the username, host name, and passwords for a Blind REFER.

Refer-To Parameter Write Access

The Refer-To Parameters write access is within the context of SIP Refer message. The Excel SIP Stack allows the host to fill the Refer-To Parameters using the NPDI TLV, SIP Refer-To Header Parameter (0x2922) in the PPL Event Request (0x0044), SIP UA 0x00A7 with event ID 0x0027.

This functionality will work for data only up to 250 bytes (including the null). If the TLV (0x2922) is sent with data greater than 250 bytes then the PPL Event Request with be nacked with status 0x1304 (Invalid data).

API Messages

The following messages are used by this feature. Refer to the *API Reference* for the formats.

- *PPL Event Request (0x0044)*
  PPL Event Indication (0x0043)

PPL Information

Component 0x00A7 - PPL Event Indications:

- 0x0021 - Blind Refer PPL Event Indication
- 0x0022 - Consultative Refer PPL Event Indication

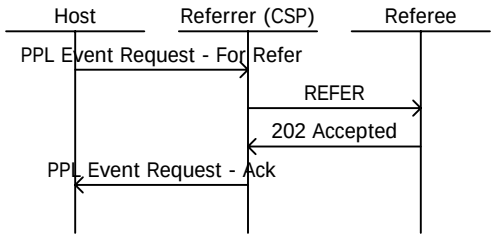
Configuring

You configure the Refer-To Parameters feature in the SIP stack. It is disabled by default.

The SIP Message Information Mask TLV (0x027F) is used in the *VoIP Protocol Configure* (0x00EE) message. Set bit 12 to enable this feature.

Call Flows

In the following diagram, the Referrer and Referee are in the answered state.



Example:

Parameters supplied in the form of TLVs by the user:

```
Username (0x291E): bob
Hostname (0x2920): biloxi.example.net
Parameter (0x2922):?Accept-
   Contact=sip:bobsdesk.biloxi.example.net&Call-
   ID%3D55432%40alicepc.atlanta.example.com1>;method=SUBS
   CRIBE
```

The final Header will look as follows:

```
Refer-To: <sip:bob@biloxi.example.net:5060?Accept-
   Contact=sip:bobsdesk.biloxi.example.net&Call-
   ID%3D55432%40alicepc.atlanta.example.com1>;method=SUBS
   CRIBE
```

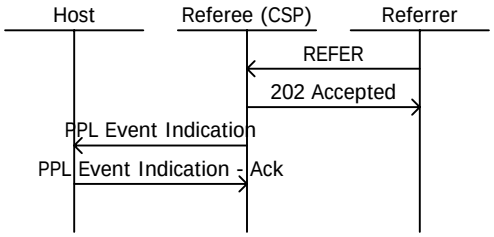
Refer-To Parameters read access

The SIP Stack will report the Refer-To Parameter, if it is present in the inbound Refer message, in raw, null-terminated ASCII format.

The reporting of the Refer-To Parameters is within the context of SIP Refer message. With the Refer-To Parameter reporting turned on, when the CSP receives a SIP REFER message, the Refer-To Parameters are reported as a null terminated string using NPDI TLV, SIP REFER-To Header Param (0x2922), in *PPL Event Indication* message (0x0043) with event id 0x0021 in case of Blind Refer or 0x0022 in case of consultative REFER.

This functionality will work for data only up to 250 bytes (including the null). If the received REFER message has Parameter greater than 250 bytes then the TLV will not be reported.

In the following diagram, the Referrer and Referee are in answered state.



Example

The inbound REFER message to has a Refer-To header as follows:

```
Refer-To:<sip:bob@biloxi.example.net:5060?Accept-
   Contact=sip:bobsdesk.biloxi.example.net&Call-
   ID%3D55432%40alicepc.atlanta.example.com1>;method=SUBS
   CRIBE
Then the PPL Event Indication will report the Refer-To
   header as
"?Accept-Contact=sip:bobsdesk.biloxi.example.net&Call-
   ID%3D55432%40alicepc.atlanta.example.com1>;method=SUBS
   CRIBE"
```

(Any data after the port number will be reported in this TLV.)

SIP Subject Header in REFER

Overview

Currently the RFC 3515 does not support Subject Header in the REFER message. With this feature, the Subject Header field is now an optional header in the REFER message. This feature allows read and write access for the Subject Header field in the REFER message.

Pertinent Specification

RFC 3515

Description

Read access allows the host to receive the content of the Subject Header field, if it is present in the inbound REFER message. You can configure whether or not to report the Subject Header field.

Write access allows the host to fill into the Subject Header field of the outbound REFER message. This functionality is available for Call Agent and non-Call Agent calls.

Subject Header Field Read/ Write Access

Read Access

The reporting of the Subject header field is within the context of the SIP REFER message. The Subject header field is reported using the SIP Subject TLV - 0295B in the *PPL Event Indication* for Blind Refer (PPL Component 0x00A7, Event 0x0027).

The SIP Subject TLV is used within the NPDI Universal ICB 0x0033. With the Subject Header field reporting enabled, when the CSP receives a SIP REFER that contains the Subject Header field, it will send a Host TLV (SIP Subject TLV - 0x295B) to the host application in the *PPL Event Indication* message. If the Subject Header field value is greater than 250 characters, then it will not be reported to the host.

Write Access

The write access is within the context of the SIP REFER message. The SIP Stack allows the host to fill the Subject Header field using the SIP Subject TLV - 0x295B in the *PPL Event Request* to generate the REFER message (PPL Component 0x00A7, Event 0x0021).

The SIP Subject TLV is used within the NPDI Universal ICB 0x0033. If the SIP Subject TLV exceeds 250, the PPL Event Request is NACKed with 0x1304.

API Messages

The following messages are used by this feature. Refer to the *API Reference* for the formats.

- *VoIP Protocol Configure* (0x00EE)
- *PPL Event Indication* (0x0043)
- *Request for Service with Data* (0x002D)

PPL Information

PPL Component 0x00A7:

- PPL Event Request 0x0027 - Generate outbound REFER message.

Configuring

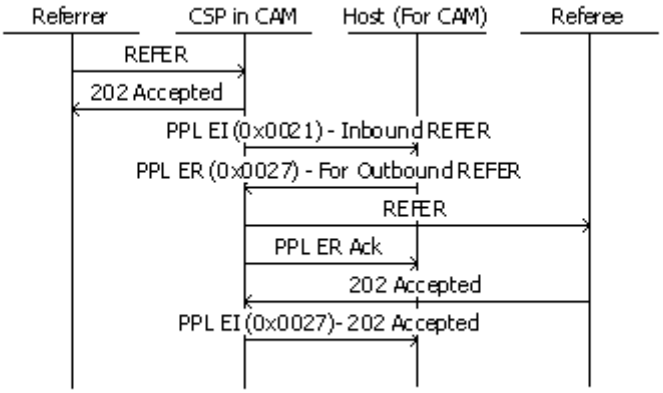
You can configure the SIP to report the Subject Header field from the REFER message. It is disabled by default. You enable it by setting a bit mask during the SIP stack configuration.

Use the SIP Message Information Mask TLV (0x027F) in the *VoIP Protocol Configure* (0x00EE) message. Set bit 11 to enable the feature.

Call Flow

This call flow assumes that the Referrer and Referee are in conversation. The Referee will contact another SIP Endpoint which is omitted in the figure below to keep it simple. This call flow also assumes that reporting Subject Header field is enabled.

The call flow below show both read and write access of Subject Header field for a REFER message. The PPL Event Indication (0x0021) reports the Subject Header field in the SIP Subject TLV. The PPL Event Request (0x0027) uses the SIP Subject TLV to generate a REFER message with the Subject Header field.



Report REFER Request URI in PPL Event Indication

Overview

The REFER message implements a call transfer service.

A user agent (referrer) uses the REFER message to request another user agent (referee) that it is in session-establish state with, to contact a third user agent (refer target).

The Refer target is identified by a SIP URI in the Refer-To header field in the REFER message.

You can enable this feature to get the Request-URI, in the Request-Line of the REFER message, reported to the host.

Example: With a SIP-to-ISDN call, if any ISDN related information like UUI, UUI encoding, or presentation indicator is present as a parameter in the Request-URI, this feature allows the host to get this data.

Pertinent Specification

RFC 3261

Description

This feature enhances the CSP SIP stack to report the Request-URI in the REFER message, including the parameters, to the host. This reporting is a configurable option in the CSP SIP stack and is disabled by default. This feature does not provide write access to Request-URI for a REFER message generated by CSP SIP stack.

Prior to this feature, the CSP SIP could not report the Request-URI in REFER to the host. When this feature is enabled, the Request-URI in the inbound REFER message is reported to the host using the following TLVs within the PPL Event Indication for REFER (Comp: 0x00A7, Event: 0x0021/0x0022 for Blind and Consultative REFER respectively)

- SIP Request URI User Name - 0x2954
- SIP Request URI Password - 0x2946
- SIP Request URI Host Name - 0x2955
- SIP Request URI Port - 0x2956
- SIP Request URI Parameters - 0x2958
- SIP Request URI Headers - 0x2947

Important! All TLVs except the port number TLV ends with a NULL terminator, therefore the actual data supported is one less

than the maximum length supported for the TLVs. Also note that the CSP nacks all SIP requests of a size greater that 1500 bytes with the 513 message, Too Large

API Call Control Messages

PPL Event Indication (0x0043)

API and CSA Configuring and Querying

This feature can be configured either using the API or CSA.

API Configuring and Querying

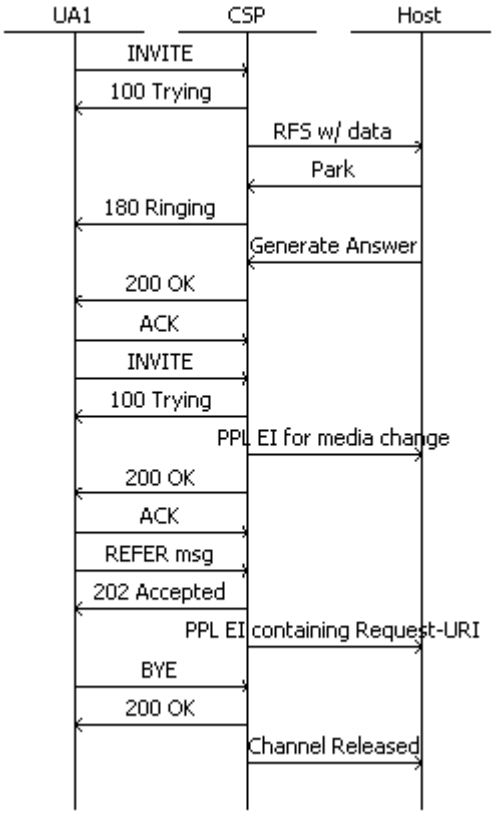
Reporting of Request-URI in the REFER message is a configurable option and is be disabled by default. To enable this reporting, set bit 21 of PPL Event Notification Mask TLV (0x027F) and send it in the *VoIP Protocol Configure* (0x00EE) message.

CSA Configuring and Querying

To configure and query the SIP stack for this feature, view the Configure SIP Advanced screen, Additional Host Signaling Parameters, and select **Report of Request-URI**.

Call Flow - Refer Request

The call flow below is a simplified scenario to depict this feature. Assume the CSP is enabled to report Request-URI in inbound REFER to the host. The CSP acts an endpoint, though in most scenarios it acts as a gateway. Once the CSP and UA1 are in session-established state (answered), UA1 puts the session on-hold (Re-INVITE), sends a REFER request to CSP and releases the call. The CSP accepts the Re-INVITE. After the CSP receives the REFER, it responds with the appropriate response (202 in the example) and reports the Request-URI to the host within the PPL Event Indication for the REFER message.



Call Trace - Refer Request

API Trace:

```
X->H
    [01 f2 00 2d 00 0c 01 00 01 0d 03 00 64 0c 00 33 01 03
   00 33 01 de 00 1a 27 4e 00 02 00 05 27 7e 00 03 08 00
   00 29 19 00 06 31 31 31 31 31 00 29 1b 00 0c 31 30 2e
   31 30 2e 31 2e 32 35 30 00 29 1c 00 04 00 00 13 c4 29
   23 00 06 32 30 35 39 37 00 29 25 00 0c 31 30 2e 31 30
   2e 31 2e 32 35 30 00 29 26 00 04 00 00 13 c4 29 28 00
   08 22 53 75 62 62 75 22 00 29 2d 00 06 32 30 35 39 37
   00 29 2f 00 0a 31 30 2e 31 30 2e 31 2e 32 00 29 30 00
   04 00 00 13 c4 29 33 00 01 01 27 18 00 08 02 00 00 00
   05 20 59 70 27 17 00 06 02 00 05 11 11 10 29 16 00 01
   01 29 54 00 06 31 31 31 31 31 00 29 55 00 0c 31 30 2e
   31 30 2e 31 2e 32 35 30 00 29 56 00 04 00 00 13 c4 29
   50 00 36 33 61 30 36 66 61 33 30 66 65 33 30 64 64 34
   33 40 63 33 4a 68 61 6d 56 75 5a 48 4a 68 4c 57 6c 69
   62 53 35 44 62 33 4a 77 4c 6e 68 73 4c 6d 4e 76 62 51
   2e 2e 00 29 51 00 09 33 39 30 65 37 34 30 31 00 29 52
   00 09 39 33 36 33 31 31 32 36 00 2a 0e 00 04 0a 0a 01
   02 29 9a 00 20 29 9b 00 02 00 22 29 9c 00 16 29 5e 00
```

```
   0a 31 30 2e 31 30 2e 31 2e 32 00 29 5f 00 04 00 00 13
   c4 29 4a 00 04 80 00 00 00 29 ff 00 91 2a 0e 00 04 0a
   0a 01 02 2a 01 00 85 2a 03 00 01 00 2a 07 00 04 00 00
   20 72 2a 13 00 01 00 2a 02 00 13 2a 08 00 02 00 02 2a
   09 00 01 02 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08
   00 02 00 01 2a 09 00 01 01 2a 0b 00 04 00 00 1f 40 2a
   02 00 13 2a 08 00 02 00 16 2a 09 00 01 16 2a 0b 00 04
   00 00 1f 40 2a 02 00 13 2a 08 00 02 00 12 2a 09 00 01
   12 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00
   17 2a 09 00 01 17 2a 0b 00 04 00 00 1f 40]

H->X
    [00 0c 00 2d 00 0c 01 00 01 0d 03 00 64 0c]

H->X
    [00 0c 00 2d 00 0c 01 00 01 0d 03 00 64 0c]

H->X
    [00 11 00 bf 00 00 01 00 02 0d 03 00 64 0c 0d 03 00 64
   0c]

X->H
    [00 07 00 bf 00 00 01 00 10]

H->X
    [00 0d 00 ba 00 00 01 00 01 0d 03 00 64 0c 01]

X->H
    [00 07 00 ba 00 00 01 00 10]

X->H
    [00 51 00 43 00 18 01 00 01 0d 03 00 64 0c 00 a7 00 1e
   01 03 00 33 00 3b 00 01 29 ff 00 35 2a 0e 00 04 00 00
   00 00 2a 01 00 29 2a 03 00 01 00 2a 07 00 04 00 00 20
   72 2a 13 00 01 02 2a 02 00 13 2a 08 00 02 00 02 2a 09
   00 01 02 2a 0b 00 04 00 00 1f 40]

H->X
    [00 05 00 43 00 18 01]

X->H
    [00 70 00 43 00 19 01 00 01 0d 03 00 64 0c 00 a7 00 21
   01 03 00 33 00 5a 00 08 29 2d 00 06 32 30 35 39 37 00
   29 19 00 06 31 32 33 34 35 00 29 1b 00 0c 31 30 2e 31
   30 2e 31 2e 32 35 30 00 29 1c 00 04 00 00 13 c4 29 29
   00 06 32 30 35 39 37 00 29 54 00 06 31 31 31 31 31 00
   29 55 00 0c 31 30 2e 31 30 2e 31 2e 32 35 30 00 29 56
   00 04 00 00 13 c4]

H->X
    [00 05 00 43 00 19 01]

X->H
```

```
    [00 57 00 69 00 0c 01 00 01 0d 03 00 64 0c 02 02 1e 2a
   00 05 01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 41
   01 11 00 04 00 00 01 1c 01 10 00 04 00 00 28 a0 01 12
   00 04 00 00 b1 80 03 00 33 00 18 00 03 27 4e 00 02 00
   10 27 92 00 04 0a 0a 01 bf 27 93 00 04 00 00 3e c4]

H->X
    [00 05 00 69 00 0c 01]

SIP Trace:
1 -RECEIVED From 10.10.1.2:5060 at 4390
INVITE sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   a35cc125786fec3e-1--d87543
-;rport
Max-Forwards: 70
Contact: <sip:20597@10.10.1.2:5060>
To: <sip:11111@10.10.1.250:5060>
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 1 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY,
   MESSAGE, SUBSCRIBE, INF
O
Content-Type: application/sdp
Supported: eventlist
User-Agent: eyeBeam release 3010n stamp 19039
Content-Length: 425

v=0
o=- 146862411 146862468 IN IP4 10.10.1.2
s=eyeBeam
c=IN IP4 10.10.1.2
t=0 0
m=audio 8306 RTP/AVP 100 0 8 3 18 98 97 5 101
a=alt:1 1 : EAF29E5C 000000DE 10.10.1.2 8306
a=fmtp:101 0-15
a=rtpmap:100 speex/16000
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:18 g729/8000
a=rtpmap:98 ilbc/8000
a=rtpmap:97 speex/8000
a=rtpmap:5 dvi4/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv

2 -SENT To 10.10.1.2:5060 at 4390
SIP/2.0 100 Trying
```

```
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 1 INVITE
Contact: 11111<sip:11111@10.10.1.250:5060>
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   a35cc125786fec3e-1--d87543
-;rport
User-Agent: Excel_CSP/83.10.66
Content-Length: 0


3 -SENT To 10.10.1.2:5060 at 4391
SIP/2.0 180 Ringing
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 1 INVITE
Contact: 11111<sip:11111@10.10.1.250:5060>
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   a35cc125786fec3e-1--d87543
-;rport
User-Agent: Excel_CSP/83.10.66
Content-Length: 0


4 -SENT To 10.10.1.2:5060 at 4391
SIP/2.0 200 OK
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 1 INVITE
Contact: 11111<sip:11111@10.10.1.250:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   a35cc125786fec3e-1--d87543
-;rport
User-Agent: Excel_CSP/83.10.66
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1155656630 1155656630 IN IP4 10.10.1.250
s=SIP_Call
c=IN IP4 10.10.1.191
t=0 0
m=audio 16068 RTP/AVP 0
```

```
a=sendrecv

5 -RECEIVED From 10.10.1.2:5060 at 4392
ACK sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   7f2675358918741f-1--d87543
-;rport
Max-Forwards: 70
Contact: <sip:20597@10.10.1.2:5060>
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 1 ACK
User-Agent: eyeBeam release 3010n stamp 19039
Content-Length: 0


6 -RECEIVED From 10.10.1.2:5060 at 4393
INVITE sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   b6777511fb20a408-1--d87543
-;rport
Max-Forwards: 70
Contact: <sip:20597@10.10.1.2:5060>
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 2 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY,
   MESSAGE, SUBSCRIBE, INF
0
Content-Type: application/sdp
Supported: eventlist
User-Agent: eyeBeam release 3010n stamp 19039
Content-Length: 187

v=0
o=- 146862411 146865057 IN IP4 10.10.1.2
s=eyeBeam
c=IN IP4 0.0.0.0
t=0 0
m=audio 8306 RTP/AVP 0
a=alt:1 1 : EAF29E5C 000000DE 10.10.1.2 8306
a=rtpmap:0 pcmu/8000
a=sendonly

7 -SENT To 10.10.1.2:5060 at 4393
SIP/2.0 100 Trying
```

```
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 2 INVITE
Contact: 11111<sip:11111@10.10.1.250:5060>
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   b6777511fb20a408-1--d87543
-;rport
User-Agent: Excel_CSP/83.10.66
Content-Length: 0


8 -SENT To 10.10.1.2:5060 at 4393
SIP/2.0 200 OK
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 2 INVITE
Contact: 11111<sip:11111@10.10.1.250:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   b6777511fb20a408-1--d87543
-;rport
User-Agent: Excel_CSP/83.10.66
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1155656630 1155656630 IN IP4 10.10.1.250
s=SIP_Call
c=IN IP4 10.10.1.191
t=0 0
m=audio 16068 RTP/AVP 0
a=sendrecv

9 -RECEIVED From 10.10.1.2:5060 at 4393
ACK sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   316c255f587e511b-1--d87543
-;rport
Max-Forwards: 70
Contact: <sip:20597@10.10.1.2:5060>
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 2 ACK
User-Agent: eyeBeam release 3010n stamp 19039
```

```
Content-Length: 0


10-RECEIVED From 10.10.1.2:5060 at 4395


11-RECEIVED From 10.10.1.2:5060 at 4396
REFER sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   011a1e4f53392f2d-1--d87543
-;rport
Max-Forwards: 70
Contact: <sip:20597@10.10.1.2:5060>
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 3 REFER
User-Agent: eyeBeam release 3010n stamp 19039
Refer-To: <sip:12345@10.10.1.250:5060>
Referred-By: <sip:20597>
Content-Length: 0


12-SENT To 10.10.1.2:5060 at 4396
SIP/2.0 202 Accepted
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 3 REFER
Contact: 11111<sip:11111@10.10.1.250:5060>
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   011a1e4f53392f2d-1--d87543
-;rport
User-Agent: Excel_CSP/83.10.66
Content-Length: 0


13-RECEIVED From 10.10.1.2:5060 at 4397
BYE sip:11111@10.10.1.250:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   97128e3806547b59-1--d87543
-;rport
Max-Forwards: 70
Contact: <sip:20597@10.10.1.2:5060>
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 4 BYE
```

```
User-Agent: eyeBeam release 3010n stamp 19039
Content-Length: 0


14-SENT To 10.10.1.2:5060 at 4397
SIP/2.0 200 OK
To: <sip:11111@10.10.1.250:5060>;tag=93631126
From: "Subbu"<sip:20597@10.10.1.250:5060>;tag=390e7401
Call-ID:
   3a06fa30fe30dd43@c3JhamVuZHJhLWlibS5Db3JwLnhsLmNvbQ..
CSeq: 4 BYE
Via: SIP/2.0/UDP 10.10.1.2:5060;branch=z9hG4bK-d87543-
   97128e3806547b59-1--d87543
-;rport
User-Agent: Excel_CSP/83.10.66
Content-Length: 0
```

Refer to Phone Number

Overview This feature allows the CSP to support a telephone number in an outbound REFER method.

Pertinent Specification RFC 3515

Description REFER is a SIP method as defined by RFC 3261 that indicates that the recipient (identified by the Request-URI) should contact a third party using the contact information provided in the request. Refer-To is a request header field with in the REFER method as defined by RFC 3515.

Prior to this feature, the CSP supports only the SIP URL in the Refer-To header. The host had to provide the username and host name in the PPL Event Request method to generate the REFER. By default CSP fills in the port number, in case the host has not supplied it. There was no provision for the host to use a telephone number in Refer-To header.

With this feature, the CSP supports telephone number in an outbound REFER method.

> **Important!** The telephone number supplied by the host is not validated by CSP.

How Supplied

The CSP SIP Stack allows the host to supply the telephone number in an outbound REFER method. The username TLV (0x291E) is optional in the PPL Event Request to generate REFER method.

Only host name TLV (0x2920) will be mandatory. The CSP will not fill the port number by default. To refer to a telephone number the host supplies only the host name TLV containing the telephone number. To refer to a SIP URL the host has to provide the host name TLV and username TLV.

Example:

The host name TLV supplied by the user for an outbound refer to a telephone number.

```
Hostname (0x2920): 080-23310818
```

(in the form of null terminated ASCII string)

The final Header will look as follows:

```
Refer-To: <sip:080-23310818>
```

SIP Population of Status in Outbound NOTIFY Message

Description

The SIP stack allows the CSP to send multiple NOTIFY messages.

The NOTIFY message follows a REFER message. If the reference is pending, then the host sends a PPL Event Request for NOTIFY with response class 1xx.

Similarly, the host sends response class 2xx if the reference was successful and 5xx or 6xx in the cases where the reference failed.

If the first PPL Event Request for the NOTIFY message has a 1xx response status, then the CSP checks for more PPL Event Request for NOTIFY messages from the host.

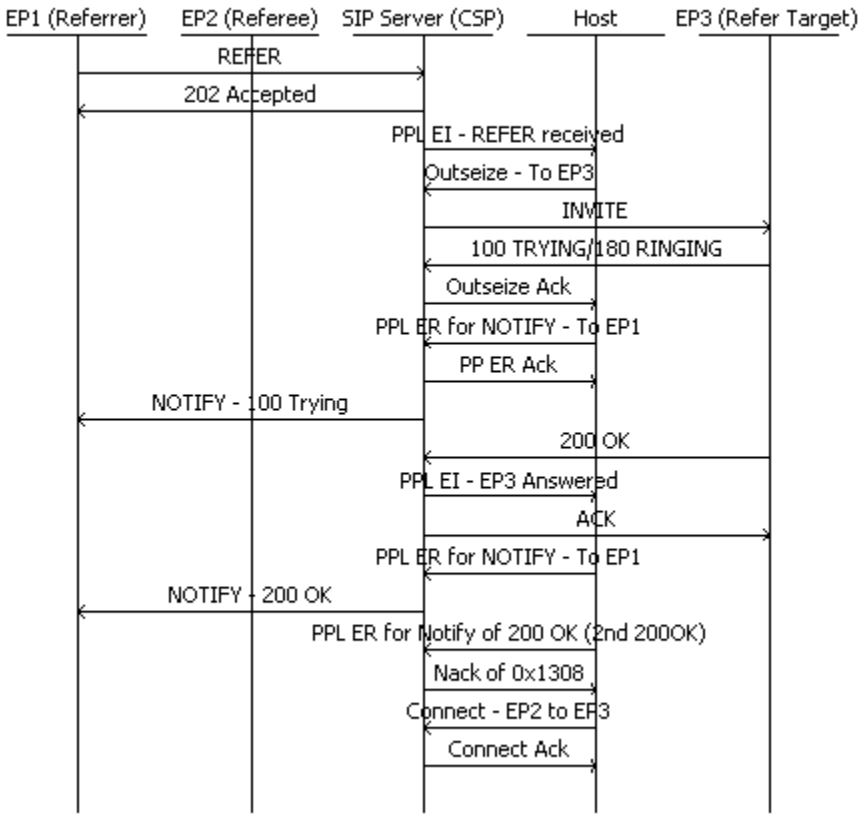
The CSP accepts and processes PPL Event Request for the NOTIFY message until it gets a PPL Event Request for the NOTIFY message with a final response status (2xx-6xx). Any subsequent PPL Event Request for the NOTIFY message after that will be nacked. The host can send any number of PPL Event Requests for the NOTIFY message with a 1xx response status before sending a final response.

API Message

The following message is used by this feature. Refer to the *API Reference* for the format.

PPL Event Request (0x0044)

Call Flow The following call flow assumes that the Referrer and the Referee are in the answered state.



SUBSCRIBE and NOTIFY Method for DTMF Detection

Overview

The CSP supports the SUBSCRIBE and NOTIFY methods.

If the remote SIP end points cannot multi-unicast the RFC 2833 stream, you can use the SIP SUBSCRIBE/NOTIFY method. This feature supports sending DTMF notifications via signaling.

The CSP is a subscriber and the remote end point is a notifier.

Pertinent Specification

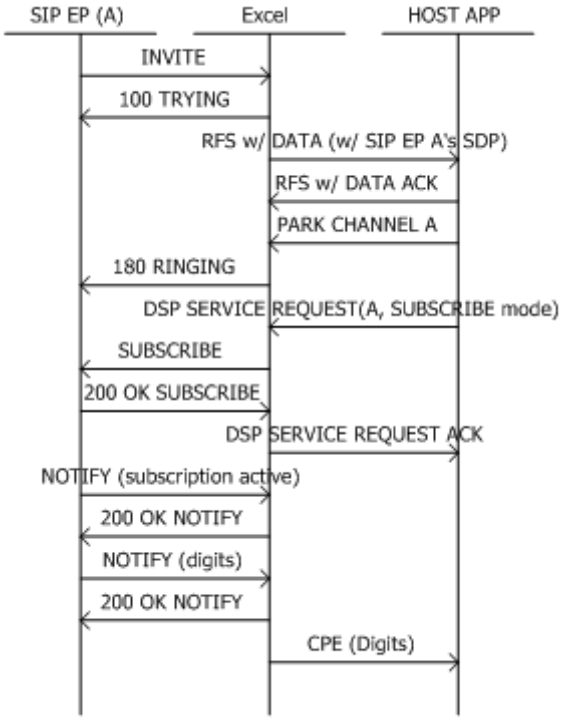
RFC 3265

Description

This feature works as follows:

1. When the host application issues a DTMF detection request in the *DSP Service Request* (0x00BD) message for a SIP call, the CSP tries to initiate a subscription for telephone events with the remote end point using the SUBSCRIBE method.
2. When the SIP remote end point accepts the subscription, the end point starts to report the digits in the SIP NOTIFY request.
3. The CSP reports these digits to the host application in the *Call Processing Event* (0x002E) message.
4. The host application can terminate a subscription by sending a *DSP Service Cancel Request* (0x00BE) message or by releasing the call.
5. At any time, a remote SIP endpoint can terminate an active subscription by sending a NOTIFY method with the “Subscription-State” header set to “terminated.”

Subscribe and Notify Call Flow



SIP Notify Subscription State

Overview

This features adds support for Subscription-State headers in REFER-NOTIFY requests in Call Agent Mode (CAM) and non-CAM configurations.

The benefits of this feature include:

- Host control on REFER-NOTIFY Subscription-State header.
- Host control on the duration of REFER-NOTIFY subscription ("expires" parameter).
- Host control on the retry-after time interval, which would be used to inform the remote UA when the REFER request could be retried in case the previous REFER failed.
- Customizable reason code in the Subscription-State header of REFER-NOTIFY that terminates the subscription.

Pertinent Specification

RFC 3265 and 3515

Description

The NOTIFY request for REFER now contains the Subscription-State header. This feature provides the following functionality:

- Access to Subscription-State header.
- Access to "expires" parameter when the subscription state is active.
- Access to reason code when the subscription state is terminated.
- Access to "retry-after" parameter when the subscription state is terminated unsuccessfully.

REFER-NOTIFY Subscription

Termination of the REFER-NOTIFY means either that the notifier (Referee) will not send any more REFER-NOTIFY messages or that the referrer will not accept any more REFER-NOTIFY requests once the REFER-NOTIFY subscription is terminated.

When the subscription is terminated the notifier CSP will nack the PPL the Event Request with message 0x1308. All REFER-NOTIFY requests to the referrer will receive a “481 Call/Transaction Does Not Exist” response.

API Call Control Messages

- *PPL Event Request* (0x0044)
- *PPL Event Indication* (0x0043)

API and CSA Configuring and Querying

This feature can be configured either using the API or CSA.

API Configuring and Querying

You can configure Subscription-State header reporting in a REFER-NOTIFY message. By default the CSP does not report the Subscription-State header if the header is present in the received REFER-NOTIFY message.

To enable the reporting, set bit 19 of the data part in SIP Message Information Mask TLV (0x027F) and send the TLV within the *VoIP Protocol Configure API* (0x00EE).

CSA Configuring and Querying

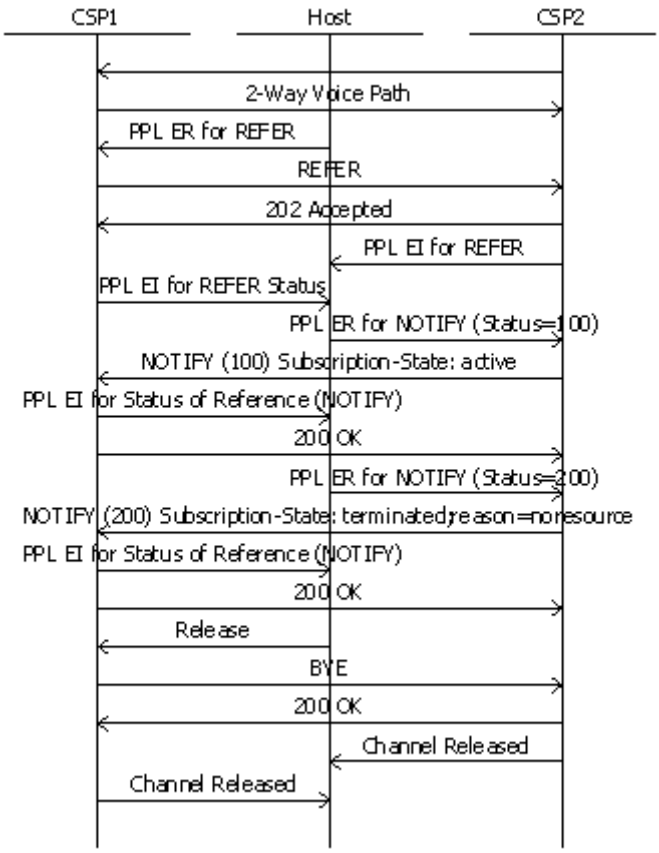
To configure and query the SIP stack for this feature, view the Configure SIP Advanced screen, Additional Host Signaling Parameters, and select **Subscription State Refer-Notify message**..

Call Flows

Blind REFER

The default PPL Event Request (that is, without the Notify Subscription State TLV) to generate NOTIFY for Blind REFER works as follows. Also, see the call flow diagram following these steps.

1. The NOTIFY will have a Subscription-State header with the string "active" if the host provides a provisional response status else it will be "terminated;reason=noresource" if the host provides a final response status.
2. Notifications for Blind refer do not contain "expires" parameter and hence the subscription never times out.
3. Notify with final response status or releasing a call would terminate the subscription.
4. If the NOTIFY gets a response in the range 3xx-6xx the subscription will be terminated.



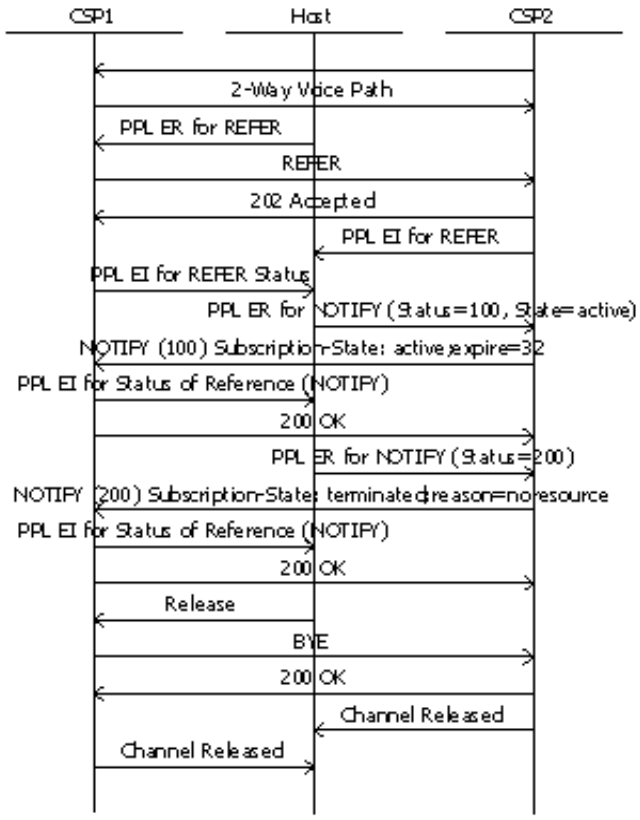
Consultative REFER

The default PPL Event Request (that is, without the Notify Subscription State TLV) to generate NOTIFY for Consultative REFER works as follows: Also, see the call flow diagram following these steps.

1. The NOTIFY will have a Subscription-State header with the string "active;expire=x" if the host provides a provisional response status where x is 64*T1 (for default T1 of 500 ms x is 32 s) else it will be "terminated;reason=noresource" if the host provides a final response status.
2. Notifications for Consultative refer would have a default expiry interval of 64*T1.
3. Notify with final response status, exceeding the time interval in the expires parameter or releasing a call would terminate the subscription.
4. A timer with interval equal to the "expires" parameter value will be started or restarted at the UAs each time NOTIFY is send/ received. The value of timer is the host supplied value or if host

has not supplied it would be the default value of 64*T1. NOTIFY with subscription state of terminated stops this timer in both the UAs.

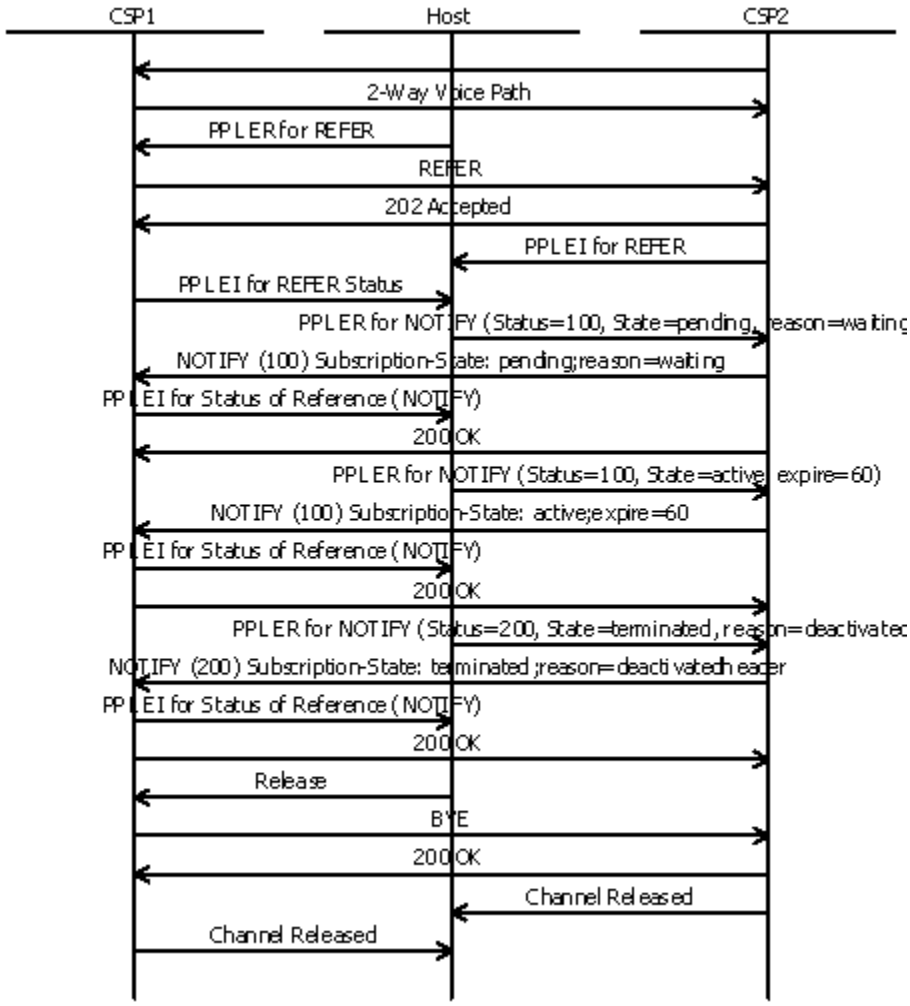
5. When the subscription expires the notifier CSP sends out a NOTIFY of status 603 Decline with Subscription-State header with string set to "terminated:reason=noresource".
6. When the subscription expires, the referrer CSP will terminate the subscription. No message will be send out to the remote UA.
7. If the NOTIFY gets a response in the range 3xx-6xx, the timer will be stopped and the subscription will be terminated.



Blind/Consultative

The PPL ER with the Notify Subscription State TLV duly filled to generate NOTIFY for REFER (Blind/Consultative) works in the following manner:

1. The NOTIFY will have a Subscription-State header with the string "active;expire=x" if the host either provides or sets the subscription state to active and expire parameter to 'x', irrespective of the response status provided.
2. The NOTIFY will have a Subscription-State header with the string "pending;reason=waiting" if the host either provides or sets the subscription state to pending and reason code to 'waiting', irrespective of the response status provided.
3. If a NOTIFY is generated when the subscription state is pending, its body should consist only of a status line containing a response code of 100 else the NOTIFY would be responded with 400 Bad Request.
4. The NOTIFY will have a Subscription-State header with the string "terminated;reason=noresource" if the host sets the subscription state to terminated and reason code to 'noresource, irrespective of the response status provided.
5. The CSP which receives the NOTIFY with expire parameter would terminate the subscription after the time interval expires. The CSP will not send any messages to inform the remote end about the termination.
6. A timer with interval equal to the "expires" parameter value, if present, will be started or restarted at the UAs each time NOTIFY is send/received. The value of timer is the host supplied value or if host has not supplied it would be the default value of 64*T1 seconds (default value only used for consultative refer).
7. The notifier CSP that sends NOTIFY with expire parameters would terminate the subscription after the time interval expires and also sends a NOTIFY of status 603 Decline with Subscription-State header string set to "terminated:reason=noresource".
8. The referrer CSP that received NOTIFY with expire parameters would terminate the subscription after the time expires. No messages would be send out to the remote UA.
9. If the NOTIFY gets a response in the range 3xx-6xx, the timer will be stopped and the subscription will be terminated.
10. The "retry-after" parameter will be added only if the host provides it in the PPL ER for REFER-NOTIFY that terminates the subscription. CSP will not add this by itself in any case.
11. The following figures show two instances of REFER-NOTIFY call flow.



Message Trace The trace for call flow in figure 4 is given below.

```
CSP 1 API Trace:
H->X
    [00 88 00 2c 00 00 01 00 01 0d 03 00 64 1f 02 03 00 1e
   00 0f 00 02 01 16 00 02 00 00 01 1a 00 03 00 00 00 03
   00 33 00 62 00 0a 27 7e 00 03 08 00 00 29 19 00 06 32
   32 32 32 32 00 29 1b 00 0c 31 30 2e 31 30 2e 31 2e 32
   35 32 00 29 1c 00 04 00 00 13 c4 29 23 00 06 31 31 31
   31 31 00 29 25 00 0c 31 30 2e 31 30 2e 31 2e 32 35 34
   00 29 26 00 04 00 00 13 c4 27 92 00 04 0a 0a 01 bf 27
   93 00 04 00 00 10 7c 29 14 00 01 01]

X->H
    [00 07 00 2c 00 00 01 00 10]
```

```
X->H
    [00 11 00 43 00 50 01 00 01 0d 03 00 64 1f 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 50 01]

H->X
    [00 05 00 43 00 50 01]

X->H
    [00 44 00 43 00 51 01 00 01 0d 03 00 64 1f 00 a7 00 20
   01 03 00 33 00 2e 00 01 29 ff 00 28 2a 0e 00 04 0a 0a
   01 ac 2a 01 00 1c 2a 03 00 01 00 2a 07 00 04 00 00 29
   bc 2a 13 00 01 00 2a 02 00 06 2a 08 00 02 00 02]

X->H
    [00 0d 00 2e 00 10 01 00 01 0d 03 00 64 1f 20]

H->X
    [00 05 00 43 00 51 01]

H->X
    [00 05 00 43 00 51 01]

H->X
    [00 32 00 44 00 00 01 00 01 0d 03 00 64 1f 00 a7 00 27
   01 03 00 33 00 1c 00 02 29 1e 00 06 32 30 35 39 37 00
   29 20 00 0c 31 30 2e 31 30 2e 31 2e 32 35 34 00]

H->X
    [00 05 00 2e 00 10 01]

X->H
    [00 07 00 44 00 00 01 00 10]

H->X
    [00 05 00 2e 00 10 01]

X->H
    [00 1e 00 43 00 52 01 00 01 0d 03 00 64 1f 00 a7 00 27
   01 03 00 33 00 08 00 01 29 15 00 02 00 ca]

H->X
    [00 05 00 43 00 52 01]

H->X
    [00 05 00 43 00 52 01]

X->H
```

```
    [00 25 00 43 00 53 01 00 01 0d 03 00 64 1f 00 a7 00 28
   01 03 00 33 00 0f 00 02 29 15 00 02 00 64 29 48 00 03
   02 00 20]

H->X
    [00 05 00 43 00 53 01]

H->X
    [00 05 00 43 00 53 01]

X->H
    [00 37 00 43 00 54 01 00 01 0d 03 00 64 1f 00 a7 00 28
   01 03 00 33 00 21 00 02 29 15 00 02 00 c8 29 48 00 15
   00 00 05 64 65 61 63 74 69 76 61 74 65 64 68 65 61 64
   65 72 00]

H->X
    [00 05 00 43 00 54 01]

H->X
    [00 05 00 43 00 54 01]

H->X
    [00 11 00 08 00 00 01 00 02 0d 03 00 64 1f 0d 03 00 64
   1f]

X->H
    [00 07 00 08 00 00 01 00 10]

X->H
    [00 57 00 69 00 10 01 00 01 0d 03 00 64 1f 02 02 1e 2a
   00 05 01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 3c
   01 11 00 04 00 00 00 3c 01 10 00 04 00 00 25 80 01 12
   00 04 00 00 25 80 03 00 33 00 18 00 03 27 4e 00 02 00
   10 27 92 00 04 0a 0a 01 bf 27 93 00 04 00 00 10 7c]

H->X
    [00 05 00 69 00 10 01]


CSP 2 API Trace:
X->H
    [00 da 00 2d 00 17 02 00 01 0d 03 00 c8 17 00 33 01 03
   00 33 00 c6 00 11 27 4e 00 02 00 05 27 7e 00 03 08 00
   00 29 19 00 06 32 32 32 32 32 00 29 1b 00 0c 31 30 2e
   31 30 2e 31 2e 32 35 32 00 29 1c 00 04 00 00 13 c4 29
   23 00 06 31 31 31 31 31 00 29 25 00 0c 31 30 2e 31 30
   2e 31 2e 32 35 34 00 29 26 00 04 00 00 13 c4 29 28 00
   06 31 31 31 31 31 00 29 2d 00 06 31 31 31 31 31 00 29
   2f 00 0c 31 30 2e 31 30 2e 31 2e 32 35 34 00 29 30 00
   04 00 00 13 c4 29 33 00 01 01 27 18 00 08 02 00 00 00
   05 11 11 10 27 17 00 06 02 00 05 22 22 20 29 ff 00 23
```

```
   2a 0e 00 04 0a 0a 01 bf 2a 01 00 17 2a 03 00 01 00 2a
   07 00 04 00 00 10 7c 2a 02 00 06 2a 08 00 02 00 02 29
   16 00 01 01]

H->X
    [00 0c 00 2d 00 17 02 00 01 0d 03 00 c8 17]

H->X
    [00 0c 00 2d 00 17 02 00 01 0d 03 00 c8 17]

H->X
    [00 11 00 bf 00 00 02 00 02 0d 03 00 c8 17 0d 03 00 c8
     17]

X->H
    [00 07 00 bf 00 00 02 00 10]

H->X
    [00 0d 00 ba 00 00 02 00 01 0d 03 00 c8 17 01]

X->H
    [00 07 00 ba 00 00 02 00 10]

X->H
    [00 32 00 43 00 17 02 00 01 0d 03 00 c8 17 00 a7 00 21
   01 03 00 33 00 1c 00 02 29 19 00 06 32 30 35 39 37 00
   29 1b 00 0c 31 30 2e 31 30 2e 31 2e 32 35 34 00]

H->X
    [00 05 00 43 00 17 02]

H->X
    [00 05 00 43 00 17 02]

H->X
    [00 25 00 44 00 00 02 00 01 0d 03 00 c8 17 00 a7 00 20
   01 03 00 33 00 0f 00 02 29 4b 00 02 00 64 29 48 00 03
   02 00 20]

X->H
    [00 07 00 44 00 00 02 00 10]

H->X
    [00 37 00 44 00 00 02 00 01 0d 03 00 c8 17 00 a7 00 20
   01 03 00 33 00 21 00 02 29 4b 00 02 00 c8 29 48 00 15
   00 00 05 64 65 61 63 74 69 76 61 74 65 64 68 65 61 64
   65 72 00]

X->H
    [00 07 00 44 00 00 02 00 10]
```

```
X->H
    [00 57 00 69 00 17 02 00 01 0d 03 00 c8 17 02 02 1e 2a
   00 05 01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 3b
   01 11 00 04 00 00 00 45 01 10 00 04 00 00 24 e0 01 12
   00 04 00 00 2b 20 03 00 33 00 18 00 03 27 4e 00 02 00
   10 27 92 00 04 0a 0a 01 ac 27 93 00 04 00 00 29 bc]

H->X
    [00 05 00 69 00 17 02]


SIP Trace from CSP1:

1 -SENT To 10.10.1.252:5060 at 6262
INVITE sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254
To: 22222<sip:22222@10.10.1.252:5060>
From: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
Contact: 11111<sip:11111@10.10.1.254:5060>
User-Agent: Excel_CSP/83.10.9
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.254
s=SIP_Call
c=IN IP4 10.10.1.191
t=0 0
m=audio 4220 RTP/AVP 0

2 -RECEIVED From 10.10.1.252:5060 at 6262
SIP/2.0 100 Trying
To: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
From: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.10.9
Content-Length: 0

3 -RECEIVED From 10.10.1.252:5060 at 6263
SIP/2.0 180 Ringing
To: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
From: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
```

```
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.10.9
Content-Length: 0

4 -RECEIVED From 10.10.1.252:5060 at 6263
SIP/2.0 200 OK
To: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
From: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 1 INVITE
Contact: 22222<sip:22222@10.10.1.252:5060>
Require: timer
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.10.9
Content-Type: application/sdp
Content-Length: 131

v=0
o=sip 1151434244 1151434244 IN IP4 10.10.1.252
s=SIP_Call
c=IN IP4 10.10.1.172
t=0 0
m=audio 10684 RTP/AVP 0
a=sendrecv

5 -SENT To 10.10.1.252:5060 at 6263
ACK sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254
To: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
From: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 1 ACK
Content-Length: 0

6 -SENT To 10.10.1.252:5060 at 6263
REFER sip:22222@10.10.1.252:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.254
From: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
To: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 2 REFER
Max-Forwards: 70
Contact: 11111<sip:11111@10.10.1.254:5060>
Refer-To: <sip:20597@10.10.1.254>
User-Agent: Excel_CSP/83.10.9
```

```
Content-Length: 0

7 -RECEIVED From 10.10.1.252:5060 at 6263
SIP/2.0 202 Accepted
To: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
From: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 2 REFER
Contact: 22222<sip:22222@10.10.1.252:5060>
Via: SIP/2.0/UDP 10.10.1.254
User-Agent: Excel_CSP/83.10.9
Content-Length: 0

8 -RECEIVED From 10.10.1.252:5060 at 6264
NOTIFY sip:11111@10.10.1.254:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
From: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 3 NOTIFY
Event: refer
Contact: 22222<sip:22222@10.10.1.252:5060>
Subscription-State: active;expires=32
Content-Type: message/sipfrag;version=2.0
Content-Length: 20

SIP/2.0 100 Trying

9 -SENT To 10.10.1.252:5060 at 6264
SIP/2.0 200 OK
To: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
From: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 3 NOTIFY
Via: SIP/2.0/UDP 10.10.1.252
User-Agent: Excel_CSP/83.10.9
Content-Length: 0

10-RECEIVED From 10.10.1.252:5060 at 6264
NOTIFY sip:11111@10.10.1.254:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.252
To: 11111<sip:11111@10.10.1.254:5060>;tag=412781131876
From: 22222<sip:22222@10.10.1.252:5060>;tag=81131b2a
Call-ID: EXCEL-CSP1.101f.6262.800@10.10.1.254
CSeq: 4 NOTIFY
Event: refer
Contact: 22222<sip:22222@10.10.1.252:5060>
Subscription-State: terminated;retry-
   after=5;reason=deactivatedheader
Content-Type: message/sipfrag;version=2.0
```

Programmable SIP URI Extensions

Overview This features allows host application developers to use the EXS API to write and access user name extensions in the Contact and Request Uniform Resource Identifiers (URIs) in SIP headers.

Description These user name extensions can be used to carry application-specific information such as PSTN trunk group information.

The CSP supports the reporting of Contact URIs and Request URIs as follows:

- The Contact URI user name is reported in TLV 0x292D in the *Request for Service with Data* message.
- The Request URI is reported in TLV 0x2954 in the *Request for Service with Data* message.
- You must first indicate in the SIP Message Information Mask TLV (0x027F) in the *VoIP Protocol Configure* message (0x00EE) that you want the CSP to report the Request URI information (Bit 3) and Contact URI information (Bit 5).

Important! The length of username or password or hostname in TO, FROM and CONTACT URIs in the inbound INVITE (coming from the network to CSP) should not exceed 179 bytes. The length of username or hostname in REQUEST URI in the inbound INVITE (coming from the network to CSP) should not exceed 179 bytes. If it exceeds the call is purged with 0xDD and the Invite is responded with 500 Internal Server error.

The following TLVs are also used to implement this feature. Refer to the *Tag Length Value Blocks* chapter in the *API Reference*.

- **0x2934 SIP Contact URI User Information Qualifier**
  Use this TLV to append a user-defined ASCII string to the Contact URI user name in outbound SIP calls from the CSP. Specifically, the ASCII string is populated in the user information of the Contact-URI header in the outbound SIP INVITE message.

  When the CSP is in a SIP/PSTN gateway environment, you can use this TLV to report the originating trunk group information in a PSTN-to-SIP call.

- **0x2935 SIP Contact URI Parameters**
  Use this TLV to receive and send Contact-URI parameters to and from host applications.

  For inbound calls, this TLV carries all of the Contact URI parameters received by the SIP software in the *Request for Service with Data* message to host applications.

  For outbound SIP calls, this TLV allows you to insert Contact URI parameters in SIP INVITE messages.

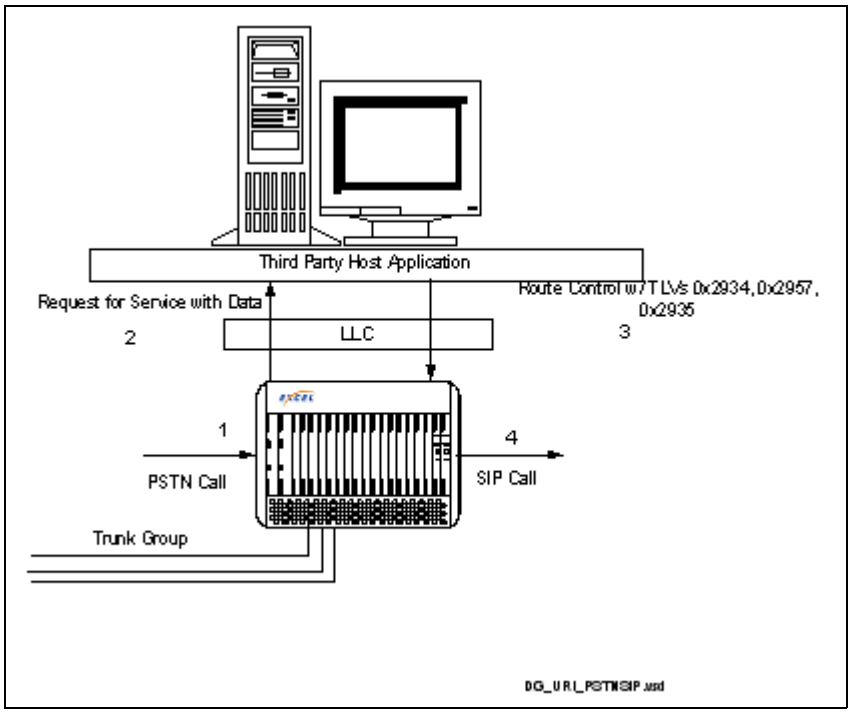
To report this TLV you must first set Bit 5 in the SIP Message Information Mask TLV (0x027F).

Examples in a Gateway Environment

This section shows calls in a network architecture where the host application controls a SIP/PSTN gateway. Each figure in this section is followed by the steps and message traces to further explain the process.

Figure 5-14 Originating and Destination Trunk Groups Reported by the CSP

The following example involves a PSTN-to-SIP call.



Steps

1. A call comes into the CSP from the PSTN for example, in an SS7 IAM message.
2. The CSP sends the *Request for Service with Data* message to the host application.
3. The host application sends the *Route Control* message to the CSP. The message could contain any or all of the following TLVs:
   - 0x2934 SIP Contact URI User Name Qualifier *
   - 0x2935 SIP Contact URI Parameters *
   - 0x2957 SIP Request URI User Name Qualifier

   * One or the other is included but usually not both together.
4. The CSP sends the SIP INVITE to the network with the enhanced URIs.

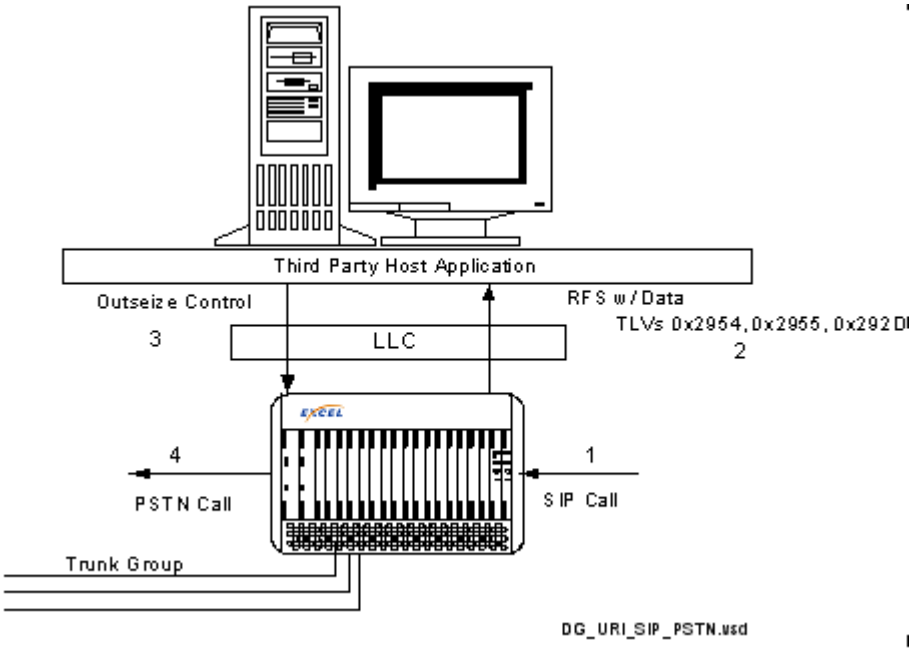
Message Trace

```
H->X

[00 71 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 42 00 04 27 7e 00
     03 08 00 00 27 17 00 05 10 00 04 11 11 29 57 00 13 74
     67 72 70 3d 6c 6f 63 61 6c 3d 74 67 35 34 33 32 31 00
     29 35 00 15 74 67 72 70 3d 22 6c 6f 63 61 6c 3d 74 67
     31 32 33 34 35 22 00]

SIP:
INVITE sip:1111;tgrp=local=tg54321@10.10.1.202:5060 SIP/
   2.0
Via: SIP/2.0/UDP 10.10.1.31
To: 1111<sip:1111@10.10.1.202:5060>
From:
   00000000<sip:00000000@10.10.1.31:5060>;tag=171166949b
Call-ID: EXCEL-CSP1.6af.155.400@10.10.1.31
Contact:
   00000000<sip:00000000;tgrp=local=tg12345@10.10.1.31:50
   60>
User-Agent: Excel_CSP/82.20.99
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 99

v=0
o=sip 0 0 IN IP4 10.10.1.31
```

```
s=SIP_Call
c=IN IP4 10.10.1.37
t=0 0
m=audio 14844 RTP/AVP 0
```

Figure 5-15 TLVs Reported in *Request for Service with Data* message

The following example involves a SIP-to-PSTN call.



Steps

1. A SIP call comes into the CSP in a SIP INVITE message.
2. The CSP sends the host application a *Request for Service with Data* message that could contain any or all of the following TLVs:
   - 0x2954 SIP Request URI User Name
   - 0x2955 SIP Request URI Host Name
   - 0x292D SIP Contact URI User Name
3. The host application sends an *Outseize Control* message to a span/channel within a given destination trunk group.
4. The CSP sends an SS7 IAM to the PSTN network over the specified destination trunk group.

Message Trace

```
X->H
[00 ec 00 2d 00 07 01 00 01 0d 03 00 01 09 00 33 01 03 00 33
     00 d8 00 13 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 05 31 31 31 31 00 29 1b 00 0b 31 30 2e 31 30 2e
     31 2e 33 31 00 29 23 00 05 31 31 31 31 00 29 25 00 0b
     31 30 2e 31 30 2e 31 2e 33 31 00 29 2d 00 18 31 31 31
     31 3b 74 67 72 70 3d 6c 6f 63 61 6c 3d 74 67 35 34 33
     32 31 00 29 2f 00 0c 31 30 2e 31 30 2e 31 2e 32 30 32
     00 29 30 00 04 00 00 13 c4 29 33 00 01 01 27 18 00 07
     02 00 00 00 04 11 11 27 17 00 05 02 00 04 11 11 27 94
     00 04 0a 0a 01 ca 27 95 00 04 00 00 70 16 27 b0 00 02
     00 02 27 b1 00 02 00 01 29 16 00 01 01 29 54 00 18 31
     31 31 31 3b 74 67 72 70 3d 6c 6f 63 61 6c 3d 74 67 31
     32 33 34 35 00 29 55 00 0b 31 30 2e 31 30 2e 31 2e 33
     31 00]

INVITE sip:1111;tgrp=local=tg12345@10.10.1.31 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.202:5060
From: "rj"
   <sip:1111@10.10.1.31>;tag=9421080012ceb664b5d730-
   29784667
To: <sip:1111@10.10.1.31>
Call-ID: 00082194-b6ce08ef-600c8cf5-178f0b21@10.10.1.202
CSeq: 101 INVITE
Expires: 180
User-Agent: Cisco-SIP-IP-Phone/2
Accept: application/sdp
Contact: sip:1111;tgrp=local=tg54321@10.10.1.202:5060
Content-Type: application/sdp
Content-Length: 219

v=0
o=CiscoSystemsSIP-IPPhone-UserAgent 341 9240 IN IP4
   10.10.1.202
s=SIP Call
c=IN IP4 10.10.1.202
t=0 0
m=audio 28694 RTP/AVP 0 8 18 101
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-11
```

Support for Request URI Parameters in SIP INVITE Messages

Overview

This feature allows host application developers to use the EXS API to write and access proprietary Request URI parameters that extend SIP messages for carrying application-specific information.

Description

SIP Request URI Parameter TLV (0x2958) supports this feature for Inbound and Outbound calls, as explained below.

Inbound Calls

The Request URI Parameters TLV (0x2958) carries all the Request URI parameters, received by the CSP SIP stack in the *Request for Service with Data* message (0x002D) to the host application.

To report this TLV to the host, you must program Bit 6 in the SIP Message Information Mask TLV (0x027F) used in the *VoIP Protocol Configure* message (0x00EE).

Example of Inbound Call

In the following example, the italic text in the SIP message represents the URI parameters contained in the *Request for Service with Data* message (0x002D).

```
X->H
    [01 2a 00 2d 00 df 04 00 01 0d 03 03 fe 1d 00 33 01 03 00 33
     01 16 00 18 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 05 31 31 31 31 00 29 1b 00 0c 31 30 2e 31 2e 32
     30 35 2e 32 35 00 29 1c 00 04 00 00 13 c4 29 23 00 05
     32 32 32 32 00 29 25 00 0c 31 30 2e 31 2e 32 30 35 2e
     31 35 00 29 26 00 04 00 00 13 c4 29 2d 00 1a 32 32 32
     32 3b 74 67 72 70 3d 22 6c 6f 63 61 6c 3d 74 67 31 32
     33 34 35 22 00 29 2f 00 0c 31 30 2e 31 2e 32 30 35 2e
     31 35 00 29 30 00 04 00 00 13 c4 29 33 00 01 01 27 18
     00 07 02 00 00 00 04 22 22 27 17 00 05 02 00 04 11 11
     27 94 00 04 0a 01 cd 4f 27 95 00 04 00 00 11 80 27 b0
     00 02 00 02 27 b1 00 02 00 04 29 16 00 01 01 29 54 00
     18 31 31 31 31 3b 74 67 72 70 3d 6c 6f 63 61 6c 3d 74
     67 35 34 33 32 31 00 29 55 00 0c 31 30 2e 31 2e 32 30
     35 2e 32 35 00 29 56 00 04 00 00 13 c4 29 35 00 17 22
     6c 75 63 65 6e 74 3d 65 78 63 65 6c 3d 74 65 6c 65 63
     6f 6d 22 00 29 53 00 02 01 00]

X->H
    [00 45 00 43 00 9e 04 00 01 0d 03 03 fe 1d 00 a7 00 23 01 03
     00 33 00 2f 00 02 29 53 00 02 02 01 29 58 00 23 74 67
     72 70 3d 67 6c 6f 61 62 61 6c 3d 74 61 67 6c 75 63 65
     6e 74 65 78 63 65 6c 73 69 70 35 30 30 30 00]

Received message:
INVITE sip:1111;tgrp=local=tg54321@10.1.205.25:5060;
   tgrp=gloabal=taglucentexcelsip5000 SIP/2.0
```

```
Via: SIP/2.0/UDP 10.1.205.15
To: 1111<sip:1111@10.1.205.25:5060>
From: 2222<sip:2222@10.1.205.15:5060>;tag=204846324c0
Call-ID: EXCEL-CSP2.800.1216.60@10.1.205.15
Contact:
   2222<sip:2222;tgrp="local=tg12345"@10.1.205.15:5060;"l
   ucent=excel=telecom">
User-Agent: Excel_CSP/82.30.70
Supported: timer
Session-Expires: 65535
Min-SE: 65534
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 137

v=0
o=sip 0 0 IN IP4 10.1.205.15
s=SIP_Call
c=IN IP4 10.1.205.79
t=0 0
m=audio 4480 RTP/AVP 0 96
a=rtpmap:96 telephone-event/8000
```

Outbound Calls

Host application developers can use the Request URI Parameters 0x2958 TLV in the *Route Control* and *Outseize Control* messages to insert Request URI parameters in SIP INVITE messages.

Example of Outbound Call

In the following example, the italic text in the SIP message represents the URI parameters that the host application inserted using the *Route Control* message (0x00E8).

```
H->X
    [00 71 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 42 00 04 27 7e 00
     03 08 00 00 27 17 00 05 10 00 04 11 11 29 58 00 13 74
     67 72 70 3d 6c 6f 63 61 6c 3d 74 67 35 34 33 32 31 00
     29 35 00 15 74 67 72 70 3d 22 6c 6f 63 61 6c 3d 74 67
      31 32 33 34 35 22 00]

SIP:

INVITE sip:1111@10.10.1.202:5060;tgrp=local=tg54321 SIP/
   2.0
Via: SIP/2.0/UDP 10.10.1.31
To: 1111<sip:1111@10.10.1.202:5060>
From:
   00000000<sip:00000000@10.10.1.31:5060>;tag=17197953780
Call-ID: EXCEL-CSP1.6b7.1920.140@10.10.1.31
```

```
Contact:
   00000000<sip:00000000@10.10.1.31:5060;tgrp="local=tg12
   345">
User-Agent: Excel_CSP/82.20.114
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 99

v=0
o=sip 0 0 IN IP4 10.10.1.31
s=SIP_Call
c=IN IP4 10.10.1.37
t=0 0
m=audio 14876 RTP/AVP 0
```

Support SIP Max Forward in INVITE Message

Overview

This feature allows the CSP to insert the Max-Forwards header in outbound requests from the CSP SIP stack.

The Max-Forwards header is optional and is disabled by default. The host can also configure the Max-Forwards value.

Specification Built From

RFC 3261

Description

Max-Forwards header limits the number of hops a request can make on the way to its destination. It consists of an integer that is decremented by one at each hop. If the Max-Forwards value reaches 0 before the request reaches its destination, it is rejected with a 483 (Too Many Hops) error response.

For example, this feature is needed by customer applications that require the Max-Forwards header in the inbound INVITE message so the application can respond to it. The host inserts this header in the INVITE message generated by the CSP SIP stack.

This feature applies to the following requests generated by the CSP SIP stack:

- INVITE
- Re-INVITE
- ACK
- BYE
- NOTIFY
- REFER
- INFO
- SUBSCRIBE
- CANCEL
- PRACK

Example: Below is an example of the INVITE message with this feature enabled:

```
INVITE sip:35751@10.10.1.19:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.135
To: 35751<sip:35751@10.10.1.19:5060>
From:
   00000000<sip:00000000@10.10.1.135:5060>;tag=734407b2
```

```
Call-ID: EXCEL-CSP16.7.1970.350@10.10.1.135
Contact: 00000000<sip:00000000@10.10.1.135:5060>
User-Agent: Excel_CSP/84.10.1
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Max-Forwards: 80
Content-Type: application/sdp
Content-Length: 100

v=0
o=sip 0 0 IN IP4 10.10.1.135
s=SIP_Call
c=IN IP4 10.10.1.206
t=0 0
m=audio 8028 RTP/AVP 0
```

API Configuring and Querying

To enable this feature using the EXS API, use the SIP Max Forwards (0x0286) TLV in the *VoIP Protocol Configure* (0x00EE) message. Refer to 0x2946 - SIP Request URI Password.

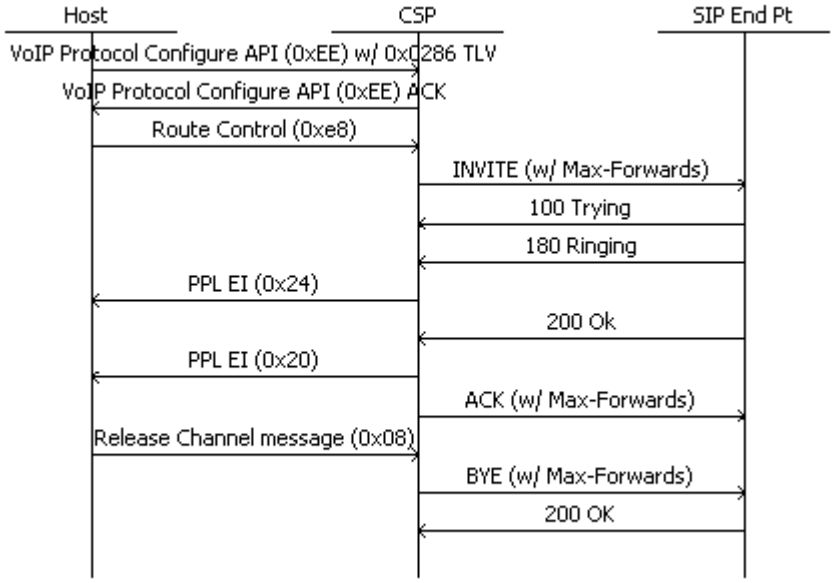
You can query this feature with the *VoIP Protocol Query* (0x00EF) message.

CSA Configuring and Querying

Follow the steps below to configure or query this feature from SwitchKit CSA.

1. To open the CSP SIP Configuration dialog box, do one of the following:
   - Select the CSP Matrix Series 3 Card in the node view**.** Go to the **Configuration** menu and select **SIP→SIP Configuration**.
   - Right-click the node view window (outside the card slots) and select **SIP Configuration** from the menu list.
2. Click the **Show Advanced** button. The **SIP Configuration, Advanced** dialog box opens.
3. Select **SIP MAX Forwards.**

Call Flow


Host
CSP
SIP End Pt
VoIP Protocol Configure API (0xEE) w/ 0xC286 TLV
VoIP Protocol Configure API (0xEE) ACK
Route Control (0xe8)
INVITE (w/ Max-Forwards)
100 Trying
180 Ringing
PPL EI (0x24)
200 Ok
PPL EI (0x20)
ACK (w/ Max-Forwards)
Release Channel message (0x08)
BYE (w/ Max-Forwards)
200 OK

PRACK Support

Overview

The CSP supports four provisional responses:

- 100 Trying
- 180 Ringing
- 182 Queued Message
- 183 Session Progress

You can enable the CSP to support the Provisional Response ACK (PRACK) method to send non 100 provisional responses more reliably over User Datagram Protocol (UDP). This feature is disabled by default.

Pertinent Specification

RFC 3262

Description

The following are the details of this feature:

- The Offer/Answer model for PRACK (section 5 of RFC 3262) is not supported.
- The Support/Require mode for PRACK is at the stack level and not on a call-by-call basis.
- The CSP now supports the Provisional Response method to send non 100 provisional over User Datagram Protocol (UDP)
- There is not indication about the presence/absence of "100rel" tag in the “Supported/Require” header to the host in the RFS, PPL EI for 180Ringing or PPL EI for 183 Session progress for the following respectively:
  - Received INVITE
  - 180 Ringing
  - 183 Session Progress
- The RFC 3262 states the following: “The UAS sending the response reliably should send provisional responses once every two and a half minutes.” The SIP stack supports this standard for the 182 Queued message.

Enabling the CSP to Report Supported and Required Header Fields

The NPDI SIP Extensions (0x294A) TLV indicates to the host the option tags present in the Supported and Required headers in the inbound INVITE, 180 Ringing, and 183 Session Progress within the *Request for Service* or *PPL Event Indication* as follows:

- 180 Ringing and 183 Session Progress - reporting in PPL Event Indication
- INVITE - reporting in Request for Service and possibly PPL Event Indication depending on the length

API Messages Used

The following messages are used by this feature. Refer to the *API Reference* for the formats.

- *Request for Service*
- *PPL Event Indication*

Configuring

Reporting SIP Supported and Required Option Tags

Prior to this feature the CSP already supported Timer option tag and with this feature CSP also supports 100rel (PRACK) tag.

The reporting is disabled by default. To enable the reporting, set bit 14 of the data part in the SIP Message Information Mask (0x027F) TLV.

PRACK Mode

The functionality in the CSP SIP stack will operate in three modes:

- Disabled (Default)
- Support
- Require

The 100rel is a SIP option tag used to indicate support for reliability of provisional responses. The CSP supports 100rel in the following headers:

- Supported
- Required

You use Reliable Provisional Response Mode (0x011B) TLV with the *VoIP Protocol Configure* message. The configuration is at the stack level.

Refer to *SIP Stack in PRACK Disabled Mode (5-277)* and *SIP Stack in PRACK Required Mode (5-279)* for a detailed explanation of the behavior of the CSP SIP stack in support and require modes.

Querying

You can query the current configuration with the *VoIP Protocol Query* (0x00EF) message by including the 0x011B TLV.

CSP to Report Inbound PRACK

The receipt of PRACK is reported to the host using the PPL Event Indication message with the event type (0x002A). Reporting is enabled by default.

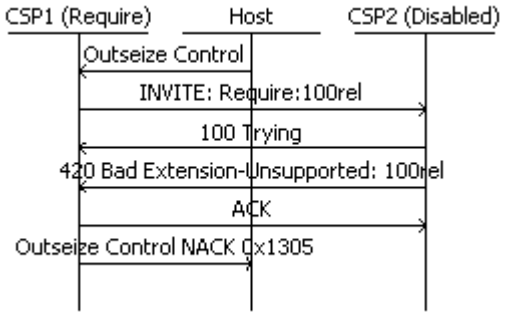
The host controls the reporting by toggling bit 5 of the data part of PPL Event Notification Mask (0x0282) TLV.

Call Flows

SIP Stack in PRACK Disabled Mode

The CSP SIP works in disabled mode - where it does not support PRACK. This mode accommodates backward compatibility.

In the call flow below, the CSP1 sends INVITE with “Require: 100rel” to CSP1 in disabled mode. CSP2 shall respond with 420 Bad Extension containing a header “Unsupported: 100reli.”

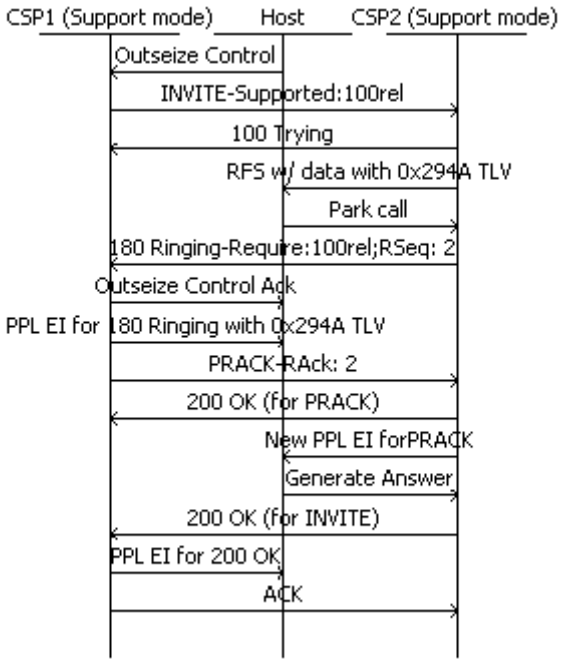


SIP Stack in PRACK Supported Mode

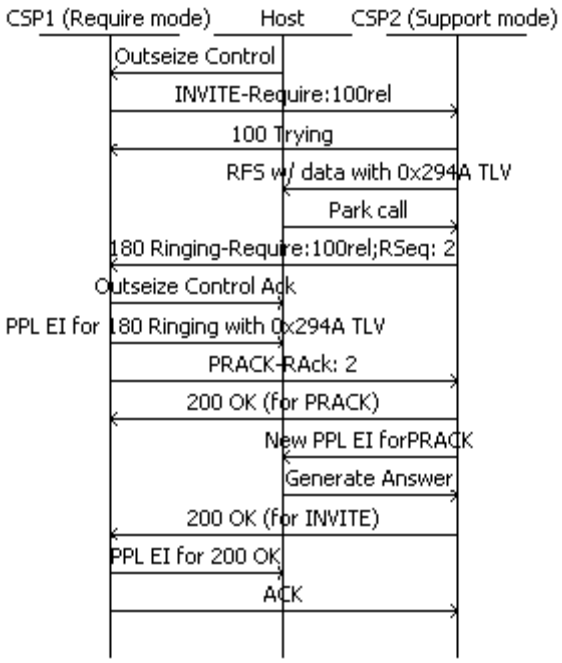
The call flow below assumes that the reporting of option tags in Supported and Require header are enabled.

The following are the behaviors of CSP SIP stack (CSP2) in supported mode:

- For an inbound call (INVITE) from UAC that supports non-100 provisional response being sent reliably (if there is a ‘Supported’ header with ‘100rel’ tag), the CSP accepts the call and sends non-100 provisional response(s), if any, reliably.



- For an inbound call (INVITE) from UAC that requires non-100 provisional response to be sent reliably (if there is a ‘Require’ header with ‘100rel’ tag), the CSP accepts the call and sends all non-100 provisional response(s), if any, reliably.



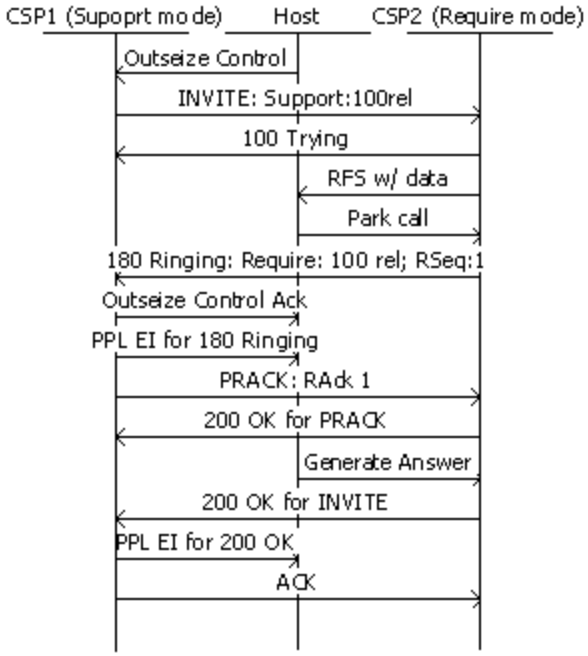
- For an inbound call (INVITE) from UAC that does not support non-100 provisional response being sent reliably (if there is neither ‘Supported’ nor ‘Require’ header with ‘100rel’ tag), then the CSP will accept the call but does not send non-100 provisional response(s), if any, reliably.
- All outbound calls (INVITE) from the CSP will contain ‘Supported’ header with ‘100rel’ tag. Refer to CSP1 in first call flow above.

SIP Stack in PRACK Required Mode

This section provides examples of the CSP SIP stack behavior in require mode.

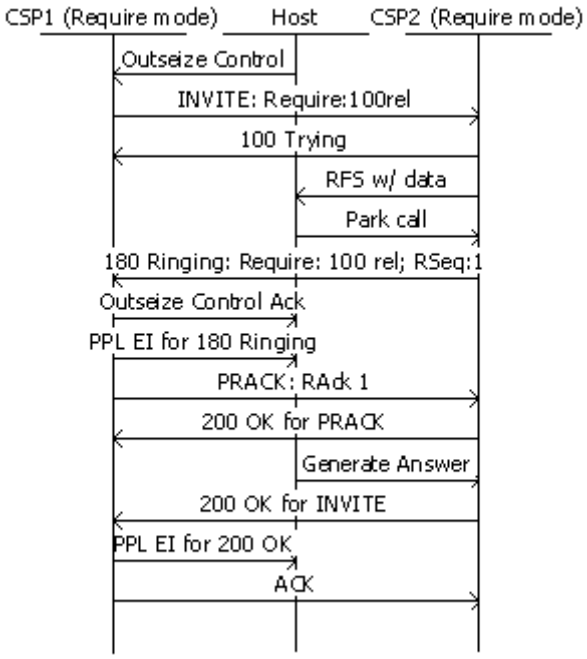
The CSP accepts the call and send all non-100 provisional response(s), if any, reliably:

- Inbound call (INVITE) from UAC that supports non-100 provisional response sent reliably - if there is a “Supported” header with 100rel tag.


CSP1 (Supoprt mode)
Host
CSP2 (Require mode)
Outseize Control
INVITE: Support:100rel
100 Trying
RFS w/ data
Park call
180 Ringing: Require: 100 rel; RSeq:1
Outseize Control Ack
PPL EI for 180 Ringing
PRACK: RAck 1
200 OK for PRACK
Generate Answer
200 OK for INVITE
PPL EI for 200 OK
ACK

The CSP accepts the following call and sends all non-100 provisional response(s), if any, reliably:

- Inbound call (INVITE) from UAC that requires non-100 provisional response to be sent reliably - if there is a “Require” header with 100rel tag.

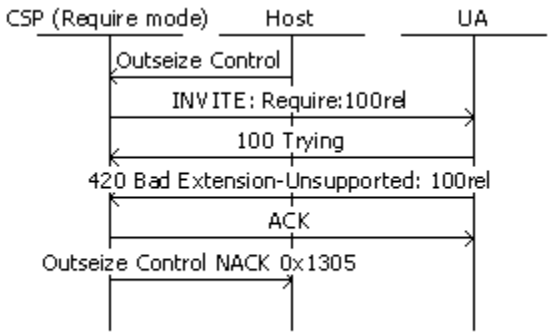


The CSP accepts the following call but does not send non-100 provisional response(s), if any, reliably:

- Inbound call (INVITE) from UAC that does not support non-100 provisional response being sent reliably - if there is neither “Supported” nor “Required” header with 1000rel tag.

All outbound calls (INVITE) from the CSP have “Required” header with “100rel” tag. See call flow above.

In case the CSP is configured for the PRACK method in require mode and if a call is made to a remote UA that does not support 100rel tag, the UA is expected to respond with 420 Bad Extension message.



If a PRACK request received by the CSP does not match any unacknowledged reliable provisional response, the CSP responds to the PRACK with a 481 response.

If a reliable provisional response is retransmitted for 64*T1 (T1=500ms) seconds without reception of a corresponding PRACK, the CSP rejects the original INVITE request with a 500 internal server response.

Support for SIP INFO Message

Overview

Prior to this feature the CSP SIP stack could not generate an INFO message nor could it report an inbound INFO message.

With this feature, the SIP stack can generate the INFO message by the host and report the receipt of an INFO message to the host.

The INFO message also supports the Subject header with read and write access.

Pertinent Specification

RFC 2976

Description

The INFO method carries the session related control information that is generated during a session. ISUP and ISDN signaling messages used to control telephony call services are examples of session control information.

By default, inbound SIP INFO messages are not reported to the host.

This feature is supported by call agent and non-call agent calls.

Reporting the Inbound INFO Message

When the CSP receives an INFO message, the CSP sends the host a *PPL Event Indication* message. This message contains the Subject Header TLV 0x295B if the inbound INFO message has a Subject header and its length is less than 250 bytes.

The PPL Event 0x002C in SIP component 0x00A7 reports the message.

The following two TLVs are supported in this Event Indication:

- Subject Header (0x295B)
- Message Body (0x295C)
- Content Type (0x295D)

The *PPL Event Indication* message contains the Subject Header TLV 0x295B if the Subject header is present in the inbound INFO message.

The *PPL Event Indication* message contains the Message Body TLV 0x295C if the inbound INFO has a message body and its length is less than 500 bytes.

If the SIP stack receives an INFO message for any call in a non-answered state, the CSP responds with error response 488 - Not Acceptable Here.

Generating the Outbound INFO Message

The CSP SIP stack allows the host to generate an outbound INFO message using the PPL Event Request (0x0025) in SIP UA component (0x00A7).

The following three TLVs are supported in this Event Request:

- Subject Header (0x295B)
- Message Body (0x295C)
- Content Type (0x295D)

These TLVs are optional - there are no mandatory TLVs.

Since the INFO method is used for communicating mid-session signaling information along the signaling path for the call, the CSP accepts this Event Request only when the call is in the answered state. In other states, the Event Request is NACKed with the value 0x1303 (Invalid PPL Event) or 0x1308 (Invalid State).

Reporting the INFO Message Response

If a User Agent Server receives an INFO request it must send a final response. The response to the INFO message is reported to the host in a PPL Event Indication message.

The PPL Event 0x002D for SIP component 0x00A7 reports the message.

The PPL Event Indication message contains the SIP Response Code TLV 0x2915. If no response is received from the endpoint, the INFO message is not re-transmitted and the CSP does not report and type of response.

API Messages Used

- *PPL Event Indication (0x0043)*
- *PPL Event Request (0x0044)*

PPL Information

PPL Component 0x00A7:

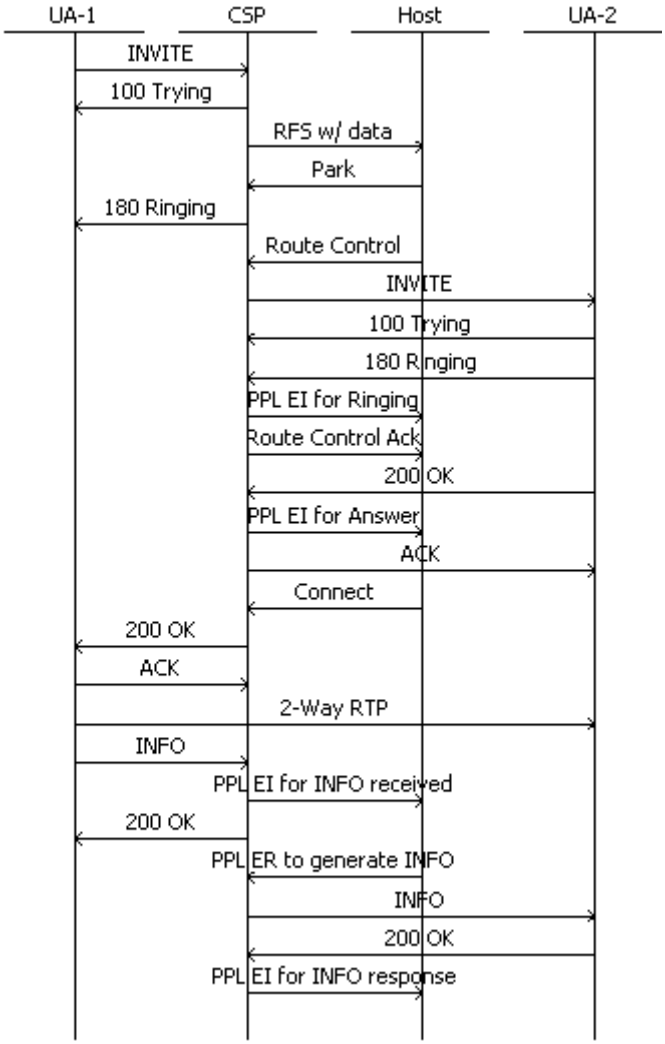
- 0x0025 - Generate INFO Request
- 0x002D - Response for INFO Send Indication
- 0x002C - Report INFO Received

Configuring

By default, inbound SIP INFO messages are not reported to the host. Set bit 6 in the PPL Event Notification Mask TLV (0x0282) in the *VoIP Protocol Configure* message to enable this feature.

Call Flow

The following call flow shows messages involved with the Reporting the Inbound INFO message.



SIP Tunneling

Feature Overview

SIP tunneling is a mechanism that transports any kind of data using SIP signaling messages.

The CSP SIP stack now supports tunneling with the host playing an important role.

The SIP stack acts as a “black box” and the host determines what data to tunnel and what format to transport.

Refer to *Description (5-90)* for a detailed explanation of this feature.

Pertinent Specifications

- RFC 3398 - ISUP to SIP Mapping
- RFC 3261 - SIP
- RFC 3666 - SIP PSTN Call Flows
- RFC 2046 - MIME Part Two: Media Types

Benefits to Customer

Transporting PSTN signaling through IP is one use case of SIP Tunneling.

The following process tunnels SS7 messages through IP.

Process

1. The CSP reports the IAM from the network to the host as SS7 ICBs within the *Request for Service with Data* (0x002D) message.
2. The received ICBs are provided to the SIP stack using the *Outseize Control* or *Route Control* API messages and Subsequent PPL Event Request (0x023). The message is embedded within the SIP INVITE message as MIME data and sent through the IP network.
3. The CSP UAC receiving the SIP INVITE message reports the MIME data to the host with the RFS.
4. The host can use this SS7 ICB received from the INVITE message to make an outbound SS7 call with appropriate parameters.

Similarly for other SS7 messages, a SIP message is used to tunnel it through the IP network.

There is not hardwired nor one-to-one mapping of SS7 messages to SIP message. Refer to *Recommended mapping between ISUP messages and SIP messages (5-297)*.

API and TLVs

Messages

- *Outseize Control* (0x002C)
- *Request for Service* (0x002D)
- *Route Control* (0x00E8)
- *PPL Event Request* (0x0044)
- *PPL Event Indication* (0x0043)
- *Release Channel with Data* (0x36)
- *Connect with Data* (0x05)
- *Channel Released with Data* (0x69)

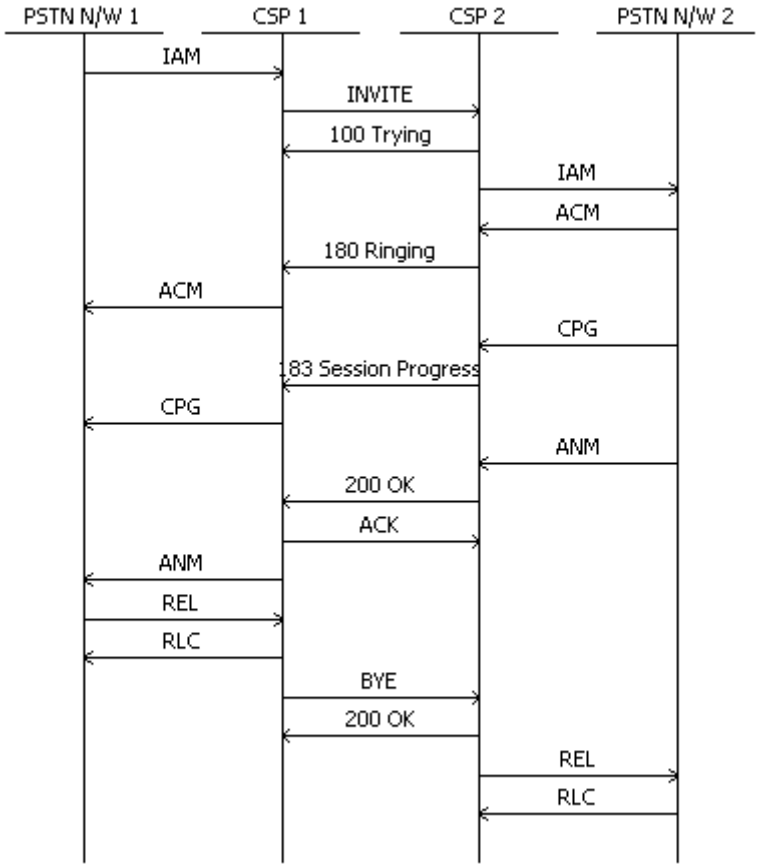
TLVs

- 0x2964 SIP MIME Information
- Call Flow

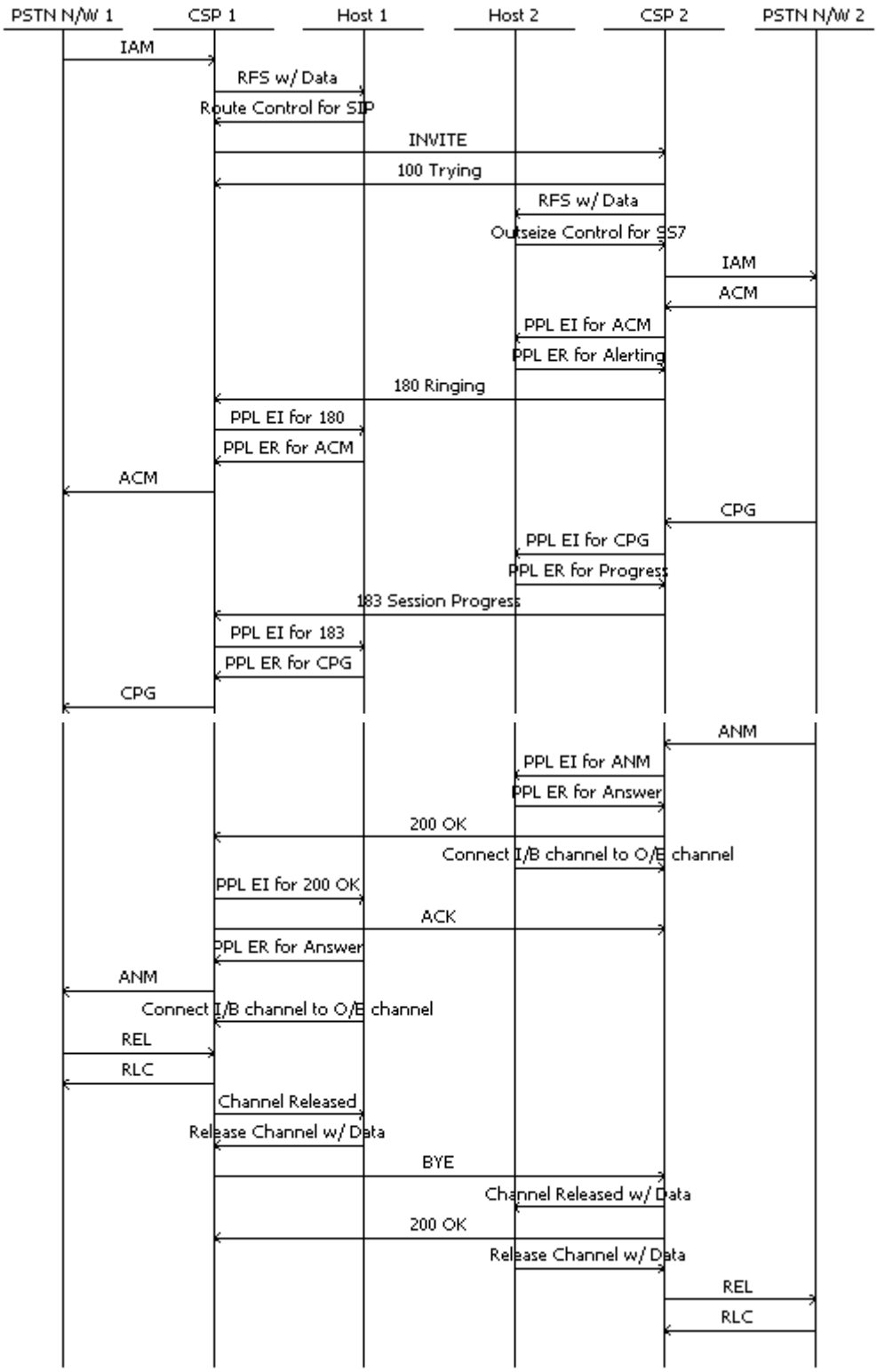
PPL Information

- PPL Event Request 0x0022 in PPL Component 0x00A7 generates 181 Call is Being Forwarded
- PPL Event Indication in PPL Component 0x00A7 reports the receipt of 181 response.

Call Flows **High-level call flow for ISUP tunneled through SIP**



SS7 Tunneled Through SIP with Host Interaction

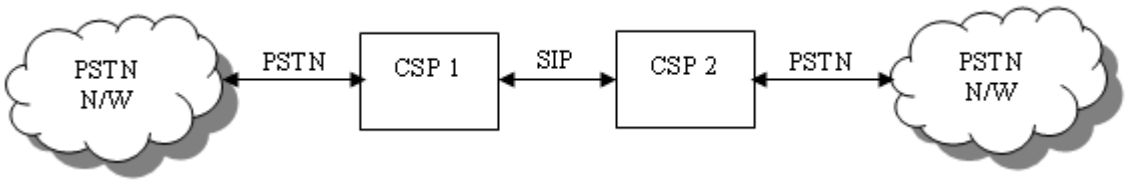


Description The data tunneled through SIP is CSP specific data, for example SS7 ICBs.

> **Important!** This feature is valid only if the end-to-end SIP communication is between **two CSPs**.

Figure 5-16 Network Diagram

Note the following regarding this feature:



- The host application must make sure the variant of inbound PSTN stack matches with the variant on the terminating side or at least is compatible.
- The CSP provides provisions to the host to insert the variant type in SIP messages but the CSP cannot validate it.
- The SIP tunneling capability in the CSP cannot always make the CSP interoperable with third party systems using the SIP interface. The data must be tunneled in a universally accepted format or at least understood by that specific third party system that needs to interwork with the CSP.

Support INFO Message in Call Setup Phase

The CSP supports the INFO message (both generating and receiving) during any phase of the call (setup and/or active) as long as the dialog is established. A dialog is established once the UAS responds to the INVITE with any provisional or 2xx response with a tag value in the To URI.

The 100 Trying response for the INVITE message might not contain a To URI tag value if it is generated from a proxy.

The CSP always inserts the To URI tag value in the 100 Trying response to the INVITE so a dialog gets established in the UAC as soon as it receives 100 Trying.

Support 181 Call is Being Forwarded Response

The following PPL Event Request and PPL Event Indication support the generation and reporting the receipt of 181 provisional response respectively.

- PPL Event Request 0x0022 in SIP UA Component 0x00A7 - allow generation of 181 response.
- PPL Event Indication 0x0030 in SIP component 0x00A7 reports the receipt of 181 response.

Support MIME in SIP Messages

MIME data is a bundle of message body and headers defining the message body. MIME is a general term referring to any data in the message body other than the SDP. MIME supports the following SIP messages in addition to the SDP:

- 180 Ringing
- 181 Call is being forwarded
- 182 Queued
- 183 Session Progress (Supports SDP and MIME)
- 200 OK for INVITE (Supports SDP and MIME)
- BYE
- Cancel
- INFO

If the NPDI data is to be reported to the host (within the PPL Event Indication) for the receipt of the following is too big to fit into a single API message, then it is reported to the host as Subsequent PPL Event Indication (Event: 0x0023 Comp: 0x00A7).

- 180 Ringing
- 181 Call is being forwarded
- 182 Queued
- 183 Session Progress
- 200 OK for INVITE

820 bytes is the maximum NPDI data length that can be reported to the host in a single PPL Event Indication. The fragmentation of large NPDI data is not supported when reporting INFO or BYE message.

TLVs to Generate and Report SIP Messages

The API messages used to generate and report these SIP message support the following TLVs within the NPDI Universal ICB (0x0033). Refer to *Mapping SIP Messages to API Messages (5-298)*.

- *0x2964 SIP MIME Information (5-299)* - This optional TLV uses the following nested TLVs within itself to define the MIME headers and MIME body.
    - SIP Content Type TLV - 0x295D
    - SIP Content Encoding TLV - 0x2965
    - SIP Content Disposition TLV - 0x2966
    - SIP Content Language TLV - 0x2967
    - SIP MIME Message Body TLV - 0x2968 - Mandatory TLV for MIME encapsulation. If the message body TLV is absent then the SIP message does not contain any MIME data. Per RFC3261, if the message body is present in a SIP message, then the Content-Type header must be present. For this reason the CSP SIP stack inserts a default value of application/custom if the host provides the Message body TLV but not the Content Type TLV.

The host must ensure that the two SS7 stacks involved in the call are compatible with each other. Because of this, the host can define a token for representing the version of the SS7 stack. This token for the version of the SS7 stack, receiving the inbound call can be communicated to the SIP UAS by inserting it in the Content Type header of the outgoing INVITE message. This version can be used to check for compatibility at the SIP receiving end.

For example the Content-Type header looks like:

Content-Type: application/ISUP; version=ANSI_SS7

Configuring and Querying with API

Configuring Reporting of Receipt of 181 Call

By default, the CSP does not report the receipt of *181 Call is Being Forwarded* using the *PPL Event Indication* (0x0043) message. To enable it, set bit 9 in the *Call Flow (5-285)* TLV and send it in *VoIP Configuration* message (0x00EE).

Configuring Sending MIME

You can configure the CSP to send MIME in outgoing SIP messages two ways. Use the first option to control the tunnel type on a per call basis and the second option to configure at the stack level.

- Set the SIP Tunnel Type TLV (0x2936) to value 0x0004 (Custom MIME Body) in the respective messages:
  - *PPL Event Request (0x0044)*
  - *Outseize Control* (0x002C)
  - *Route Control* (0x00E8)
  - *Connect with Data* (0x05)
  - *Release Channel with Data* (0x36)
- Set the SIP Tunnel Type TLV (0x0270) to value 0x0004 (Custom MIME Body) in the *VoIP Protocol* message (0x00EE).

Configuring Reporting Income MIME

By default, the CSP does not report the incoming MIME. To enable it, set bit 20 of the SIP Message Information Mask TLV (0x027F) and send it in the *VoIP Protocol Configure* (0x00EE) message.

Querying

Use the *VoIP Protocol Query* (0x00EF) message to query the following:

- SIP PPL Event Notification Mask value
- Message Information Mask value
- SIP Tunnel Type

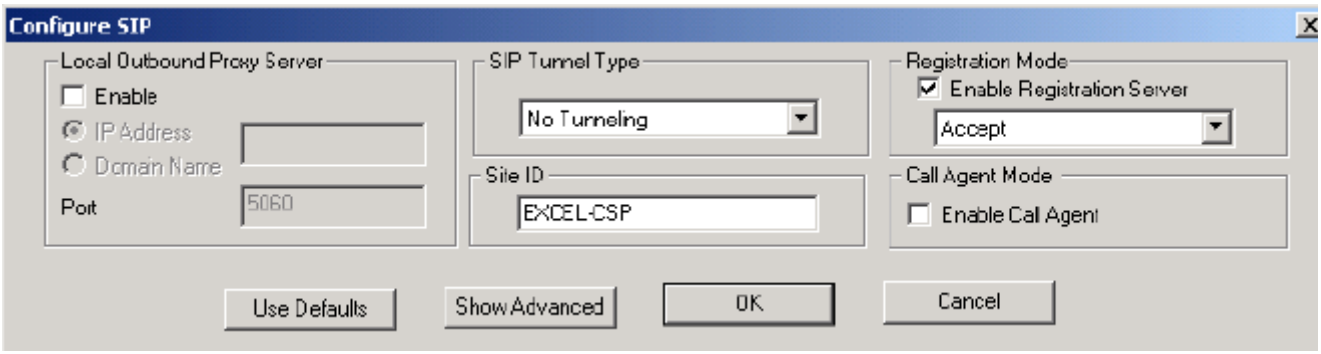
Configuring with CSA

Configuring Reporting of Receipt of 181 Call

1 To open the CSP SIP Configuration dialog box, do one of the following:

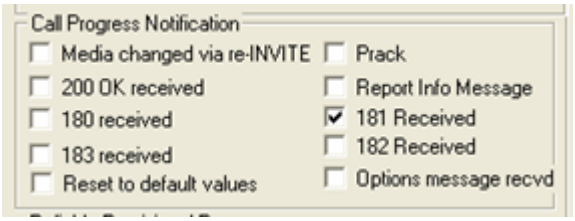
- Select the CSP Matrix Series 3 Card in the node view. Go to the **Configuration** menu and select **SIP> SIP Configuration.**
- Right-click the node view window (outside the card slots) and select **SIP Configuration** from the menu list.

The dialog box opens showing only the common settings and default values.



2 Click the **Show Advanced** button. The **SIP Configuration, Advanced** dialog box opens.

3 Under **Call Progress Notification** are the following options:

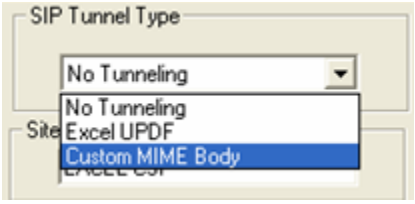


4 Select **181 Received**.

Configuring Reporting Income MIME

1 Follow steps 1-2 above to access the **SIP Configuration, Advanced** dialog box.

2 Under **SIP Tunnel Type** the following options appear.



3 Select **Customer MIME Body**.

Mapping The following tables provide the mapping supported by this feature.

Table 5-5 Recommended mapping between ISUP messages and SIP messages

ISUP Message	SIP Message
Initial address message (IAM)	INVITE
Address Complete Message (ACM)	180 Ringing
Call Progress Message (CPG) before receiving ANM	180 Ringing 181 Call Is Being Forwarded/183 Session Progress
Alerting	180 Ringing
Progress	183 Session Progress
In-band information	183 Session Progress
Call forward; line busy	181 Call is being forwarded
Call forward; no reply	181 Call is being forwarded
Call forward; unconditional	181 Call is being forwarded
Answer message (ANM)	200 OK for INVITE
Connect message (CON)	200 OK for INVITE
Call progress message (CPG) after having received ANM	INFO
Information Request Message (INR)	INFO
Information Message (INF)	INFO
Release message (REL)	BYE/CANCEL
Release complete message (RLC)	N.A.

Table 5-6 Mapping SIP Messages to API Messages

SIP Message	API (Requests)	API (Indications)
180 Ringing	PPL Event Request - Event 0x00CA L4 CH Component (0x0061)	PPL Event Indication - Event 0x0024 SIP Component (0x00A7)
181 Call is being forwarded	PPL Event Request - Event 0x0022 SIP Component (0x00A7)	PPL Event Indication - Event 0x0030 SIP Comp (0x00A7)
182 Queued	PPL Event Request - Event 0x0026 SIP Component (0x00A7)	PPL Event 0x002E SIP Component (0x00A7)
183 Session Progress	PPL Event Request - Event 0x00CB L4 CH Component (0x0061) PPL Event Request - Event 0x001F SIP UA Comp (0x00A7)	PPL Event Indication - Event 0x001F SIP Comp (0x00A7)
200 OK for INVITE	PPL Event Request - Event 0x00C9 L4 CH Comp (0x0061)	PPL Event Indication - Event 0x0020 SIP Component (0x00A7)
INFO	PPL Event Request - Event 0x0025 SIP Component (0x00A7)	PPL Event Indication - Event 0x002C SIP Component (0x00A7)
BYE/Cancel	Release with Data 0x0036	Channel Released with Data (0x0069)

TLVs 0x2964 SIP MIME Information

The following new TLV supports this feature.

Used in:
0x0033 NPDI Universal ICB in:
Route Control message
Outseize Control message
Release Channel with Data
Connect with Data
Channel Released with Data
Request for Service with Data message
PPL Event Request message
PPL Event Indication message

Byte	Description	
0, 1	Tag	0x2964
2, 3	Length	Variable (Maximum of 780 bytes)

0x0282 PPL Event Notification Mask

Use the following TLV to configure this feature.

Used in:
VoIP Protocol Configure message
VoIP Protocol Query message

Byte	Description	
0, 1	Tag	0x0282
2, 3	Length	0x0004
4-7	Value[0-3]	32 bit mask with each bit selects specific PPL Event Indication. Bit 9 - Report receipt of 181 call

SIP Support for MIME

Overview

Currently, the CSP Session Initiation Protocol (SIP) does not support Multipurpose Internet Mail Extensions (MIME) in outgoing messages. This feature allows a user to send proprietary MIME or standard MIME such as ISUP in outgoing messages. MIME data in incoming messages will be reported to the host.

Currently, MIME is supported by the SIP INVITE message. If the incoming INVITE message does not have SDP, then the CSP SIP stack assumes it is a delayed media scenario.

Pertinent Specifications

RFC 2046, RFC 3261

Description

The sending and reporting of MIME data requires the use of the new nested the new *0x2964 SIP MIME Information* TLV to enhance the CSP SIP stack to send proprietary MIME in outgoing messages and report MIME data in incoming messages to the host.

This nested TLV contains the MIME header TLVs and MIME message body TLV and are listed below.

Header TLVs

- *0x295D Content Type* (mandatory)
- *0x2965 SIP Content Encoding* (optional, new)
- *0x2966 SIP Content Disposition* (optional, new)
- *0x2967 SIP Content Language* (optional, new)

Message Body TLV

- *0x2968 SIP MIME Message Body* (mandatory, new)

The SIP MIME Message Body (0x2968) TLV is mandatory data. The Message Body data has to be provided by the host. If Content Type, mandatory header, is not provided by the host in the SIP Content Type (0x295D) TLV, then a default value of "application/custom" is inserted. All of the other headers listed above are optional and are populated only if provided by the host.

MIME will be included in the outgoing INVITE message only if data is available for tunneling. Otherwise, the other host provided TLVs (MIME headers) will be ignored.

Read and write access will be given to only one level of MIME (excluding the SDP). The message body will contain the SDP and another set of tunneled data.

MIME data in incoming INVITE messages is reported to the host in the *PPL Event Indication* (0x0043) and *Request For Service with Data* (0x002D) messages. The contents of the MIME (headers and message) are reported in associated TLVs without parsing. It is up to the host to interpret the MIME header contents and message body.

The MIME data can be tunneled in the outgoing INVITE message. The SIP Tunnel Type TLV (0x2936) is modified to include another type of tunneling which is the 0x0004 Customize Tunnel Type. It can be used to tunnel proprietary data within CSP SIP stack. For example, the contents of the data TLVs will not be parsed, but will be used in the outgoing message.

Refer to the detailed information on modified and new TLVs below.

Message Byte Length

The individual TLVs listed above do not have any individual limits on the message byte length, but collectively within the SIP MIME Information (0x2964) TLV the length must not exceed 780 bytes. A maximum of 780 bytes of MIME data (including headers, header content and the actual tunneled data) can be reported to the host. Any data beyond that will not be reported.

The CSP SIP stack allows NPDI data to be sent in a *Route Control* or *Outseize Control* message and followed by *PPL Event Request* message for subsequent data in order to support a call request. Sending NPDI data of a smaller byte size may only require using the *Route Control* or *Outseize Control* message. Using the *PPL Event Request* message may not be required.

The byte limitation for the total NPDI data (excluding redundant TLVs) in the API messages should not exceed 820 bytes.

> **Important!** The SIP MIME Information (0x2964) TLV collective maximum byte length of 780 bytes and the total NPDI data maximum byte length of 820 bytes combined can not support

messages of a size greater than 1500 bytes. The CSP nacks all SIP requests greater that 1500 bytes with the 513 message, Too Large.

API Call Control Messages

The following messages support the TLVs listed below that are used in the 0x0033 NPDI Universal ICB.

- *Route Control* (0x00E8)
- *Outseize Control* (0x002C)
- *Request for Service with Data* (0x002D)
- *PPL Event Indication* (0x0043)
- *PPL Event Request* (0x0044)

API and CSA Configuring and Querying

This feature can be configured either using the API or CSA.

API Configuring and Querying

The reporting of incoming MIME is a configurable option and is disabled by default. To enable this option, bit 20 of the *0x027F SIP Message Information Mask* TLV has to be set and sent in the *VoIP Protocol Configure* (0x00EE) message.

MIME can be sent in an outgoing INVITE message by using the *0x2936 SIP Tunnel Type* TLV, set to value 0x0004 Custom MIME Body in the *Route Control* (0x00E8) or *Outseize Control* (0x002C) message and followed by a *PPL Event Request* (0x0044) message for subsequent data. For a call, the tunnel type set using the SIP Tunnel Type TLV will have precedence over the tunnel type configured in the stack. If stack is preconfigured for a tunnel type, the host is not required to use the SIP Tunnel Type TLV (0x2936) unless the stack configuration for the tunnel type needs to be overridden.

MIME can also be sent in an outgoing INVITE message by using the *0x0270 SIP Tunnel Type* TLV, set to value 0x0004 Custom MIME Body in the *VoIP Protocol Configure* (0x00EE) message.

These TLVs enable or disable the use of tunneled data and specify the tunneled data type.

- The SIP Tunnel Type (0x2936) TLV can be used to set the tunneled data type to 0x0004 Custom MIME Body on a per call basis.
- The SIP Tunnel Type (0x0270) TLV can be used to set the tunneled data type to 0x0004 Custom MIME Body to configure the stack.

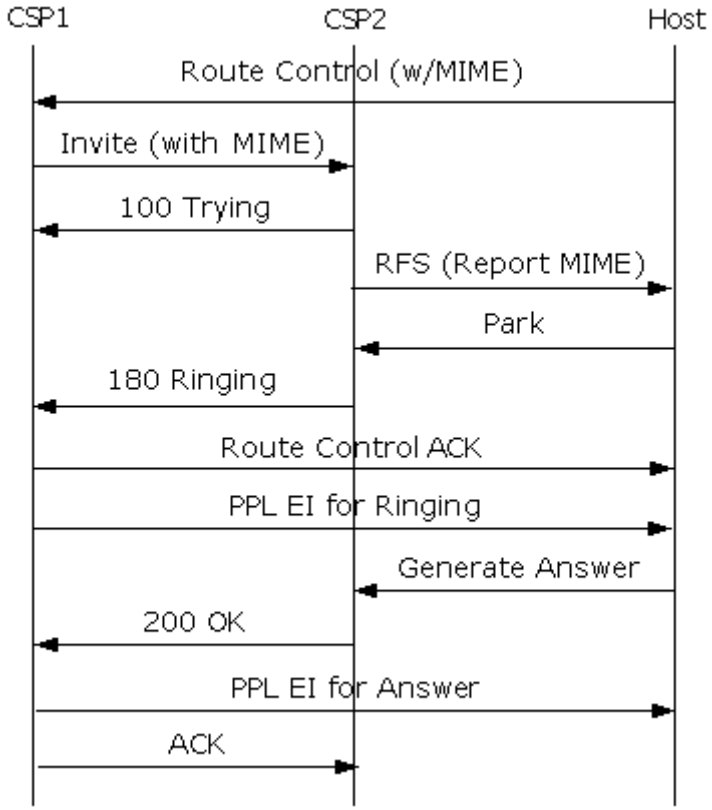
In case tunneled data type is set both during configuration time and also on a per call basis, then the tunneled data type set on the per call basis takes precedence.

The *VoIP Protocol Query* (0x00EF) message is used to check the SIP Message Information Mask value.

CSA Configuring and Querying

To configure and query the SIP stack for this feature, view the Configure SIP Advanced screen, Additional Host Signaling Parameters, and select **Report MIME Data in Incoming Messages**.

Call Flow



TLV Details

Modified TLVs

The following TLVs have been abbreviated to show the inclusion of the Support for Multipurpose Internet Mail Extensions information as indicated by the change bars.

0x0270 SIP Tunnel Type

For all outgoing messages, this TLV enables and disables the use of tunneled data and specifies the tunneled data type.

Used in:
VoIP Protocol Configure message
VoIP Protocol Query message

Byte	Description	
0, 1	Tag	0x0270
2, 3	Length	0x0002
4-6	Value[0-2]	0x0001 No tunneling (Default) 0x0002 CSP UPDF Tunneling 0x0004 Customize MIME Body

0x027F SIP Message Information Mask

Used in:
VoIP Protocol Configure message
VoIP Protocol Query message

The host uses this TLV to specify the additional SIP message information to the host in the *Request for Service with Data* message.

Byte	Description	
0, 1	Tag	0x027F
2, 3	Length	0x0004
4-7	Value[0-3]	This field is a 32-bit mask. Each bit selects specific SIP message fields and is listed from LSB to MSB. 0 - Disabled (Default) 1 - Enabled Bit 20 Selects reporting of MIME data in incoming messages. Bits 22-31 Reserved

0x2936 SIP Tunnel Type

For all outgoing messages, this TLV enables and disables the use of tunneled data and specifies the tunneled data type.Used in:
0x0033 NPDI Universal ICB in:
Route Control message
Outseize Control message
PPL Event Request message

Byte	Description	
0, 1	Tag	0x2936
2, 3	Length	0x0002
4-6	Value[0-2]	0x0001 No tunneling (Default) 0x0002 CSP UPDF Tunneling 0x0004 Custom MIME Body

0x295D Content Type

This TLV adds the content type to a SIP message. It also reports the content type, if present, in a SIP message. Prior to this feature, this TLV adds/reports content type only for the SIP INFO message. If there is a valid message body but the host has not supplied the content type, then the SIP stack adds the following content type as the default: “Content-Type: application/custom”

PPL Event Request message
PPL Event Indication message

Byte	Description	
0, 1	Tag	0x295D
2, 3	Length	Variable (Maximum of 100)
4-n	Value	Null terminated ASCII string

New TLVs

0x2964 SIP MIME Information

Used in:
0x0033 NPDI Universal ICB in:
Route Control message
Outseize Control message
Request for Service with Data message
PPL Event Request message
PPL Event Indication message

Byte	Description	
0, 1	Tag	0x2964
2, 3	Length	Variable (Maximum of 780 bytes)

0x2965SIP Content Encoding

Used in:
0x0033 NPDI Universal ICB

Byte	Description	
0, 1	Tag	0x2965
2, 3	Length	Variable
4-n	Value[0-n]	Null Terminated ASII String

0x2966SIP Content Disposition

Used in:
0x0033 NPDI Universal ICB

Byte	Description	
0, 1	Tag	0x2966
2, 3	Length	Variable
4-n	Value[0-n]	Null Terminated ASII String

0x2967 SIP Content Language

Used in:
0x0033 NPDI Universal ICB

Byte	Description	
0, 1	Tag	0x2967
2, 3	Length	Variable
4-n	Value[0-n]	Null Terminated ASII String

0x2968 SIP MIME Message Body

Used in:
0x0033 NPDI Universal ICB

Byte	Description	
0, 1	Tag	0x2968
2, 3	Length	Variable
4-n	Value[0-n]	Null Terminated ASII String

Outbound SIP Call with Call Agent Mode

Overview

With this feature, the CSP has the bearer on/off switching capability for outbound call legs too.

This capability is needed when the media services in the CSP (like DSP-2 card) or media transit path (like external TDM IVR or operator) is required for the called party.

Description

Call Agent Mode (CAM) in the CSP provides dynamic switching of media streams on or off the CSP RTP channels with a minimal amount of SIP messages.

Inbound Calls

Prior to this feature, the CSP supported bearer on/off switching of Call Agent Mode for inbound calls only. The CSP can connect the inbound leg of a SIP call to a TDM network or to a DSP media service.

This functionality allows the caller to be connected to an operator in a PSTN network. It also allows the application of DSP services such as playing announcements to the calling (inbound) leg.

Bearer switching is based on a coupling and decoupling mechanism. Coupling associates a physical timeslot with a virtual timeslot to enable bearer-switched service. Decoupling dissociates a physical timeslot from a virtual timeslot to enable bearer-free switched service.

The bearer switching takes place whenever required with minimal interaction from the host.

Outbound Calls

With this feature, the CSP has the bearer on/off switching capability for outbound call legs too.

This capability is needed when the media services in the CSP (like DSP-2 card) or media transit path (like external TDM IVR or operator) is required for the called party.

> **Example:** A directory assistance operator is located behind a TDM switch. The initial conversation between the operator and called party is bearer-switched through the CSP. Later, the operator drops out of the call flow while connecting the calling party and the called party bypassing the bearer path from the CSP.

PPL Event Request to Generate RE-INVITE message - PPL Event Request (0x0024)

The PPL Event Request (0x0024) is used to generate a RE-INVITE message to the SIP endpoint connected to the channel indicated in the channel AIB.

This PPL Event Request will work only if the call is in bearer-free mode or else it will NACK with 0x130C (Call is not in bearer-free mode.)

The PPL Event Request is added to the SIP UA (0x00A7) component. When the PPL Event Request generates a RE-INVITE message, the 200 OK SIP message received from the other end having the SDP information will be reported to the host using the PPL Event Indication (0x0020) by the SIP UA component.

The data in the PPL Event Indication could be used within the subsequent *Connect with Data* (0x0005) message. The received PPL Event Request is ACKED if the data is valid.

Generic PPL ICB in *Outseize Control* (0x002C) message

The *Outseize Control* (0x002C) message is enhanced to support the following two TLVs within the Generic PPL ICB (0x001E):

- Channel Service TLV (0x0116)
- Call Agent Physical Channel ID TLV (0x011A)

0x001E Generic PPL

ICB Type	0x03 (Extended Data)
ICB ID	0x001E
Data Length (2 bytes)	Variable
Number of TLVs (2bytes)	Variable
Tag	TLV 1 Tag
Length	TLV 1 Length
Value	TLV 1 Value
:	TLV n Tag
:	TLV n Length
:	TLV n Value
TLVs	0x0116 Channel Service 0x011A Call Agent Physical Channel ID

NACK

Value	Description	Corrective User Action
0x130C	Call not in bearer-free mode	Be sure call is in bearer-free mode.

API Messages Used

The following messages are used by this feature. Refer to the *API Reference* for the formats.

- *PPL Event Request* (0x0044) message
- *Outseize Control* (0x002E) message

PPL Information

PPL Component 0x00A7:

- Event Request 0x0024 - Generate RE-INVITE message
- Event Indication 0x0020 - Answer on B-Side

6 Call Agent

Purpose This chapter explains the Call Agent feature in the CSP.

Overview

Calls with or without Bearer Path

The Call Agent feature enables the CSP to connect VoIP calls with or without bearer paths through the CSP. In the current release, only SIP software uses the Call Agent functionality. The CSP can accept requests from SIP clients for a specific host-based voice service and then establish a SIP signaling connection for the requested service.

Modes

The Call Agent feature supports calls in the following two modes, on a call-by-call basis:

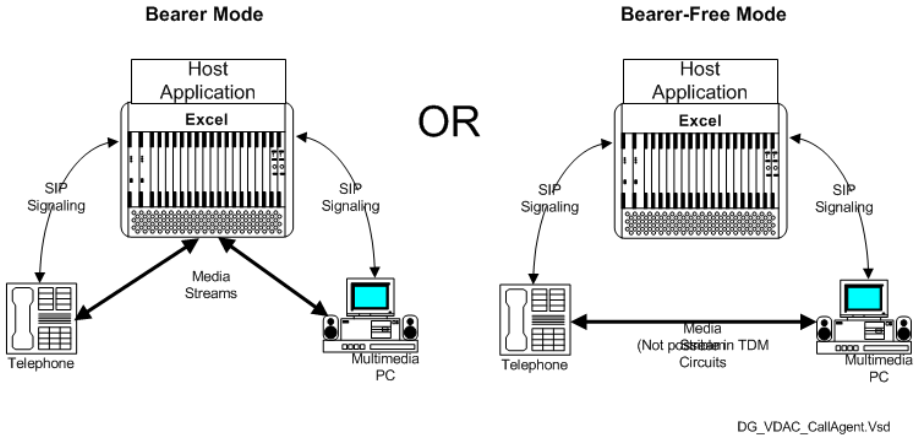
- bearer mode
- bearer-free mode

In bearer mode, the RTP stream passes through the VDAC-ONE or IP Network Interface Series 2 card. Bearer-free mode avoids "hairpinning" the call because the RTP stream does not have to pass through the CSP via the VDAC-ONE or IP Network Interface Series 2 card.

In addition, you can have a call channel transition between the two modes during a call. This dual switching scenario is especially applicable to pre-paid applications. See *Dual Switching Scenario (6-95)*.

In both modes, call control exists on the CSP Matrix Series 3 Card.

Figure 6-1 Bearer and Bearer-Free Mode



Benefits of Call Agent

The benefits of the Call Agent mode include the following:

- Lower CODEC costs
- Lower port costs
- Higher Quality of Service (fewer IP-TDM-IP transitions and lower delay)
- Increased scaling beyond CSP physical channel capacity
- Ability for host applications to use media resources separate from the CSP.
- Reduced bandwidth requirements, because no bearer packets go through the CSP.

Virtual and Physical Span/ Channels

With the Call Agent feature, there is a clear separation between virtual and physical span/channels.

The virtual span/channels provide the context for call processing only. The physical span/channels support bearer switching and call processing.

The Call Agent feature is optimized to offer maximum Busy Hour Call Attempt (BHCA) throughput when the total number of virtual span/ channel resources are divided into smaller resource groups within the internal router database.

> **Example:** If there are 10,000 virtual span/channels provisioned in the CSP, you should configure the internal router database with ten (or more) resource groups with 1024 (or less) span/channels to get maximum BHCA rate.

Host-to-CSP API

The *Route Control* message and *PPL Download* message support a criteria type named Virtual VoIP. During host-initiated routing, the host uses the *Route Control* message to hunt for a virtual VoIP channel and returns a virtual span/channel to the host.

Call Agent Switching

Call Agent switching is characterized by the absence of bearer resources, and therefore the absence of physical switching service, within the CSP. In Call Agent switching, Layer 4 (L4) and host applications are exposed to more network signaling data than in traditional switching, where only Layer 3 Plus (L3P) is exposed to this data. Call Agent switching propagates B-side media parameters to the A-side by sending a new *PPL Event Indication* message to host applications.

RFC 2833 Multi-Unicast

Some host applications that use bearer-free mode still need access to DTMF events. RFC 2833 meets this requirement.

The CSP can receive DTMF digit events from an endpoint and pass them to a host application. The CSP does this by requesting the RTP terminal to multi-unicast the RFC 2833 telephony event stream to the CSP and remote endpoint. The CSP does the following:

1. Terminates this stream on the CSP Matrix Series 3 Card.
2. Interprets the RFC 2833 stream.
3. Processes the events as requested by the host.
4. Passes the digit strings up to the host in the *Call Processing Event* (0x002E) message.

At this point the host can act on the event.

The CSP also supports the Subscribe and Notify method. Refer to *SUBSCRIBE and NOTIFY Method for DTMF Detection (5-250)*.

API Messages

The following API Messages support the Call Agent feature. They are documented in the *API Reference.*

- *Connect With Data (0x0005)*
- *PPL Event Indication (0x0043)*
- *PPL Table Download (0x00D6)*
- *Route Control (0x00E8)*
- *VoIP Protocol Configure (0x00EE)*
- *VoIP Protocol Query (0x00EF)*
- *DSP Service Request (0x00BD)*
- *DSP Service Cancel (0x00BE)*
- *Virtual Card Configure (0x00E0)* (supports virtual VDAC card for Call Agent)
- *Call Processing Event (0x002E)*

TLVs

The following TLVs support the Call Agent feature. They are documented in *Tag Length Value Blocks* chapter of the *API Reference*.

- 0x0065 Physical VoIP Channel
- 0x0071 Virtual VoIP Channel
- 0x0115 Call Agent Mode
- 0x0116 Channel Service TLV
- 0x0119 Media Offer Stage

- 0x01ED RFC 2833 Jitter Buffer Size
- 0x01EE Invalid Packet Alarm Threshold
- 0x01EF Initial UDP Port Number
- 0x01F0 Number of UDP Ports
- 0x01F1 Dynamic Payload Type
- 0x01 Add Virtual Card (supports virtual VDAC card for call agent)
- 0x02 Remove Virtual Card (supports virtual VDAC card for call agent)
- 0x011A Call Agent Physical Channel ID
- 0x2954 SIP Request URI User Name
- 0x2A00 Media Remote End Point Information
- 0x2A01 Media Per Stream Information
- 0x2A02 Media Per Codec Information
- 0x2A03 Media Type
- 0x2A07 Media Port
- 0x2A08 Media Payload Type
- 0x2A09 Media Payload Description
- 0x2A0A Media Payload Size
- 0x2A0B Media Clock Rate
- 0x2A0E Media Connection Address
- 0x2A13 Media Flow Direction

Atomic Functions

Atomic Functions numbers 225-234 support the Call Agent feature. They are documented in *Layer 4 Atomic Functions* in the *Developer's Guide: Programmable Protocol Language*.

NPDI Data Model

Overview

The Call Agent feature uses an NPDI data model that is structured to represent a multimedia call scenario. This call scenario has a single calling session and multiple underlying media sessions for carrying audio, video, and application data media streams. Each media stream is described by an “m=” line in the Session Description Protocol (SDP).

SIP is the signaling protocol used in the example but other IP signaling protocols could be used in the future.

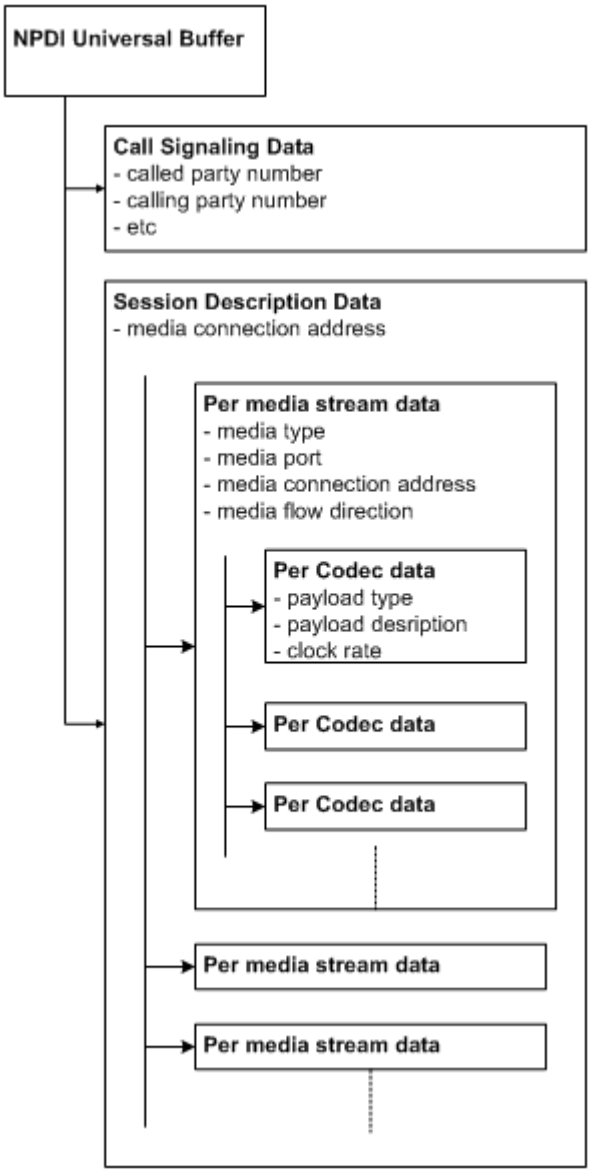
Hierarchical Model

The NPDI data model is hierarchical. It can have multiple levels of media streams in a call, and multiple codecs within a media stream. You can have a maximum of four media streams and eight codecs per media stream.

The following figure shows the NPDI data model with its flat call session data and nested media description data.

This figure shows how the data must be nested in TLVs. For example, the Per Media Stream Data and Per Codec Data must be nested as indicated in the figure.

Figure 6-2 NPDI Data Model



Call Setup

During the call setup, the originating SIP end point typically puts out a set of media sessions, each with a set of codec capabilities. In addition, each codec within a media stream can have its own attributes such as payload type and size. When setting up a call, the SIP software on the CSP consolidates media session parameters into two Session Description Data TLVs.

- Media Local End Point TLV (0x29FF)
- Media Remote End Point TLV (0x2A00)

Other TLVs containing additional information are nested below these two TLVs. Refer to the following TLVs in the Tag Length Value chapter of the *API Reference*:

- 0x2954 SIP Request URI User Name
- 0x2A00 Media Remote End Point Information
- 0x2A01 Media Per Stream Information
- 0x2A02 Media Per Codec Information
- 0x2A03 Media Type
- 0x2A07 Media Port
- 0x2A08 Media Payload Type
- 0x2A09 Media Payload Description
- 0x2A0A Media Payload Size
- 0x2A0B Media Clock Rate
- 0x2A0E Media Connection Address
- 0x2A13 Media Flow Direction

The SIP end points negotiate media parameters to set up the call based on their respective capabilities. For example, one SIP phone might accommodate audio, video and fax, while another might accommodate audio, fax, and SMS (Instant Messaging). When a call is made between the two, they can exchange audio and fax, but not video or SMS streams.

In this example, two SIP endpoints negotiate to set up streams for audio and fax, but ignore video and SMS. Because each stream's codecs are nested within a high-level TLV, the TLVs for the streams not being used can be ignored by the host application.

TLV Placement

This data model does not place any restrictions on the placement and positioning of individual TLVs between call signaling and media description areas. The TLVs can be placed between two areas based on the call scenario and data available on the signaling front.

NPDI Local and Remote Media Parameters

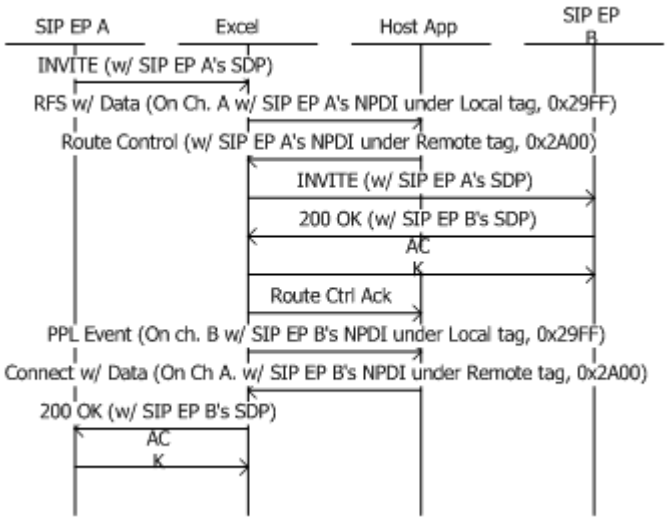
As described above, the SIP software on the CSP consolidates media session parameters into a Session Description Data TLV when setting up a call.

For bearer-free calls, the end point's Session Description Data is considered local or remote, depending on the span/channel that the call is on. (Note: This concept does not apply to bearer calls.)

When the call comes into the CSP, a *Request for Service* message with span/channel information is sent to the host application and this side of the call is considered local. The other end point is considered remote.

See the *Call Flow (6-10) for* more information on the local and remote sides.

Call Flow The call flow below details each step of the call setup for a bearer-free call.



Important! When an end point causes the CSP to send an API message to the host application that end point as well as the associated span/channel is considered the local side. (In the call flow above, these messages are the *Request for Service with Data* and *PPL Event Indication* messages.)

An incoming call manifests a *Request for Service with Data* message that from the CSP to the host that contains the NPDI Media Local End Point TLV (0x29FF) in the Session Description Data of the local side (channel A) of the call. This TLV must not exceed 250 bytes.

In order to switch the call to the egress side, the host delivers a *Route Control* message to the CSP. Apart from signaling the egress call, the *Route Control* message conveys channel A's media parameters to channel B. In order to indicate to channel B that the media parameters contained in the *Route Control* message belong to channel A, the host application copies over the NPDI Media Local End Point TLV (0x29FF) from the *Request for Service with Data* message to the NPDI Media Remote End Point TLV (0x2A00).

Conversely, the host application assigns the Media Remote End Point TLV (0x2A00) to the Session Description Data of the remote side (channel B) of the call. This happens in the *Route Control* message that the host sends to the CSP.

At this point, channel B knows about the channel A media parameters but channel A does not know about the channel B media parameters. Upon receipt of the media parameters over signaling, the channel B side sends its media parameters to the host in the *PPL Event Indication* message. The host application copies the contents of the *PPL Event Indication* (TLV 0x29FF) into the *Connect with Data* message (TLV 0x2A00) and sends it to the CSP. Evidently both channel A and channel B are now aware of each other's media parameters, as required for media communication to begin.

Message Trace

The following is a sample message trace for this process.

```
X->H
    [00 e7 00 2d 00 17 ff 00 01 0d 03 00 07 13 00 33 01 03 00 33
     00 d3 00 0b 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 05 31 32 33 34 00 29 1b 00 0d 31 30 2e 31 30 2e
     31 30 30 2e 31 30 00 29 23 00 05 34 34 34 34 00 29 25
     00 0d 31 30 2e 31 30 2e 31 30 30 2e 31 30 00 29 2d 00
     05 34 34 34 34 00 29 2f 00 0d 31 30 2e 31 30 2e 31 30
     30 2e 34 30 00 27 18 00 07 02 00 00 00 04 44 44 27 17
     00 05 02 00 04 12 34 29 ff 00 5e 2a 0e 00 04 0a 0a 64
     28 2a 01 00 52 2a 03 00 01 00 2a 07 00 04 00 00 22 3e
     2a 02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00
     04 00 00 1f 40 2a 02 00 13 2a 08 00 02 01 60 2a 09 00
     01 32 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02
     00 01 2a 09 00 01 01 2a 0b 00 04 00 00 1f 40]

H->X
    [00 0c 00 2d 00 17 ff 00 01 0d 03 00 07 13]

H->X
    [00 ac 00 e8 00 01 ff 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 71 00 0f 00
     01 0b 00 71 00 02 00 00 03 00 33 00 7d 00 04 27 7e 00
     03 08 01 00 29 23 00 05 34 34 34 34 00 29 19 00 05 31
     32 33 34 00 2a 00 00 5e 2a 0e 00 04 0a 0a 64 28 2a 01
     00 52 2a 03 00 01 00 2a 07 00 04 00 00 22 3e 2a 02 00
     13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04 00 00
     1f 40 2a 02 00 13 2a 08 00 02 01 60 2a 09 00 01 32 2a
     0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a
     09 00 01 01 2a 0b 00 04 00 00 1f 40]

X->H
    [00 14 00 e8 00 01 ff 00 10 01 02 1e 09 00 01 00 39 00 03 00
    07 14]

H->X
    [00 11 00 bf 00 01 ff 00 02 0d 03 00 07 13 0d 03 00 07
     13]

X->H
    [00 07 00 bf 00 01 ff 00 10]
```

```
X->H
    [00 63 00 43 00 03 ff 00 01 0d 03 00 07 14 00 a7 00 20 01 03
     00 33 00 4d 00 01 29 ff 00 47 2a 0e 00 04 0a 0a 64 05
     2a 01 00 3b 2a 03 00 01 00 2a 07 00 04 00 00 8e 30 2a
     02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04
     00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01
     01 2a 0b 00 04 00 00 1f 40]

H->X
    [00 05 00 43 00 03 ff]

H->X
    [00 72 00 05 00 01 ff 00 02 0d 03 00 07 13 0d 03 00 07
     14 01 02 03 00 1e 00 08 00 01 01 16 00 02 00 01 03 00
     33 00 4d 00 01 2a 00 00 47 2a 0e 00 04 0a 0a 64 05 2a
     01 00 3b 2a 03 00 01 00 2a 07 00 04 00 00 8e 30 2a 02
     00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04 00
     00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01 01
     2a 0b 00 04 00 00 1f 40]

X->H
    [00 07 00 05 00 01 ff 00 10]



1 -RECEIVED From 10.10.100.40:1028 at 2588
INVITE sip:1234@10.10.100.10 SIP/2.0
From: sip:4444@10.10.100.10;tag=1c29051
To: sip:1234@10.10.100.10
Call-Id: call-973574993-5@10.10.100.40
Cseq: 1 INVITE
Contact: sip:4444@10.10.100.40
Content-Type: application/sdp
Content-Length: 197
Accept-Language: en
Allow: INVITE, ACK, CANCEL, BYE, REFER, OPTIONS, NOTIFY,
   REGISTER, SUBSCRIBE
Supported: sip-cc, sip-cc-01, timer
User-Agent: Pingtel/1.2.6 (VxWorks)
Via: SIP/2.0/UDP 10.10.100.40

v=0
o=Pingtel 5 5 IN IP4 10.10.100.40
s=phone-call
c=IN IP4 10.10.100.40
t=0 0
m=audio 8766 RTP/AVP 0 96 8
a=rtpmap:0 pcmu/8000/1
a=rtpmap:96 telephone-event/8000/1
a=rtpmap:8 pcma/8000/1

2 -SENT To 10.10.100.40:5060 at 2588
SIP/2.0 100 Trying
To: sip:1234@10.10.100.10;tag=4206a1c
From: sip:4444@10.10.100.10;tag=1c29051
Call-ID: call-973574993-5@10.10.100.40
CSeq: 1 INVITE
Contact: 1234<sip:1234@10.10.100.10:5060>
Via: SIP/2.0/UDP 10.10.100.40
User-Agent: Excel/82.0.66
Content-Length: 0


3 -SENT To 10.10.100.5:8982 at 2589
INVITE sip:10.10.100.5:8982 SIP/2.0
```

```
Via: SIP/2.0/UDP 10.10.100.10
To: 1234<sip:1234@10.10.100.5:8982>
From: 4444<sip:4444@10.10.100.10:5060>;tag=41165572a1d
Call-ID: Excel-CSP255.1014.2589.0@10.10.100.10
Contact: 4444<sip:4444@10.10.100.10:5060>
User-Agent: Excel/82.0.66
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 202

v=0
o=sip 946687389 946687389 IN IP4 10.10.100.10
s=SIP_Call
c=IN IP4 10.10.100.40
t=0 0
m=audio 8766 RTP/AVP 0 96 8
a=rtpmap:0 PCMU/8000
a=rtpmap:96 telephone-event/8000
a=rtpmap:8 PCMA/8000

4 -RECEIVED From 10.10.100.5:1298 at 2589
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.100.10
From: 4444<sip:4444@10.10.100.10:5060>;tag=41165572a1d
To: 1234<sip:1234@10.10.100.5:8982>;tag=08ca52fc-4b12-4f11-
   a101-55fbc433e3b1
Call-ID: Excel-CSP255.1014.2589.0@10.10.100.10
CSeq: 1 INVITE
User-Agent: Windows RTC/1.0
Content-Length: 0


5 -RECEIVED From 10.10.100.5:1298 at 2589
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.10.100.10
From: 4444<sip:4444@10.10.100.10:5060>;tag=41165572a1d
To: 1234<sip:1234@10.10.100.5:8982>;tag=08ca52fc-4b12-4f11-
   a101-55fbc433e3b1
Call-ID: EXCEL-CSP255.1014.2589.0@10.10.100.10
CSeq: 1 INVITE
User-Agent: Windows RTC/1.0
Content-Length: 0


6 -SENT To 10.10.100.40:5060 at 2589
SIP/2.0 180 Ringing
To: sip:1234@10.10.100.10;tag=4206a1c
From: sip:4444@10.10.100.10;tag=1c29051
Call-ID: call-973574993-5@10.10.100.40
CSeq: 1 INVITE
Contact: 1234<sip:1234@10.10.100.10:5060>
Via: SIP/2.0/UDP 10.10.100.40
User-Agent: Excel/82.0.66
Content-Length: 0


7 -RECEIVED From 10.10.100.5:1298 at 2592
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.10.100.10
From: 4444<sip:4444@10.10.100.10:5060>;tag=41165572a1d
To: 1234<sip:1234@10.10.100.5:8982>;tag=08ca52fc-4b12-4f11-
   a101-55fbc433e3b1
```

```
Call-ID: EXCEL-CSP255.1014.2589.0@10.10.100.10
CSeq: 1 INVITE
Contact: <sip:10.10.100.5:8982>
User-Agent: Windows RTC/1.0
Content-Type: application/sdp
Content-Length: 159

v=0
o=vrao 0 0 IN IP4 10.10.100.5
s=SIP_Call
c=IN IP4 10.10.100.5
b=CT:1000
t=0 0
m=audio 36400 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000

8 -SENT To 10.10.100.5:8982 at 2592
ACK sip:10.10.100.5:8982 SIP/2.0
Via: SIP/2.0/UDP 10.10.100.10
To: 1234<sip:1234@10.10.100.5:8982>;tag=08ca52fc-4b12-4f11-
   a101-55fbc433e3b1
From: 4444<sip:4444@10.10.100.10:5060>;tag=41165572a1d
Call-ID: EXCEL-CSP255.1014.2589.0@10.10.100.10
CSeq: 1 ACK
Content-Length: 0


9 -SENT To 10.10.100.40:5060 at 2593
SIP/2.0 200 OK
To: sip:1234@10.10.100.10;tag=4206a1c
From: sip:4444@10.10.100.10;tag=1c29051
Call-ID: call-973574993-5@10.10.100.40
CSeq: 1 INVITE
Contact: 1234<sip:1234@10.10.100.10:5060>
Require: timer
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP 10.10.100.40
User-Agent: Excel/82.0.66
Content-Type: application/sdp
Content-Length: 165

v=0
o=sip 946687393 946687393 IN IP4 10.10.100.10
s=SIP_Call
c=IN IP4 10.10.100.5
t=0 0
m=audio 36400 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000

10-RECEIVED From 10.10.100.40:1028 at 2593
ACK sip:1234@10.10.100.10:5060 SIP/2.0
Contact: sip:4444@10.10.100.40
From: sip:4444@10.10.100.10;tag=1c29051
To: sip:1234@10.10.100.10;tag=4206a1c
Call-Id: call-973574993-5@10.10.100.40
Cseq: 1 ACK
Session-Expires: 1800
Accept-Language: en
User-Agent: Pingtel/1.2.6 (VxWorks)
Via: SIP/2.0/UDP 10.10.100.40
Content-Length: 0
```

Virtual IP Channels

Licensing of Virtual IP Channels

To use the Call Agent feature, you must configure virtual IP channels in the CSP.

> **Important!** To enable more than 2,000 virtual channels you require at least two licenses, one to enable the initial virtual 2,000 channels, and a second license to incrementally increase to greater than 2,000 virtual channels. You also must license SIP software to use the Call Agent feature.

The following licensing options are available by model number:

- Initial license for 2,000 virtual IP channels
- Incremental 2,000 virtual IP channels
- Incremental 4,000 virtual IP channels
- Incremental 6,000 virtual IP channels
- Incremental 8,000 virtual IP channels

Refer to *Downloading License Keys to the CSP* in the *Licensing Overview* chapter in the *Developer's Guide: Overview* and the *Product License Download* message (0x0079) in the *API Reference.*

Virtual Cards

Virtual IP channels function on virtual cards. Virtual cards function the same way as physical cards in all respects. You logically insert and remove virtual cards for the CSP without interruption to service using the *Virtual Card Configure (0x00E0)* message.

> **Important!** The same messages that reset a physical card's tag configuration also reset the equivalent virtual card. You must send the *Tag Configuration* (0x00D0) message to retag the virtual card. Refer to the *Tag Configuration (0x00D0)* message in the *API Reference* for a list of configuration messages that reset the tag configuration of physical cards.

The host receives the *Card Status Report (0x00A6)* as if the virtual card actually existed. The *Virtual Span Control (0x00E2)* message supervises the actual spans that the virtual card controls.

The following card type represents the 32-span version of the Virtual VDAC card.

- 0x80 Virtual VDAC, 32-span

To support the expanded number of channels, you must extend the range of slots using the *Virtual Card Configure (0x00E0)* message.

The following table breaks down the number of virtual cards required for the additional virtual spans.

Table 6-1 Virtual VDAC Cards and Spans

Channels Licensed	Virtual VDAC Cards	Virtual Channels	Virtual Spans
2,000	2	2,048	64
4,000	4	4,096	128
6,000	6	6,144	192
8,000	8	8,192	256
10,000	10*	10,080	315

* Nine virtual VDAC cards have 32 spans. The tenth card has 27 spans to get to the maximum 315 virtual VDAC spans.

Important! When you configure any virtual card, the resources are allocated for the maximum configuration. For example, when you configure a 32-span VDAC virtual card, the resources are reserved for all 32 spans even if you configure only one span.

Refer to the section, *Configuring SS7 10K Virtual Channels* in the *Developer's Guide: Common Channel Signaling* for a full explanation.

Exnet® Ring

You cannot connect virtual IP channels over the Exnet® Ring. In order for a call to be connected over the Exnet® Ring, the channels must be terminated locally (that is, on the CSP or Exnet Connect® card).

Host-Based Control of Channel Allocation

Application developers can have the host application hunt and allocate the coupled physical VoIP channel in a Call Agent Mode call.

This coupling can be achieved in two ways:

- The host can send the Call Agent Mode Physical Channel ID TLV (0x011A) in the *Connect with Data* message. By doing so, the host conveys the physical VoIP span/channel identifier to the CSP.
- The host can send the CSP a *PPL Event Request* message that includes the following:
  - Call Agent Mode Physical Channel ID TLV (0x011A)
  - PPL Component - L4 CH PPL (0x0061)
  - PPL Event Request - Modify Bearer Service (0x00CC)

  Important! This message is used when a request for DSP media services is involved.

Conversely, this *PPL Event Request* message can decouple the physical span/channel from a virtual span/channel in Call Agent Mode. By doing so, the PPL Event Request allows the host application developers to toggle the bearer nature of a virtual VoIP channel. When a physical channel is coupled with the virtual channel, service is modified to bearer-switched. When the physical channel is decoupled from the virtual channel, service is modified to bearer-free.

With this feature, the virtual channel does not need to request a physical channel from the L4 Router. Instead, the virtual channel busys-out the physical channel and removes it from the free pool of L4 Router channels.

The reverse happens when the call is torn down or when the virtual channel transitions from bearer-switched to bearer-free mode.

Process

The following summarizes how the host can hunt and allocate the coupled physical VoIP channel on the inbound side of a Call Agent Mode call received by the CSP.

1. The host sends the CSP either of the following messages:
   - *Connect with Data* message, with the Call Agent Physical Channel ID TLV (0x011A). This TLV contains the physical span/channel to use for the call.

- *PPL Event Request* message, with the Modify Bearer Service Event. This message contains the PPL Generic ICB to carry the Physical Channel ID TLV and an NPDI Universal ICB to carry the physical VoIP channel parameters (Source IP 0x2792 and Source RTP 0x2793). The CSP must be in Answer State because this message causes SIP RE-INVITE messages.

2. The virtual channel requests the L4 Router to busy-out the physical channel in the pool of free channels.
3. The virtual channel busys-out the physical channel, essentially coupling the virtual channel and the physical channel.
4. The CSP takes the appropriate signaling action, such as sending a SIP RE-INVITE message to the caller.

Refer to the following:

- Call Agent Mode Physical Channel ID TLV (0x011A) in the *Tag Length Value Blocks* chapter in the *API Reference*.
- *CH - Channel Management (0x0061)* in the *Layer 4 Call Control PPL Information* chapter in the *Developer's Guide: Overview.*

Deploying Call Agent

Basic Steps

To deploy the Call Agent feature on your CSP, you must follow the basic steps below. Refer to the indicated portions of the CSP publications set for additional information necessary to complete these steps.

1. Activate the Call Agent feature by downloading the appropriate Call Agent licenses. Refer to *Licensing of Virtual IP Channels (6-15)* and *Downloading License Keys to the CSP* in the *Licensing Overview* chapter in the *Developer's Guide: Overview* and the *Product License Download* message (0x0079) in the *API Reference.*
2. Configure the SIP software. See in the Session-Initiation Protocol chapter.

   You configure the SIP software on the CSP with the *VoIP Protocol Configure* message (0x00EE). This message supports the Call Agent Mode TLV (0x0115) which controls whether or not Call Agent is used. Without this TLV, Call Agent is not enabled.

3. Provision the virtual VDAC cards and the virtual span/channels on them.

   A virtual VDAC card is a virtual 32-span card that provides a call processing context for the CSP switching software, as well as a licensing mechanism. The virtual cards are assigned to virtual slots: numbers 0x40-0x5F. You can assign up to 10,000 virtual IP channels on a node. See the *API Reference* for details on the *Virtual Card Configure* message (0x00E0) and the Add Virtual Card TLV (0x01).

4. Configure the router database to route bearer-free calls to virtual span/channels.

   The criteria type Virtual VoIP Channel (0x0071) allows host applications to hunt and seize Virtual VoIP span/channels if they require bearer-free calls. See the *PPL Table Download* message (0x00D6) in the *API Reference*.

5. Configure RFC 2833 Multi-Unicast. This step is optional. To enable RFC 2833, use TLV 0x01E2 RFC 2833 Enable in the *Resource Attribute Configure* message (0x00E3).

To configure the key RFC 2833 attributes, such as port number and RTP payload type, use TLV 0x01F1 Dynamic Payload Type with the *VoIP Protocol Configure* message (0x00EE). Note that RFC 2833 is a licensed feature per CSP chassis. For details, refer to *Downloading License Keys to the CSP* in the *Licensing Overview* chapter in the *Developer's Guide: Overview* and the *Product License Download* message (0x0079) in the *API Reference.*

Activating Call Agent

To activate Call Agent you must use a Product License Key, which Dialogic provides when you purchase the Product License. The key is unique and encrypted.

Follow the steps below:

1. Download the Product License to enable Call Agent software. Refer to *Product Licenses* in the *Developer's Guide: Overview.*

2. The Call Agent license key is provided as a text file, usually *LICENSE.CFG* or *SERIALNO.CFG.*

   For example, the file named *3452.CFG* indicates that this license file can be downloaded to only the CSP Matrix Series 3 Card in chassis serial number 3452.

3. You must download the license key each time new system software is downloaded or when you cycle the power.

Important! Be certain to download your license files from the smallest increment to the largest increment.

Sample Configuration File for Call Agent

Overview

The sample configuration file in this section accomplishes the following.

- Assigns logical node ID
- Deletes existing spans
- Enables a virtual span license
- Enables a SIP license
- Enables a Call Agent license
- Enable an RFC 2833 multi-unicast license
- Deletes VDAC IP address
- Assigns a VDAC IP address
- Assigns VDAC-based logical spans
- Configures route and resource tables
- Brings spans into service
- Brings channels into service
- Makes virtual spans operational
- Configures the SIP protocol with Call Agent mode
- Enables local registration lookup
- Configure RFC 2833 VoIP protocol

Spans in this file

This file configures three spans for Call Agent as follows.

- Span 6 - VDAC
- Span 7 - Virtual VDAC
- Span 8 - Virtual T-ONE

Fifteen channels are made routable on each span.

Configuration File (Best viewed in PDF file)

```
'------------------------------------------------------------------------------

'------------------------------------------------------------------------------
' Assign Logical Node ID
'------------------------------------------------------------------------------
'                  _logical_node_id
'                  |
'00 0e 00 10 00 00 ff 00 01 10 05 00 00 1a df ff

'------------------------------------------------------------------------------
' Delete currently assigned spans
'------------------------------------------------------------------------------
00 0d 00 a8 00 00 ff 00 01 11 04 ff ff ff ff

'------------------------------------------------------------------------------
' Enable software license for Virtual Spans (for shelf serial # 00 00 02 3b)
'------------------------------------------------------------------------------
00 19 00 79 00 00 ff 01 02 24 10 30 37 42 4a 43 34 38 49 30 41 33 55 58 4f 36 53

'------------------------------------------------------------------------------
' Enable software license for SIP (for shelf serial # 00 00 02 3b)
'------------------------------------------------------------------------------
00 19 00 79 00 00 ff 01 02 24 10 30 38 42 4a 43 34 38 49 30 42 32 34 54 50 4f 4b

'------------------------------------------------------------------------------
' Enable software license for Call Agent (for shelf serial # 00 00 02 3b)
'------------------------------------------------------------------------------
00 19 00 79 00 00 ff 01 02 24 10 30 39 42 4a 43 34 38 49 30 41 33 44 4c 4e 36 38

'------------------------------------------------------------------------------
' Enable software license for 2833 multi-unicast (for shelf serial # 00 00 02 3b)
'------------------------------------------------------------------------------
00 19 00 79 00 00 ff 01 02 24 10 30 41 42 4a 43 34 38 49 30 42 34 31 35 54 4d 53

'------------------------------------------------------------------------------
                           ' Delete IP addresses assigned to VDAC-1 modules and
                              mother board
'------------------------------------------------------------------------------
'                                  _ VDAC slot #
'                                  |
00 45 00 e7 00 00 ff 00 01 01 01 02 00 06 \
01 09 ff ff ff ff ff ff ff ff 00 \ ' mother board; IP address, Subnet mask
01 09 00 ff ff ff ff ff ff ff 00 \ ' module 0
01 09 01 ff ff ff ff ff ff ff 00 \ ' module 1
01 09 02 ff ff ff ff ff ff ff 00 \ ' module 2
01 09 03 ff ff ff ff ff ff ff 00 \ ' module 3
02 00 ' engage IP action
```

```
'------------------------------------------------------------------------------
' Assign IP addresses to VDAC-1 modules and mother board
'------------------------------------------------------------------------------
'                                 _ VDAC slot #
'                                 |
00 4c 00 e7 00 00 ff 00 01 01 01 02 00 07 \
01 09 ff 10 10 10 02 ff ff ff 00 \ ' mother board; IP address, Subnet mask
01 09 00 10 10 10 03 ff ff ff 00 \ ' module 0
01 09 01 10 10 10 04 ff ff ff 00 \ ' module 1
01 09 02 10 10 10 05 ff ff ff 00 \ ' module 2
01 09 03 10 10 10 06 ff ff ff 00 \ ' module 3
05 05 05 10 10 10 01 \ ' IP gateway for all boards
02 00 ' engage IP action


'------------------------------------------------------------------------------
' Assign VDAC-1 based logical spans
'------------------------------------------------------------------------------
'              logical_span_id_msb_  _lsb
'                                 |  |  _slot
'                                 |  |  |  _physical_span_offset
'                                 |  |  |  |
00 0d 00 a8 00 00 ff 00 01 11 04 00 06 02 04

'------------------------------------------------------------------------------
' Assign VDAC and T1 virtual spans
' - assign virtual slot
' - assign a logical span on the virtual slot
'------------------------------------------------------------------------------
' board type = d2 = 16 span virtual T1
' board type = 80 = 32 Span Virtual VoIP
'
                          ' VIRTUAL VDAC
'



                                            virtual slot
'                         TLVs              /
'                    null To      TLV  Len /
'                    AIB  flw Len Type /  / board type
'                    __|__ |  |   |   /  / /
00 0d 00 e0 00 00 ff 00 00 01 01 01 02 40 80
'
'                                      _slot
'                                      |
00 0d 00 a8 00 00 ff 00 01 11 04 00 07 40 00
'
' VIRTUAL T1
'                                      _virtual slot
```

```
'                                        |   _board type
'                                        |   |
00 0d 00 e0 00 00 ff 00 00 01 01 01 02 41 d2
'
'                                         _slot
'                                         |
00 0d 00 a8 00 00 ff 00 01 11 04 00 08 41 00


'-------------------------------------------------------------------------
' Configure route and resource tables
'-------------------------------------------------------------------------
' Route_Table
00 21 00 d5 00 00 ff 00 64 03 01 00 00 00 67 72 6f 75 74 65 72 5f 74 61 62 6c 65 00
   00 00 00 00 00 00 00
00 30 00 d6 00 00 ff 00 64 03 01 00 01 80 02 00 01 00 04 27 17 03 00 09 03 06 00 00
   00 00 00 09 0f 00 02 00 0d 00 15 00 02 00 02 00 63 00 03 01 00 01
00 29 00 d6 00 00 ff 00 64 03 01 00 02 c0 01 00 01 00 01 00 65 01 00 02 00 00 00 02
   00 0d 00 15 00 02 00 01 00 63 00 03 01 00 02
00 29 00 d6 00 00 ff 00 64 03 01 00 03 c0 01 00 01 00 02 00 71 01 00 02 00 00 00 02
   00 0d 00 15 00 02 00 02 00 63 00 03 01 00 03
' Resource_Group_Table
00 21 00 d5 00 00 ff 00 64 04 01 00 00 00 48 72 65 73 2f 67 72 6f 75 70 20 74 61 62
   6c 65 00 00 00 00 00
00 37 00 d6 00 00 ff 00 64 04 01 00 01 00 01 00 01 00 01 00 0c 01 02 0d 03 00 06 00
   0d 03 00 06 0f 00 03 00 14 01 68 00 04 10 10 10 03 01 00 00 02 00 01 01 01 00 02
   00 01
00 23 00 d6 00 00 ff 00 64 04 01 00 02 00 01 00 01 00 01 00 0c 01 02 0d 03 00 07 00
   0d 03 00 07 0f 00 00 00 00
```

```
                          '-------------------------------------------------------
                            -------------------
' Bring logical spans in service
'-------------------------------------------------------------------------------
00 0b 00 0a 00 00 ff 00 01 0c 02 00 06 f0
00 0b 00 0a 00 00 ff 00 01 0c 02 00 07 f0
00 0b 00 0a 00 00 ff 00 01 0c 02 00 08 f0


'-------------------------------------------------------------------------------
' Bring channels in service
'-------------------------------------------------------------------------------
00 12 00 0a 00 00 ff 01 02 0d 03 00 06 00 0d 03 00 06 1f f0
00 12 00 0a 00 00 ff 01 02 0d 03 00 07 00 0d 03 00 07 0f f0
00 12 00 0a 00 00 ff 01 02 0d 03 00 08 00 0d 03 00 08 0f f0

'-------------------------------------------------------------------------------
' Make virtual spans operational
'-------------------------------------------------------------------------------
'                                  ____virtual_span_id
'                                  |  |
00 0e 00 e2 00 00 ff 00 01 0c 02 00 07 00 01 00
00 0e 00 e2 00 00 ff 00 01 0c 02 00 08 00 01 00

'-------------------------------------------------------------------------------
' Configure VoIP protocol (SIP)
' w/ call agent mode turned on
'-------------------------------------------------------------------------------
00 1e 00 ee 00 00 ff 00 00 00 00 04 01 c8 00 01 04 02 6a 00 01 04 02 6b 00 \
01 02 01 15 00 01 01

'-------------------------------------------------------------------------------
' Configure local registration lookup
'-------------------------------------------------------------------------------
00 14 00 ee 00 00 ff 00 00 00 00 02 01 c8 00 01 04 02 79 00 01 01


'-------------------------------------------------------------------------------
' Configure RFC 2833 for Call Agent
'-------------------------------------------------------------------------------
00 2c 00 ee 00 00 ff 00 00 00 00 \    'voip config
06 \                                  'number of tlvs
01 c8 00 01 06 \                      'protocol type (2833)
01 ed 00 02 00 80 \                   'jitter buffer size (80 milliseconds)
01 ee 00 02 00 10 \                   'invalid packet limit
01 ef 00 02 27 10 \                   'initial udp port #
01 f0 00 02 00 10 \                   'num udp ports = 16
01 f1 00 01 65                        'dynamic payload type
```

Call Agent Reconnect

Purpose

In general, scenarios like the one below require call agent to reconnect calls.

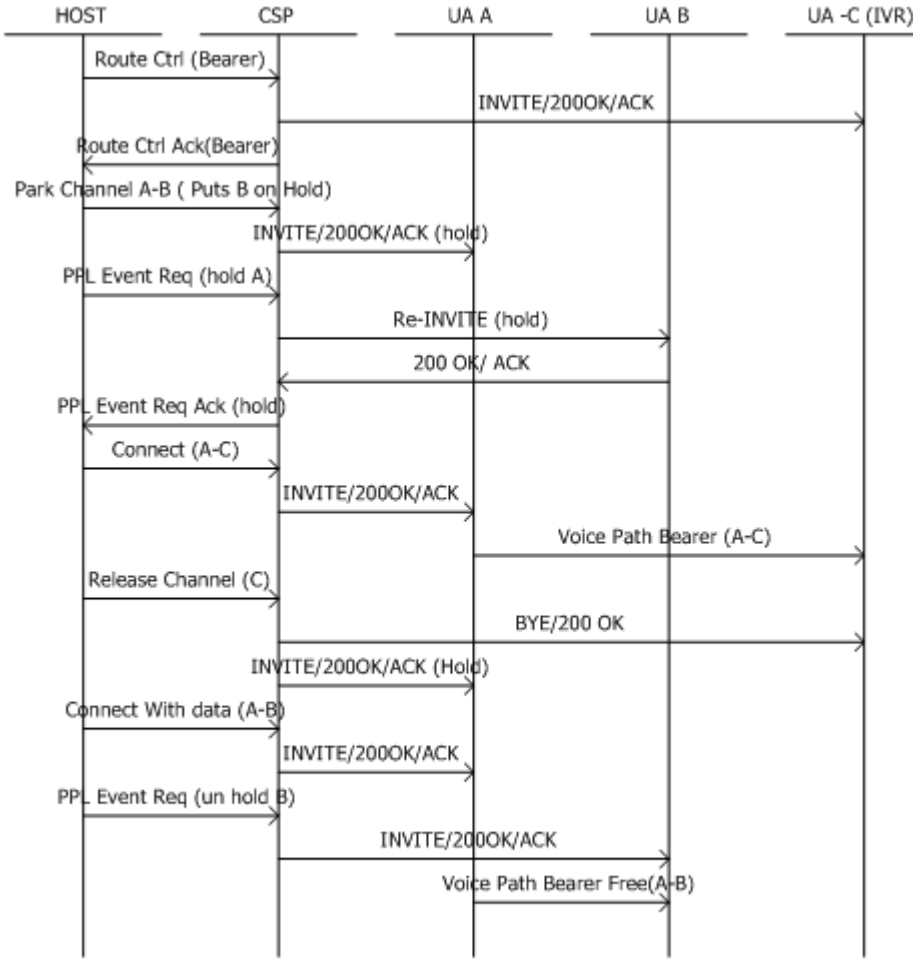
A and B are SIP endpoints. C could be an IP/PSTN channel or a DSP resource playing an announcement.

1. A and B are connected in bearer-free mode.
2. B is put on hold.
3. A gets connected to C in bearer mode.
4. C is released.
5. A get reconnected to B.

This section explains two specific reconnect scenarios supported by the CSP. Each scenario includes a call flow and message trace.

Using Third Party Integrated Voice Recognition Equipment

The following call flow assumes endpoints A and B are already connected in bearer-free mode.



Message Trace

Initial State

```
[0x1004 (4100)] (07,04) <--> [0x1003 (4099)] (07,03) 2-
   Way Connect (Endpoint A)
[0x1003 (4099)] (07,03) <--> [0x1004 (4100)] (07,04) 2-
   Way Connect  (Endpoint B)
```

1. *Route Control* message to Endpoint C (Bearer ON)

```
H->X
   [00 81 00 e8 00 05 ff 00 01 29 02 ff fe 02 03 00 1e 00
   19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 52 00 09 27 7e 00
   03 08 01 00 29 37 00 01 00 29 38 00 07 6c 75 63 65 6e
     74 00 29 19 00 05 33 33 33 33 00 29 1b 00 0c 31 30 2e
   31 30 2e 36 35 2e 33 37 00 29 24 00 05 33 36 33 38 00
```

```
    29 3a 00 05 31 32 33 34 00 29 3c 00 05 31 32 33 34 00
   29 14 00 01 00]
X->H
    [00 14 00 e8 00 05 ff 00 10 01 02 1e 09 00 01 00 39 00
   03 00   01 07]
```

2. Park A-B

```
H->X
[00 11 00 bf 00 03 ff 00 02 0d 03 00 07 03 0d 03 00 07
   04]
X->H   [00 07 00 bf 00 03 ff 00 10]

Resulting Channel State
[0x1003 (4099)] (07,03) <-- IDLE, L4 STATE 7
[0x1004 (4100)] (07,04) <-- IDLE, L4 STATE 7
```

3. Put B on Hold (optional)

```
H->X
[00 1d 00 44 00 04 ff 00 01 0d 03 00 07 04 00 a7 00 1e 01
   03 00 33 00 07 00 01 27 b3 00 01 03]
X->H  [00 07 00 44 00 04 ff 00 10]
```

Resulting Channel State

```
[0x1003 (4099)] (07,03) <-- IDLE, L4 STATE 7
[0x1004 (4100)] (07,04) <-- IDLE, L4 STATE 7
```

4. Connecting A to C

```
H->X
[00 11 00 00 00 06 ff 00 02 0d 03 00 07 03 0d 03 00 01
   07]
X->H  [00 07 00 00 00 06 ff 00 10]
```

Resulting Channel State

```
[0x1003 (4099)] (07,03) <--> [0x25f (0607)] (01,07) 2-Way
   Connect
[0x1004 (4100)] (07,04) <-- IDLE, L4 STATE 7

[0x25f (0607)] (01,07) <--> [0x287 (0647)] (01,08) 2-Way
   Connect
[0x287 (0647)] (01,08) <--> [0x25f (0607)] (01,07) 2-Way
   Connect
```

5. Releasing C

```
H->X
[00 11 00 08 00 07 ff 00 02 0d 03 00 01 07 0d 03 00 01 07]
X->H  [00 07 00 08 00 07 ff 00 10]

X->H
[00 57 00 69 00 11 ff 00 01 0d 03 00 01 07 02 02 1e 2a 00
   05  01 04 00 04 ff ff ff ff 01 05 00 04 00 00 05 e9 01
   11  00 04 00 00 05 ee 01 10 00 04 00 03 b1 a0 01 12 00
   04  00 03 b4 c0 03 00 33 00 18 00 03 27 4e 00 02 00 10
   27  92 00 04 0a 0a 41 65 27 93 00 04 00 00 28 bc]
H->X [00 05 00 69 00 11 ff]
```

Channel A is parked by L4

```
X->H [00 0e 00 42 00 3b ff 00 01 0d 03 00 07 03 04 00]
H->X [00 05 00 42 00 3b ff]
```

Resulting Channel State

```
[0x1003 (4099)] (07,03) <-- IDLE, L4 STATE 7
[0x1004 (4100)] (07,04) <-- IDLE, L4 STATE 7
```

6. Reconnecting A to B

```
H->X
[00 4e 00 05 00 08 ff 00 02 0d 03 00 07 03 0d 03 00 07   04
   01 02 03 00 1e 00 08 00 01 01 16 00 02 00 01 03 00  33
   00 29 00 01 2a 00 00 23 2a 0e 00 04 0a 0a 41 87 2a   01
   00 17 2a 03 00 01 00 2a 07 00 04 00 00 14 f8 2a 02  00
   06 2a 08 00 02 00 02]
X->H  [00 07 00 05 00 08 ff 00 10]
```

Resulting Channel State (Same as initial state)

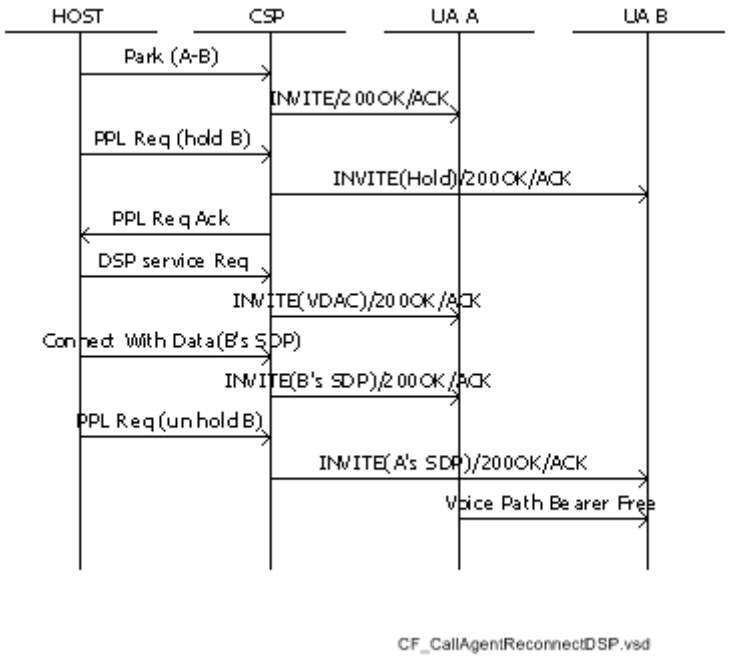
```
[0x1003 (4099)] (07,03) <--> [0x1004 (4100)] (07,04) 2-
   Way Connect
[0x1004 (4100)] (07,04) <--> [0x1003 (4099)] (07,03) 2-
   Way Connect
```

7. Putting B on 2 way voice path (un hold)

```
H->X
[00 1d 00 44 00 09 ff 00 01 0d 03 00 07 04 00 a7 00 1e  01
   03 00 33 00 07 00 01 27 b3 00 01 00]
X->H  [00 07 00 44 00 00 ff 00 10]
```

Using DSP Card

In the following call flow there is no third party equipment involved for playing announcements or collecting digits. The CSP uses the resources on the DSP card instead.



Message Trace

Recorded Announcement Connect message (0x0055)

```
H->X
[00 23 00 55 00 00 01 00 01 0d 03 00 01 00 00 03 0a 02 16
   02 17 02 18 02 16 02 16 02 16 02 16 02 16 02 16 02 17]
```

Outbound SIP Call with Call Agent

Dynamic Switching Media Streams

Call Agent Mode (CAM) in the CSP provides dynamic switching of media streams on or off the CSP RTP channels with a minimal amount of SIP messages.

Inbound Calls

Prior to this feature, the CSP supported bearer on/off switching of Call Agent Mode for inbound calls only. The CSP can connect the inbound leg of a SIP call to a TDM network or to a DSP media service.

This functionality allows the caller to be connected to an operator in a PSTN network. It also allows the application of DSP services such as playing announcements to the calling (inbound) leg.

Bearer switching is based on a coupling and decoupling mechanism. Coupling associates a physical timeslot with a virtual timeslot to enable bearer-switched service. Decoupling dissociates a physical timeslot from a virtual timeslot to enable bearer-free switched service.

The bearer switching takes place whenever required with minimal interaction from the host.

Outbound Calls - New

With this feature, the CSP has the bearer on/off switching capability for outbound call legs too.

This capability is needed when the media services in the CSP (like DSP-2 card) or media transit path (like external TDM IVR or operator) is required for the called party.

> **Example:** A directory assistance operator is located behind a TDM switch. The initial conversation between the operator and called party is bearer-switched through the CSP. Later, the operator drops out of the call flow while connecting the calling party and the called party bypassing the bearer path from the CSP.

Scenarios

This section provides two scenarios that this feature enables.

Connecting Outbound SIP Call to DSP Media Services

The CSP can couple or decouple whenever there is any kind of DSP Service Request or Cancel. The call switches to bearer mode, if it is in bearer free mode, implicitly.

Connecting an Outbound SIP Call to a TDM Call

The CSP can connect to a TDM call (for example ISDN). The call switches to bearer mode implicitly. The host has to explicitly instruct the CSP if coupling is not required (for example Connecting two SIP calls in Call Agent Mode). The host sends a Connect or Connect with Data message to the incoming ISDN channel which connects the incoming ISDN call to the outgoing SIP call leg. The connect message is ACKED if both the calls are in the same mode - either bearer-switched or bearer-free.

To connect an inbound TDM call to the outbound SIP call, the host issues the Connect message to the incoming TDM call. Layer 4 of the outbound SIP call does not know when to couple. Because of this, the Outseize Control message is enhanced to instruct the coupling.

Outseize Control message

The *Outseize Control* (0x002C) is enhanced to support the following two TLVs within the Generic PPL ICB (0x001E):

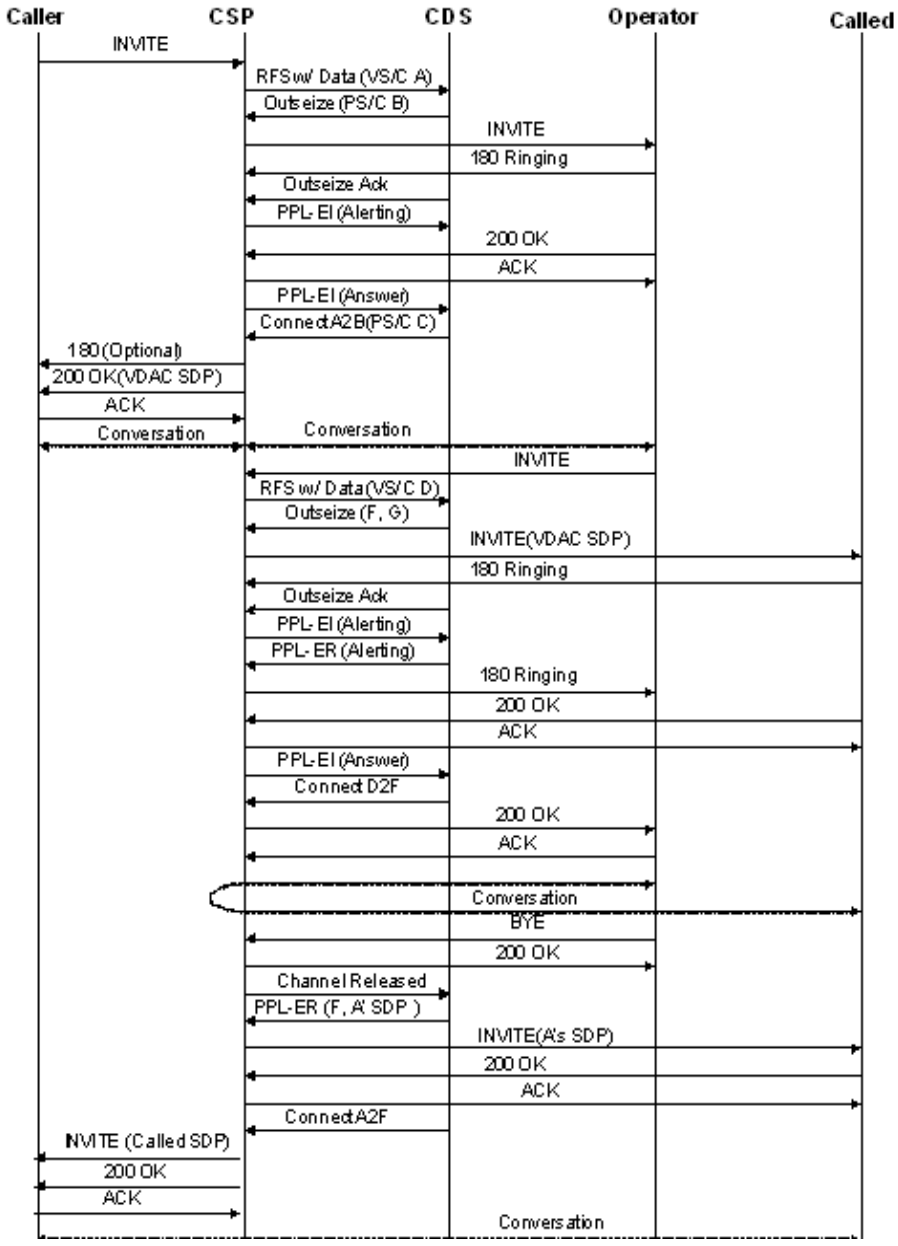
- Channel Service TLV (0x0116)
- Call Agent Physical Channel ID TLV (0x011A)

The host should use the Channel Service TLV (set to 0 for bearer switched service) in the *Outseize Control* message to instruct coupling. If the host wants to provide the physical channel to be used for coupling, the host should use both the Channel Service TLV and the Call Agent Physical Channel ID TLV (includes the physical span/channel).

If the host is provides the physical span/channel, then the host should also provide the complete the NPDI information including the RTP source IP and port address. The *Outseize Control* message can only be used to instruct coupling - not decoupling. Decoupling is triggered when the TDM call connected to the outbound SIP call gets released.

Call Flow

The following call shows an outbound SIP call leg connected to an incoming ISDN call.



API Changes The following API changes support this feature.

PPL Event Request to Generate RE-INVITE message

New PPL Event Request (0x0024)

The new PPL Event Request (0x0024) is used to generate a RE-INVITE message to the SIP endpoint connected to the channel indicated in the channel AIB.

This PPL Event Request will work only if the call is in bearer-free mode or else it will NACK with 0x130C (Call is not in bearer-free mode.)

The new PPL Event Request is added to the SIP UA (0x00A7) component. When the PPL Event Request generates a RE-INVITE message, the 200 OK SIP message received from the other end having the SDP information will be reported to the host using the PPL Event Indication (0x0020) by the SIP UA component.

The data in the PPL Event Indication could be used within the subsequent *Connect with Data* (0x0005) message. The received PPL Event Request is ACKED if the data is valid.

Generic PPL ICB in *Outseize Control* (0x002C) message

The *Outseize Control* (0x002C) message is enhanced to support the following two TLVs within the Generic PPL ICB (0x001E):

- Channel Service TLV (0x0116)
- Call Agent Physical Channel ID TLV (0x011A)

0x001E Generic PPL

ICB Type	0x03 (Extended Data)
ICB ID	0x001E
Data Length (2 bytes)	Variable
Number of TLVs (2bytes)	Variable
Tag	TLV 1 Tag
Length	TLV 1 Length
Value	TLV 1 Value
:	TLV n Tag
:	TLV n Length
:	TLV n Value
TLVs	0x0116 Channel Service 0x011A Call Agent Physical Channel ID

NACK

Value	Description	Corrective User Action
0x130C	Call not in bearer-free mode	Be sure call is in bearer-free mode.

PPL Changes New PPL Event (0x0024)

PPL Event ID	Purpose	Description	AIB Used
0x0024	Generate RE-INVITE	Generates RE-INVITE message to the SIP endpoint connected to that channel. Works only if the call is in bearer-free mode.	0x0D - Channel

Session Description Protocol (SDP) Pass-Through for Call Agent Mode

Overview

The Call Agent Mode (CAM) turns the CSP into a centralized SIP call controller that allows direct flow between the external end-points.

In CAM, the CSP acts as a SIP Back-to-Back User Agent and is not only a full-fledged SIP signaling end-point (User Agent) but also a Session Description Protocol (SDP) signaling endpoint. The CSP originates and terminates SIP and SDP signaling and establishes independent signaling sessions with external endpoints.

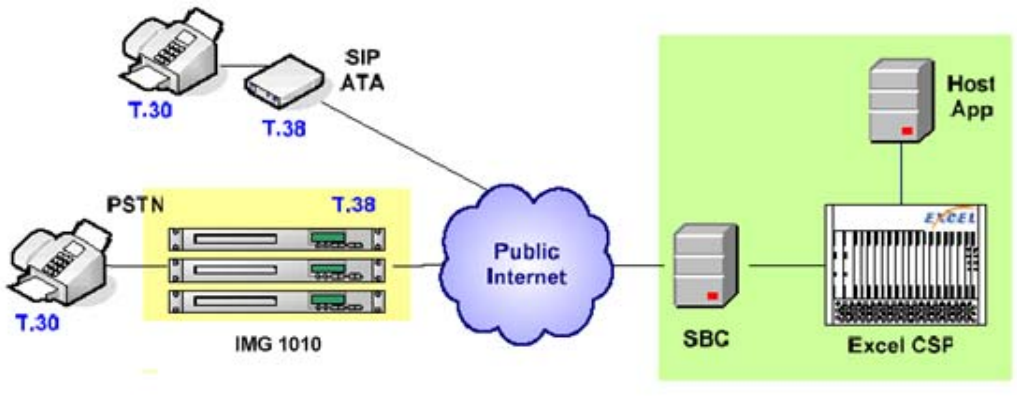
In order to facilitate direct media flow between the external endpoints, the CSP interworks essential inbound SDP parameters into NPDI TLVs to carry this information to the host application. The reverse process occurs for outbound SDP parameters.

Prior to this release, the SDP processing in the CSP could handle only audio media sessions. With this release, the SIP stack is extended to provide more transparency when media capabilities are negotiated between endpoints.

This transparency comes by tunneling an essential portion of raw SDP text between the external endpoints. This approach is flexible because it allows any non-audio media session (not just T.38 fax) to get setup across the CSP. In addition, the endpoint's media capabilities can evolve without affecting the CSP.

Figure Figure 6-3 below shows a network architecture where the external points want to establish a T.38 fax session with each other. A fax call is typically started as an audio session and then switches to T.38 once the IMG or SIP ATA detects the preamble tone. The T.38 parameters are represented in the SDP and conveyed mid-session through the SIP RE-INVITE mechanism.

Figure 6-3 Network Architecture



Description

This feature pertains to only non-audio DSP parameters in the RE-INVITE messages. This feature does not affect the processing of audio SDP parameters and the SDP/NPDI interworking for audio media streams.

Demarcation of SDP

The set of essential SDP parameters is the tunneled portion of the raw SDP text. It is demarcated based on the m=lines in the SDP.

Nested NPDI

The SDP parameters in CAM are structured and communicated as nested NPDI TLVs within the NPDI ICB (0x0033) in the EXS API messages.

There is a TLV, SDP Media Stream Native Text (0x2A17) added within this structure. It will carry the raw SDP text internally in the CSP and between the CSP and the host. Refer to the following for the formats.

Media Streams

An API message can carry up to 1K bytes. Based on this restriction only one media stream worth of information will be tunneled through. For a media stream to be tunneled through, it has to be a non-audio media stream.

Audio and non-audio media streams cannot be present in the SDP at the same time.

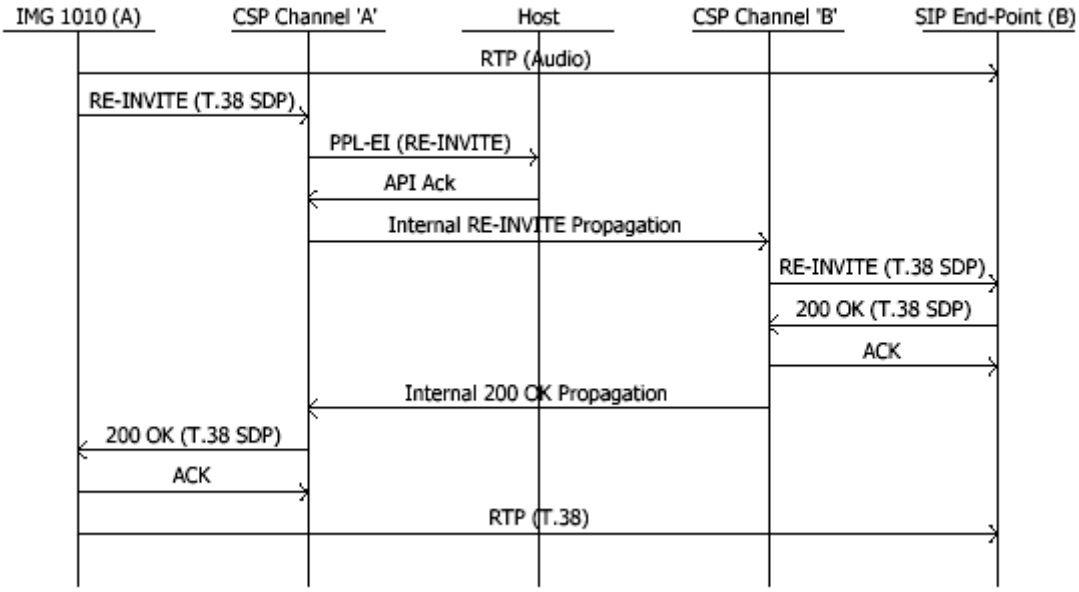
Configuring

This feature is disabled by default. To enable it, set Bit 17, in the SIP Message Information Mask (0x027F) TLV sent in the *VoIP Protocol Configure* (0x00EE) message.

Call Flow and Message Traces

The figure below shows just the portion of a call flow that is relevant to this feature. The steps are described below the call flow.

The IMG 1010 is the Integrated Media Gateway 1010 from Dialogic Technology.



Steps

1. The call flow assumes that an audio session is in place when the terminating gateway (in this example, the IMG 1010 ‘A’) detects the Fax preamble tone. The gateway initiates a RE-INVITE message with an SDP describing its T.38 parameters.
2. The CSP fields this RE-INVITE message on the User Agent session that is already in place with this endpoint.
3. The CSP extracts the raw text including the m=line and lines up to the next m=line or the end of the SDP and pushes it to the host application in a *PPL Event Indication* message.

4. At the same time the CSP internally propagates the RE-INVITE (send by channel A) and sends a RE-INVITE message (with the raw SDP portion reproduced) to the other endpoint.
5. The SIP endpoint B accepts the inbound T.38 SDP offer and responds with an outbound T.38 SDP answer. The 200 OK internally propagates through the signaling and call processing layers in the CSP and is delivered to IMG 1010 ‘A’.

Message Trace -Tunneled Text

As noted in a previous section, a media stream is demarcated based on the m=lines in the SDP. A media stream starts at an m=line and ends at the next m=line, which marks the beginning of the next media stream.

For instance, the underlined portion of text in the message trace below is tunneled through.

```
INVITE sip:+1-650-555-2222@iftgw.there.com SIP/2.0
Via: SIP/2.0/UDP obelix.wcom.com:5060; branch=2d007.1
Via: SIP/2.0/UDP ifax.here.com:5060
From: sip:+1-303-555-1111@ifax.here.com;tag=ab111
To: sip:+1-650-555-2222@obelix.wcom.com
Call-ID: 1717@ifax.here.com
CSeq: 17 INVITE
Contact: <sip:+1-303-555-1111@ifax.here.com>
Content-Type: application/sdp
Content-Length: 320
v=0
o=ifaxgw1 2890846527 2890846527 IN IP4 ifax.here.com
s=Session SDP
c=IN IP4 ifaxmg.here.com
t=0 0
m=image 15002 udptl t38
a=T38FaxVersion:0
a=T38MaxBitRate:14400
a=T38FaxFillBitRemoval:0
a=T38FaxTranscodingMMR:0
a=T38FaxTranscodingJBIG:0
a=T38FaxRateManagement:transferredTCF
a=T38FaxMaxBuffer:72
a=T38FaxMaxDatagram:316
a=T38FaxUdpEC:t38UDPFEC
a=T38FaxUdpEC:t38UDPRedundancy
```

Nested TLV Structure

The SDP Native Media Stream TLV (0x2A17) is contained inside the nested NPDI TLV structure used in the Call Agent Mode to propagate SDP information from one endpoint to another.

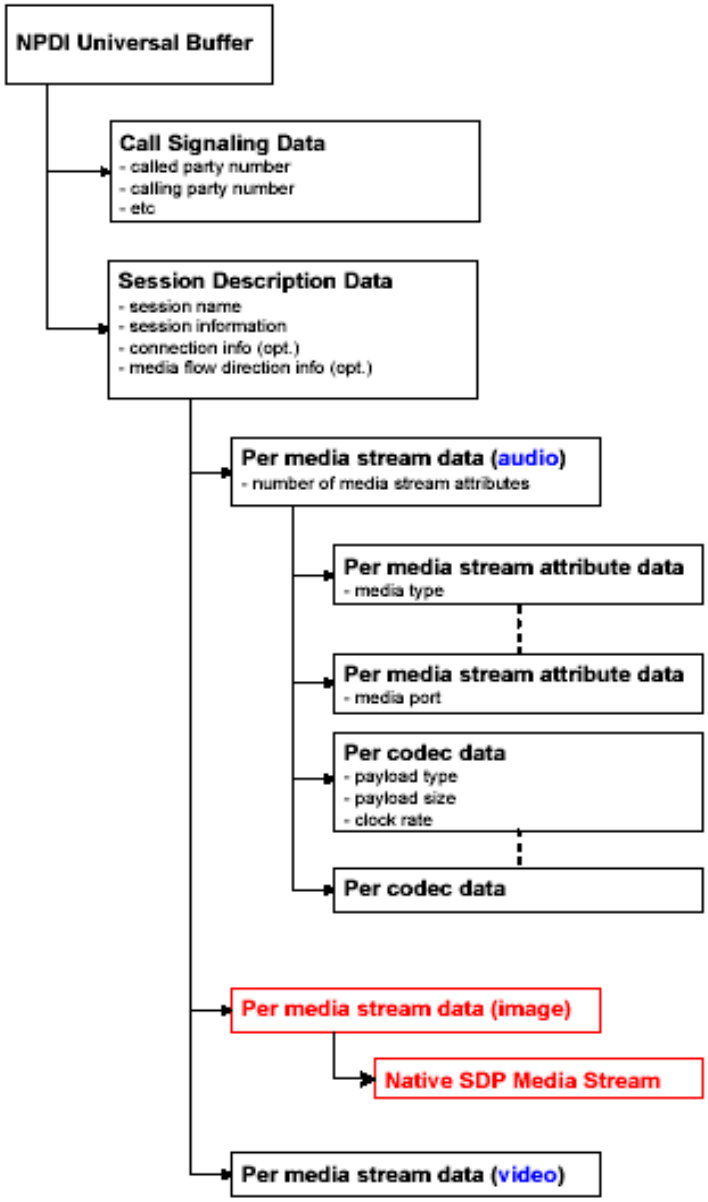
This TLV is analogous to an “audio” media stream. The only difference being the CSP SDP stack parses the audio media stream and interworks those parameters into NPDI TLVs and sends that processes information to the host.

This TLV carries information that is similar in semantics but different in syntax because it carries raw SDP text.

The nested NPDI structure treats a media stream regardless of its type (for example, audio, video, image, whiteboard, application, data) therefore the DSP Native Media Stream TLV is contained inside the NPDI Per-Media Stream TLV (0x2A01).

Figure Figure 6-4 shows the layout of the nested NPDI structure.

Figure 6-4 Native SDP Media Stream TLV within Nested NPDI



Message Trace -RE-INVITE with T.38 Parameters

The following message trace shows a SIP RE-INVITE message with T.38 fax parameters and the corresponding PPL Event Indication message from the CSP to the host.

```
INVITE sip:service@192.168.1.200:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.105:5062
From: sipp <sip:sipp@192.168.1.105:5062>;tag=1
To: sut <sip:service@192.168.1.200:5060>;tag=891265
Call-ID: 1.17832.192.168.1.105@sipp.call.id
Cseq: 2 INVITE
Contact: sip:sipp@192.168.1.105:5062
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 320
v=0
o=faxgw1 2890844527 171090 IN IP4 iftgw.there.com
s=Session SDP
c=IN IP4 192.168.100.100
t=0 0
m=image 49172 udptl t38
a=T38FaxVersion:0
a=T38MaxBitRate:9600
a=T38FaxFillBitRemoval:0
a=T38FaxTranscodingMMR:0
a=T38FaxTranscodingJBIG:0
a=T38FaxRateManagement:transferredTCF
a=T38FaxMaxBuffer:72
a=T38FaxMaxDatagram:316
a=T38FaxUdpEC:t38UDPFEC
a=T38FaxUdpEC:t38UDPRedundancy

X->H
[01 4c 00 43 00 0a ff 00 01 0d 03 00 07 06 00 a7 00 1e 01 03
00 33 01 36 00 01 29 ff 01 30 2a 0e 00 04 c0 a8 64 64
2a 01 01 24 2a 17 01 20 6d 3d 69 6d 61 67 65 20 34 39
31 37 32 20 75 64 70 74 6c 20 74 33 38 0d 0a 61 3d 54
33 38 46 61 78 56 65 72 73 69 6f 6e 3a 30 0d 0a 61 3d
54 33 38 4d 61 78 42 69 74 52 61 74 65 3a 39 36 30 30
0d 0a 61 3d 54 33 38 46 61 78 46 69 6c 6c 42 69 74 52
65 6d 6f 76 61 6c 3a 30 0d 0a 61 3d 54 33 38 46 61 78
54 72 61 6e 73 63 6f 64 69 6e 67 4d 4d 52 3a 30 0d 0a
61 3d 54 33 38 46 61 78 54 72 61 6e 73 63 6f 64 69 6e
67 4a 42 49 47 3a 30 0d 0a 61 3d 54 33 38 46 61 78 52
61 74 65 4d 61 6e 61 67 65 6d 65 6e 74 3a 74 72 61 6e
73 66 65 72 72 65 64 54 43 46 0d 0a 61 3d 54 33 38 46
61 78 4d 61 78 42 75 66 66 65 72 3a 37 32 0d 0a 61 3d
54 33 38 46 61 78 4d 61 78 44 61 74 61 67 72 61 6d 3a
33 31 36 0d 0a 61 3d 54 33 38 46 61 78 55 64 70 45 43
3a 74 33 38 55 44 50 46 45 43 0d 0a 61 3d 54 33 38 46
61 78 55 64 70 45 43 3a 74 33 38 55 44 50 52 65 64 75
6e 64 61 6e 63 79 0d 0a]
```

Support REFER Request in Call Agent

Purpose

This feature allows the CSP to process an inbound REFER request and to generate an outbound REFER request in Call Agent Mode.

This section contains the following:

- Description of the feature
- Sample calls flows
- API and PPL changes to support the feature

Description

Whenever there is an incoming REFER request, the REFER-TO header is reported to the host in the PPL Event Indication (Event 0x0021; Component 0x00A7). The host can propagate the REFER request to the outbound leg using the *PPL Event Request* message (Event 0x0027; Component 0x00A7).

The status of the reference is reported to the SIP component in the NOTIFY messages. This status is reported to the host in the PPL Event Indication (Event 0x0028; Component 0x00A7).

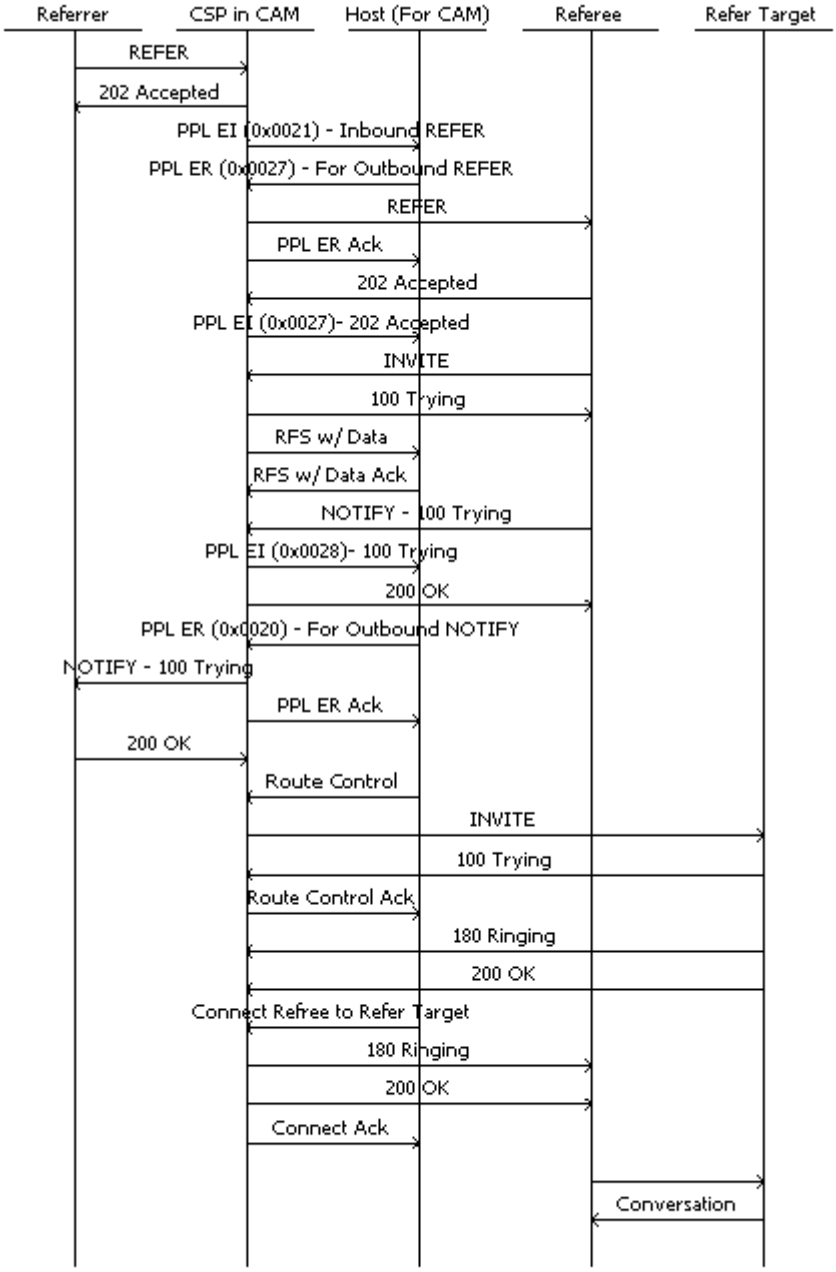
Call Flows

The following call flows show three scenarios:

- Success REFER request and NOTIFY message in Call Agent Mode (Includes sample API and SIP message traces.)
- REFER Request Failure in Call Agent Mode
- Reference Failure in Call Agent Mode

Successful REFER and NOTIFY in Call Agent Mode

This call flow assumes that the Referrer and Referee are in the answered state. The second PPL Event Request (0x0020) to generate the NOTIFY message is not supported as of now, but will be supported in a future release.



API Messages

```
X->H
    [00 35 00 43 00 27 04 00 01 0d 03 00 64 08 00 a7 00 21
   01 03
     00 33 00 1f 00 02 29 19 00 0b 39 39 39 39 39 39 39 39
     30 31 00 29 1b 00 0a 31 30 2e 31 30 2e 31 2e 32 00]

H->X
    [00 05 00 43 00 27 04]
```

```
H->X
    [00 05 00 43 00 27 04]

H->X
    [00 3d 00 44 00 01 04 00 01 0d 03 00 64 09 00 a7 00 27
     01 03 00 33 00 27 00 03 29 1e 00 0b 39 39 39 39 39 39
     39 39 30 31 00 29 20 00 0a 31 30 2e 31 30 2e 31 2e 32
     00 29 21 00 04 00 00 13 ff]

X->H
    [00 07 00 44 00 01 04 00 10]

X->H
    [00 1e 00 43 00 28 04 00 01 0d 03 00 64 09 00 a7 00 27
   01 03
     00 33 00 08 00 01 29 15 00 02 00 ca]

H->X
    [00 05 00 43 00 28 04]

H->X
    [00 05 00 43 00 28 04]

X->H
    [00 1e 00 43 00 29 04 00 01 0d 03 00 64 09 00 a7 00 28
   01 03
     00 33 00 08 00 01 29 15 00 02 00 64]

H->X
    [00 05 00 43 00 29 04]

H->X
    [00 05 00 43 00 29 04]

H->X
    [00 1e 00 44 00 01 04 00 01 0d 03 00 64 08 00 a7 00 20
     01 03 00 33 00 08 00 01 29 4b 00 02 00 64]

X->H
    [00 07 00 44 00 01 04 00 10]
```

SIP Trace

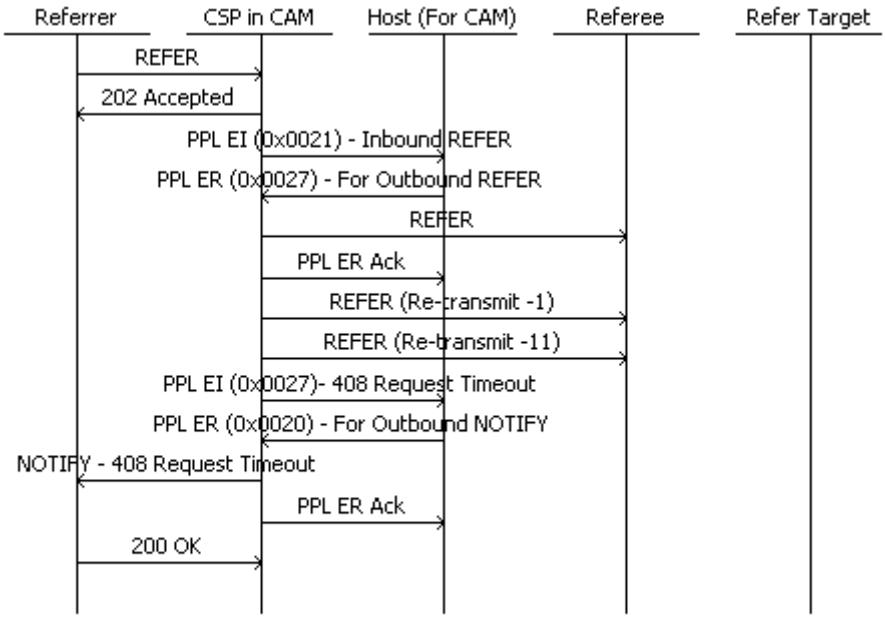
```
1 -RECEIVED From 10.10.1.2:5060 at 12425
2 -RECEIVED From 10.10.1.47:1024 at 12432
REFER sip:12345@10.10.1.133:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
From: 12345<sip:12345@10.10.1.47:5060>;tag=719188306d
To: 12345<sip:12345@10.10.1.133:5060>;tag=91883079
Call-ID: EXCEL-CSP27.47.12397.140@10.10.1.47
CSeq: 2 REFER
```

```
Max-Forwards: 70
Contact: 12345<sip:12345@10.10.1.47:5060>
Refer-To: sip:9999999901@10.10.1.2:5119
User-Agent: Excel_CSP/83.10.69
Content-Length: 0


3 -SENT To 10.10.1.47:5060 at 12432
SIP/2.0 202 Accepted
To: 12345<sip:12345@10.10.1.133:5060>;tag=91883079
From: 12345<sip:12345@10.10.1.47:5060>;tag=719188306d
Call-ID: EXCEL-CSP27.47.12397.140@10.10.1.47
CSeq: 2 REFER
Contact: 12345<sip:12345@10.10.1.65:5060>
Via: SIP/2.0/UDP 10.10.1.47
User-Agent: Excel_CSP/83.10.69
Content-Length: 0


4 -RECEIVED From 10.10.1.2:5060 at 12436


5 -SENT To 10.10.1.133:5060 at 12436
REFER sip:12345@10.10.1.133:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.65
From: 12345<sip:12345@10.10.1.47:5060>;tag=410511353079
To: 12345<sip:12345@10.10.1.133:5060>;tag=113559f
Call-ID: EXCEL-CSP4.1009.12409.900@10.10.1.65
CSeq: 2 REFER
Max-Forwards: 70
Contact: 12345<sip:12345@10.10.1.65:5060>
Refer-To: sip:9999999901@10.10.1.2:5119
User-Agent: Excel_CSP/83.10.69
Content-Length: 0


6 -RECEIVED From 10.10.1.133:1024 at 12436
SIP/2.0 202 Accepted
To: 12345<sip:12345@10.10.1.133:5060>;tag=113559f
From: 12345<sip:12345@10.10.1.47:5060>;tag=410511353079
Call-ID: EXCEL-CSP4.1009.12409.900@10.10.1.65
CSeq: 2 REFER
Contact: 12345<sip:12345@10.10.1.133:5060>
Via: SIP/2.0/UDP 10.10.1.65
User-Agent: Excel_CSP/83.10.69
Content-Length: 0


7 -RECEIVED From 10.10.1.133:1024 at 12442
NOTIFY sip:12345@10.10.1.65:5060 SIP/2.0
```

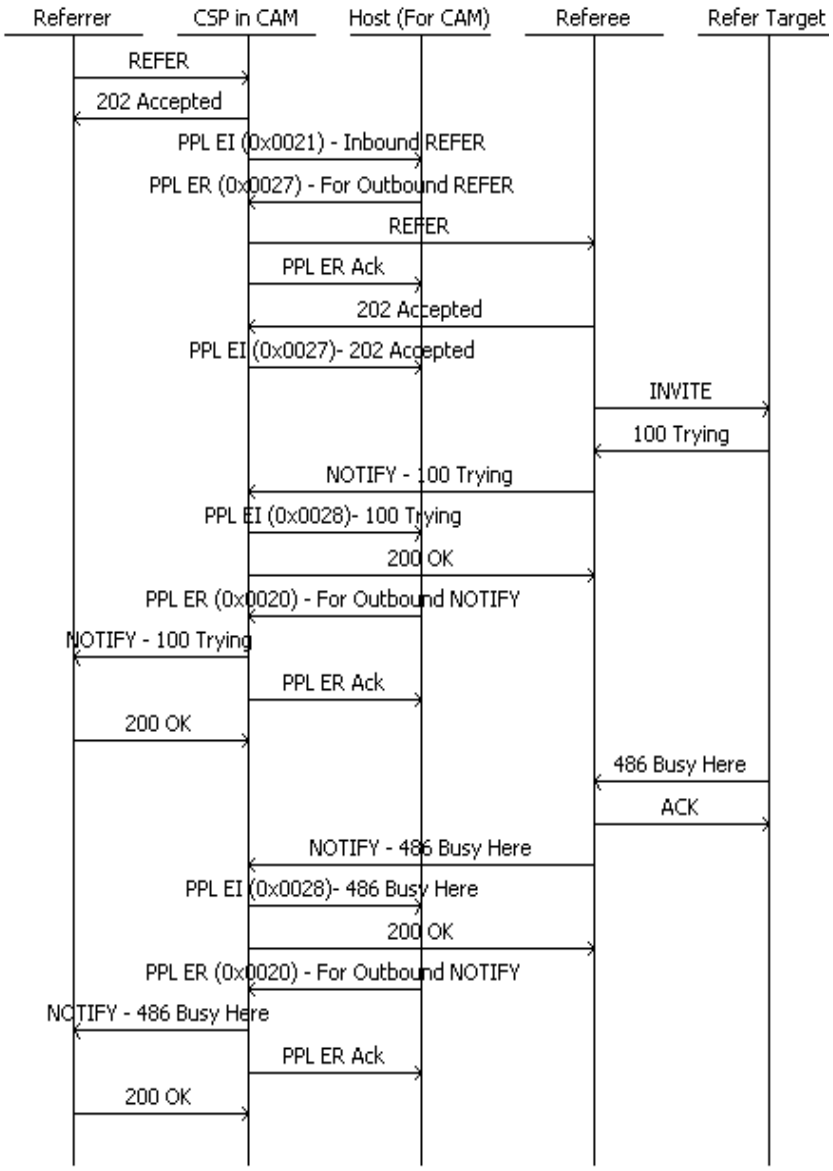
```
Via: SIP/2.0/UDP 10.10.1.133
To: 12345<sip:12345@10.10.1.47:5060>;tag=410511353079
From: 12345<sip:12345@10.10.1.133:5060>;tag=113559f
Call-ID: EXCEL-CSP4.1009.12409.900@10.10.1.65
CSeq: 3 NOTIFY
Event: refer
Contact: 12345<sip:12345@10.10.1.133:5060>
Content-Type: message/sipfrag;version=2.0
Content-Length: 20

SIP/2.0 100 Trying

8 -SENT To 10.10.1.133:5060 at 12442
SIP/2.0 200 OK
To: 12345<sip:12345@10.10.1.47:5060>;tag=410511353079
From: 12345<sip:12345@10.10.1.133:5060>;tag=113559f
Call-ID: EXCEL-CSP4.1009.12409.900@10.10.1.65
CSeq: 3 NOTIFY
Via: SIP/2.0/UDP 10.10.1.133
User-Agent: Excel_CSP/83.10.69
Content-Length: 0


9 -RECEIVED From 10.10.1.2:5060 at 12446


10-RECEIVED From 10.10.1.2:5060 at 12456


11-RECEIVED From 10.10.1.2:5060 at 12466


12-RECEIVED From 10.10.1.2:5060 at 12477


13-SENT To 10.10.1.47:5060 at 12478
NOTIFY sip:12345@10.10.1.47:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.65
To: 12345<sip:12345@10.10.1.47:5060>;tag=719188306d
From: 12345<sip:12345@10.10.1.133:5060>;tag=91883079
Call-ID: EXCEL-CSP27.47.12397.140@10.10.1.47
CSeq: 3 NOTIFY
Event: refer
Contact: 12345<sip:12345@10.10.1.65:5060>
Content-Type: message/sipfrag;version=2.0
Content-Length: 20

SIP/2.0 100 Trying

14-RECEIVED From 10.10.1.47:1024 at 12478
```

```
SIP/2.0 200 OK
To: 12345<sip:12345@10.10.1.47:5060>;tag=719188306d
From: 12345<sip:12345@10.10.1.133:5060>;tag=91883079
Call-ID: EXCEL-CSP27.47.12397.140@10.10.1.47
CSeq: 3 NOTIFY
Via: SIP/2.0/UDP 10.10.1.65
User-Agent: Excel_CSP/83.10.69
Content-Length: 0


15-RECEIVED From 10.10.1.2:5060 at 12487
```

REFER Failure in Call Agent Mode



Reference Failure in Call Agent Mode



API Changes The section provides the API changes that support this feature.

The *PPL Event Request* (0x0044) message can include the following four new TLVs:

- 0x291ENPDI SIP Refer to Username
- 0x291F NPDI SIP Refer to Password (optional)
- 0x2920 NPDI SIP Refer to Host Name
- 0x2921 NPDI SIP Refer to Port (optional)

The *PPL Event Indication* (0x0043) message includes the NPDI SIP Response Code TLV - 0x2915 as follows:

PPL Event Indication (0x0043)

MESSAGE (White)		RESPONSE (Gray)	
Byte	**Field Description**	**Byte**	**Field Description**
0	Frame (0xFE)	0	Frame (0xFE)
1, 2	Length (0x00NN)	1, 2	Length (0x0005)
3, 4	Message Type (0x0043)	3, 4	Message Type (0x0043)
5	Reserved (0x00)	5	Reserved (0x00)
6	Sequence Number	6	Same Sequence Number
7	Logical Node ID	7	Logical Node ID
:	AIB Channel (0x0D)	8	Checksum
:	PPL Component ID - 0x00A7		
:	PPL Event 0x0027/0x0028		
:	Number of ICBs to follow - 0x01		
:	ICB Type (0x03) - Extended Data ICB		
:	ICB ID - NPDI Universal ICB - 0x0033)		
:	Data Length - 0x0008		
:	Number of TLVs - 0x0001		
:	NPDI SIP Response Code TLV -0x2915 (See format below.)		
:	Checksum		

0x2915 NPDI SIP Response Code

The PPL Event Indications now supports this existing TLV within the NPDI Universal ICB (0x0033).

This TLV provides the response code received for any SIP message generated by the CSP.

Used in:
0x0033 NPDI Universal ICB in:
Channel Released with Data message
Release Channel with Data message
PPL Event Indication message
Request for Service with Data message

Byte	Description	
0, 1	Tag	0x2915
2, 3	Length	0x0002
4, 5	Value	SIP Response Code (2 Bytes) This TLV, when reported in the PPL Event Indication for Event (0x0027/0x0028), will have any valid SIP Response value. The response code to be interpreted for event 0x0027 is as follows: 202 - Accepted (The REFER request is successful) 405 – Method Not Allowed (REFER request failed) 408 – Request Timed Out (REFER request failed) The response code to be interpreted for event 0x0028 is as follows: 100 – Trying 200 - OK (The reference is successful) 503 - Service Unavailable (The reference failed) 603 - Decline (The reference failed)

PPL Changes

New Events

The following new PPL Events for the SIP UA PPL Component (0x00A7).

Event Request

0x0027 - Allows the CSP to generate an outbound REFER request with the refer target URL filling in the TLV.

Event Indications

0x0027 - Informs the host about the result of the sent REFER request (Success - 2xx or Failure - 4xx-6xx).

0x0028 - Informs the host that sent the REFER request about the status of the reference (Trying, Success, or Failure). The value in the SIP Response Code TLV (0x2915) reports the exact response code.

PPL Event Request for RE-INVITE Message

Overview

This section includes a description of the feature, call flows, and the PPL changes to support the feature.

Description

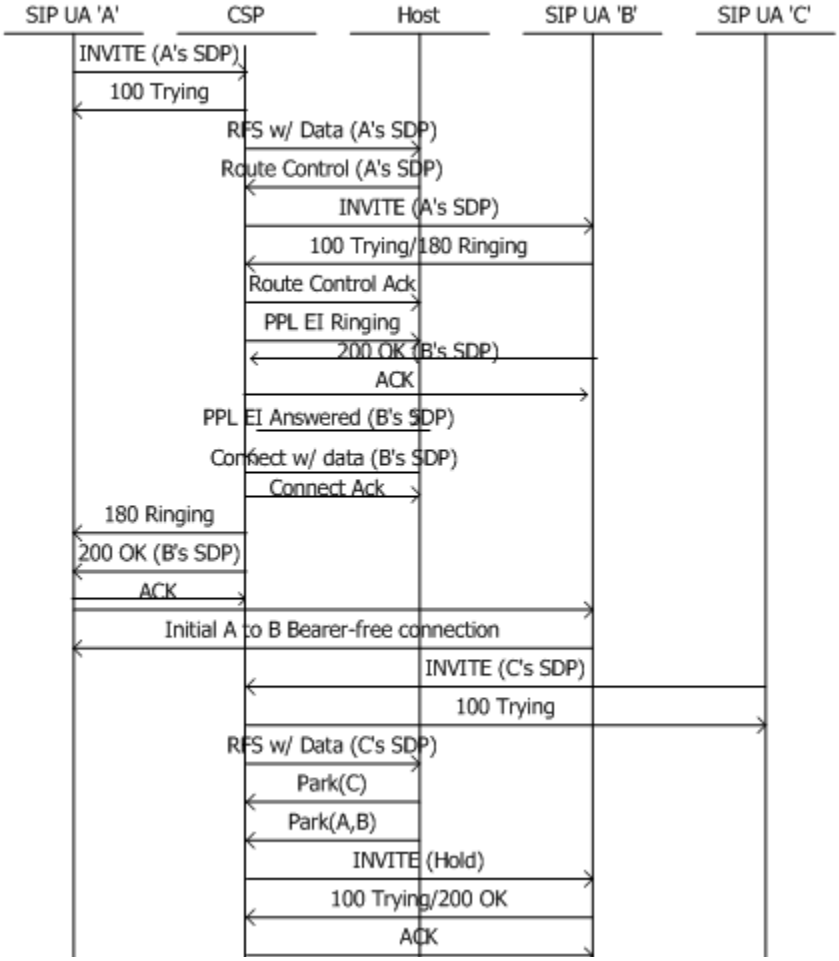
This feature allows the CSP to support the RE-INVITE message for a Call Agent Mode call in bearer-free mode using PPL Event Request (0x0024). This PPL Event Request generates a Re-INVITE regardless of the call direction: incoming or outgoing.

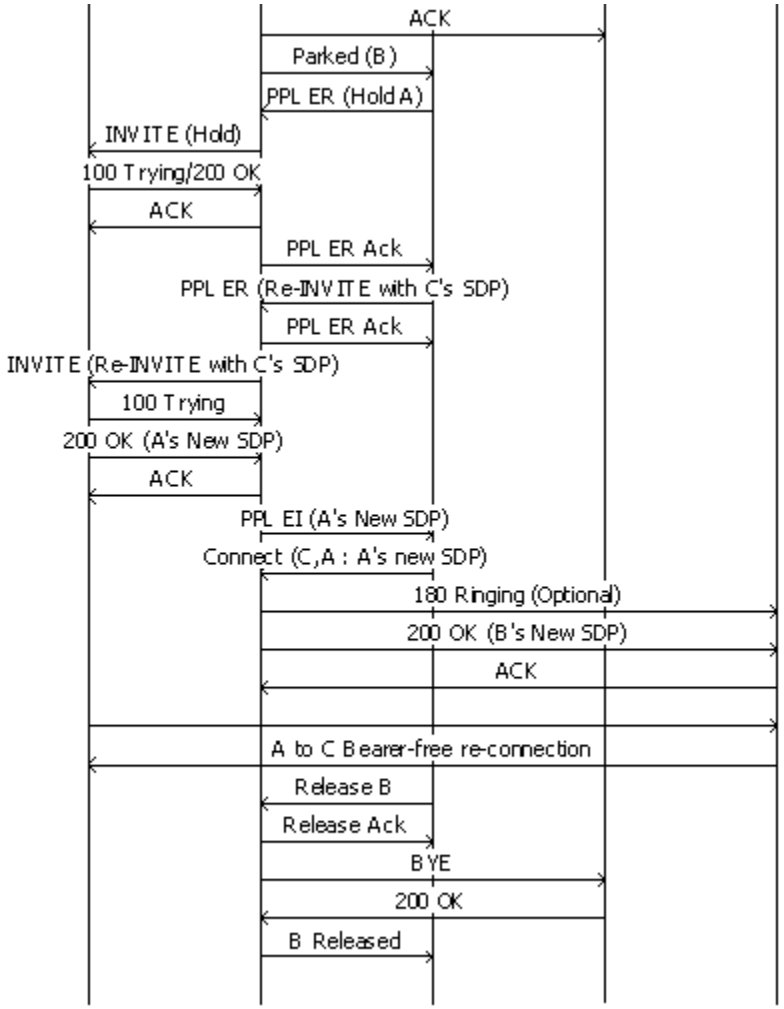
This PPL Event Request will work only if the call is in bearer-free mode or else it will NACK with a value 0x130C (Call is not in bearer-free mode.)

Call Flows

Third Party Connected to Called Party

The call flow below shows a scenario where the initial conversation between SIP endpoints (A/B) is put on hold and a third party (C) is connected to the called party of the initial conversation (B). The call flow begins below and continues on the next page.


SIP UA 'A'
CSP
Host
SIP UA 'B'
SIP UA 'C'
INVITE (A's SDP)
100 Trying
RFS w/ Data (A's SDP)
Route Control (A's SDP)
INVITE (A's SDP)
100 Trying/180 Ringing
Route Control Ack
PPL EI Ringing
200 OK (B's SDP)
ACK
PPL EI Answered (B's SDP)
Connect w/ data (B's SDP)
Connect Ack
180 Ringing
200 OK (B's SDP)
ACK
Initial A to B Bearer-free connection
INVITE (C's SDP)
100 Trying
RFS w/ Data (C's SDP)
Park(C)
Park(A,B)
INVITE (Hold)
100 Trying/200 OK
ACK



API Message

```
X->H
    [01 1d 00 2d 00 08 0c 00 01 0d 03 00 64 0c 00 33 01 03
   00 33
     01 09 00 10 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 06 33 35 37 35 31 00 29 1b 00 0b 31 30 2e 31 30
     2e 31 2e 31 34 00 29 1c 00 04 00 00 13 c4 29 23 00 06
     32 30 35 39 37 00 29 25 00 0a 31 30 2e 31 30 2e 31 2e
     32 00 29 26 00 04 00 00 13 c4 29 2d 00 06 32 30 35 39
     37 00 29 2f 00 0a 31 30 2e 31 30 2e 31 2e 32 00 29 30
     00 04 00 00 13 c4 29 33 00 01 01 27 18 00 08 02 00 00
     00 05 20 59 70 27 17 00 06 02 00 05 35 75 10 29 ff 00
     75 2a 0e 00 04 0a 0a 01 02 2a 01 00 69 2a 03 00 01 00
     2a 07 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 02
```

```
     2a 09 00 01 02 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a
     08 00 02 00 01 2a 09 00 01 01 2a 0b 00 04 00 00 1f 40
     2a 02 00 13 2a 08 00 02 00 16 2a 09 00 01 16 2a 0b 00
     04 00 00 1f 40 2a 02 00 13 2a 08 00 02 01 65 2a 09 00
     01 32 2a 0b 00 04 00 00 1f 40 29 16 00 01 01]

H->X
    [00 0c 00 2d 00 08 0c 00 01 0d 03 00 64 0c]

H->X
    [00 0c 00 2d 00 08 0c 00 01 0d 03 00 64 0c]

H->X
    [01 40 00 e8 00 00 0c 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 71 00 0f 00
     01 0b 00 71 00 02 00 00 03 00 33 01 11 00 11 27 4e 00
     02 00 05 27 7e 00 03 08 00 00 29 19 00 06 33 35 37 35
     31 00 29 1b 00 0b 31 30 2e 31 30 2e 31 2e 31 34 00 29
     1c 00 04 00 00 13 c4 29 23 00 06 32 30 35 39 37 00 29
     25 00 0a 31 30 2e 31 30 2e 31 2e 32 00 29 26 00 04 00
     00 13 c4 29 2d 00 06 32 30 35 39 37 00 29 2f 00 0a 31
     30 2e 31 30 2e 31 2e 32 00 29 30 00 04 00 00 13 c4 29
     33 00 01 01 27 18 00 08 02 00 00 00 05 20 59 70 27 17
     00 06 02 00 05 15 75 80 2a 00 00 75 2a 0e 00 04 0a 0a
     01 02 2a 01 00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f
     40 2a 02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b
     00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09
     00 01 01 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00
     02 00 16 2a 09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02
     00 13 2a 08 00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00
     00 1f 40 29 16 00 01 01 2a 0e 00 04 0a 0a 01 02]

X->H
    [00 14 00 e8 00 00 0c 00 10 01 02 1e 09 00 01 00 39 00
   03 00
     64 0d]

X->H
    [00 11 00 43 00 0a 0c 00 01 0d 03 00 64 0d 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 0a 0c]

H->X
    [00 05 00 43 00 0a 0c]

X->H
    [00 91 00 43 00 0b 0c 00 01 0d 03 00 64 0d 00 a7 00 20
   01 03
```

```
     00 33 00 7b 00 01 29 ff 00 75 2a 0e 00 04 0a 0a 01 0e
     2a 01 00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f 40 2a
     02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04
     00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01
     01 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00
     16 2a 09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02 00 13
     2a 08 00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00 00 1f
     40]

H->X
    [00 05 00 43 00 0b 0c]

H->X
    [00 05 00 43 00 0b 0c]

H->X
    [00 9f 00 05 00 00 0c 00 02 0d 03 00 64 0c 0d 03 00 64
     0d 01 02 03 00 1e 00 07 00 01 01 16 00 01 01 03 00 33
     00 7b 00 01 2a 00 00 75 2a 0e 00 04 0a 0a 01 0e 2a 01
     00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f 42 2a 02 00
     13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04 00 00
     1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01 01 2a
     0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 16 2a
     09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08
     00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00 00 1f 40]

X->H
    [00 07 00 05 00 00 0c 00 10]

X->H
    [01 1b 00 2d 00 09 0c 00 01 0d 03 00 64 0e 00 33 01 03
   00 33
     01 07 00 10 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 06 32 30 35 39 37 00 29 1b 00 0a 31 30 2e 31 30
     2e 31 2e 32 00 29 1c 00 04 00 00 13 c4 29 23 00 05 38
     33 32 33 00 29 25 00 0b 31 30 2e 31 30 2e 31 2e 32 39
     00 29 26 00 04 00 00 13 c4 29 2d 00 05 38 33 32 33 00
     29 2f 00 0b 31 30 2e 31 30 2e 31 2e 32 39 00 29 30 00
     04 00 00 13 c4 29 33 00 01 01 27 18 00 07 02 00 00 00
     04 83 23 27 17 00 06 02 00 05 20 59 70 29 ff 00 75 2a
     0e 00 04 0a 0a 01 1d 2a 01 00 69 2a 03 00 01 00 2a 07
     00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 02 2a 09
     00 01 02 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00
     02 00 01 2a 09 00 01 01 2a 0b 00 04 00 00 1f 40 2a 02
     00 13 2a 08 00 02 00 16 2a 09 00 01 16 2a 0b 00 04 00
     00 1f 40 2a 02 00 13 2a 08 00 02 01 65 2a 09 00 01 32
     2a 0b 00 04 00 00 1f 40 29 16 00 01 01]

H->X
    [00 0c 00 2d 00 09 0c 00 01 0d 03 00 64 0e]
```

```
H->X
    [00 0c 00 2d 00 09 0c 00 01 0d 03 00 64 0e]

H->X
    [00 11 00 bf 00 00 0c 00 02 0d 03 00 64 0e 0d 03 00 64
     0e]

X->H
    [00 07 00 bf 00 00 0c 00 10]

H->X
    [00 11 00 bf 00 00 0c 00 02 0d 03 00 64 0c 0d 03 00 64
     0d]

X->H
    [00 07 00 bf 00 00 0c 00 10]

X->H
    [00 0e 00 42 00 64 0c 00 01 0d 03 00 64 0d 04 00]

H->X
    [00 1d 00 44 00 00 0c 00 01 0d 03 00 64 0c 00 a7 00 1e
     01 03 00 33 00 07 00 01 27 b3 00 01 03]

H->X
    [00 05 00 42 00 64 0c]

H->X
    [00 05 00 42 00 64 0c]

X->H
    [00 07 00 44 00 00 0c 00 10]

H->X
    [00 91 00 44 00 00 0c 00 01 0d 03 00 64 0c 00 a7 00 24
     01 03 00 33 00 7b 00 01 2a 00 00 75 2a 0e 00 04 0a 0a
     01 1d 2a 01 00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f
     40 2a 02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b
     00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09
     00 01 01 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00
     02 00 16 2a 09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02
     00 13 2a 08 00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00
     00 1f 40]

X->H
    [00 07 00 44 00 00 0c 00 10]

X->H
```

```
    [00 91 00 43 00 0c 0c 00 01 0d 03 00 64 0c 00 a7 00 20
   01 03
     00 33 00 7b 00 01 29 ff 00 75 2a 0e 00 04 0a 0a 01 02
     2a 01 00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f 40 2a
     02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04
     00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01
     01 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00
     16 2a 09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02 00 13
     2a 08 00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00 00 1f
     40]

H->X
    [00 05 00 43 00 0c 0c]

H->X
    [00 05 00 43 00 0c 0c]

H->X
    [00 9f 00 05 00 00 0c 00 02 0d 03 00 64 0e 0d 03 00 64
     0c 01 02 03 00 1e 00 07 00 01 01 16 00 01 01 03 00 33
     00 7b 00 01 2a 00 00 75 2a 0e 00 04 0a 0a 01 02 2a 01
     00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f 40 2a 02 00
     13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04 00 00
     1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01 01 2a
     0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 16 2a
     09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08
     00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00 00 1f 40]

X->H
    [00 07 00 05 00 00 0c 00 10]

H->X
    [00 11 00 08 00 00 0c 00 02 0d 03 00 64 0d 0d 03 00 64
     0d]

X->H
    [00 07 00 08 00 00 0c 00 10]

X->H
    [00 0c 00 49 00 0a 0c 00 01 0d 03 00 64 0d]

H->X
    [00 05 00 49 00 0a 0c]
```

```
2
 Printing all SIP messages

1 -RECEIVED From 10.10.1.2:5060 at 15995
INVITE sip:35751@10.10.1.14 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bK1FDA7D2876B04953942
   90D381FB3
389C
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
To: <sip:35751@10.10.1.14>
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 INVITE
Max-Forwards: 70
Content-Type: application/sdp
User-Agent: X-Lite release 1103m
Content-Length: 290

v=0
o=20597 722096328 722096343 IN IP4 10.10.1.2
s=X-Lite
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

2 -SENT To 10.10.1.2:5060 at 15995
SIP/2.0 100 Trying
To: <sip:35751@10.10.1.14>;tag=45103e7b
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 INVITE
Contact: 35751<sip:35751@10.10.1.47:5060>
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bK1FDA7D2876B04953942
   90D381FB3
389C
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


3 -SENT To 10.10.1.14:5060 at 15996
```

```
INVITE sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751<sip:35751@10.10.1.14:5060>
From: 20597<sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
Contact: 20597<sip:20597@10.10.1.47:5060>
User-Agent: Excel_CSP/83.10.75
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 223

v=0
o=sip 1122578382 1122578382 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

4 -RECEIVED From 10.10.1.14:5060 at 15996
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


5 -RECEIVED From 10.10.1.14:5060 at 15996
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


6 -RECEIVED From 10.10.1.2:5060 at 15998
```

```
7 -RECEIVED From 10.10.1.14:5060 at 15998


8 -RECEIVED From 10.10.1.29:5060 at 15999


9 -RECEIVED From 10.10.1.14:5060 at 16004
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 1 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 292

v=0
o=35751 292794390 292802031 IN IP4 10.10.1.14
s=X-Lite
c=IN IP4 10.10.1.14
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

10-SENT To 10.10.1.14:5060 at 16004
ACK sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751<sip:35751@10.10.1.14:5060>;tag=4107074591
From: 20597<sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 1 ACK
Content-Length: 0


11-RECEIVED From 10.10.1.2:5060 at 16008


12-RECEIVED From 10.10.1.14:5060 at 16008
```

```
13-SENT To 10.10.1.2:5060 at 16009
SIP/2.0 180 Ringing
To: <sip:35751@10.10.1.14>;tag=45103e7b
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 INVITE
Contact: 35751<sip:35751@10.10.1.47:5060>
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bK1FDA7D2876B04953942
   90D381FB3
389C
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


14-SENT To 10.10.1.2:5060 at 16009
SIP/2.0 200 OK
To: <sip:35751@10.10.1.14>;tag=45103e7b
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 INVITE
Contact: 35751<sip:35751@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bK1FDA7D2876B04953942
   90D381FB3
389C
User-Agent: Excel_CSP/83.10.75
Content-Type: application/sdp
Content-Length: 224

v=0
o=sip 1122578395 1122578395 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 10.10.1.14
t=0 0
m=audio 8002 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

15-RECEIVED From 10.10.1.2:5060 at 16009
ACK sip:35751@10.10.1.47:5060 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bKDC857736F60944AEB97
   C9B1D843C
7181
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
To: <sip:35751@10.10.1.14>;tag=45103e7b
```

```
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 ACK
Max-Forwards: 70
Content-Length: 0


16-RECEIVED From 10.10.1.29:5060 at 16010


17-RECEIVED From 10.10.1.29:5060 at 16016
INVITE sip:20597@10.10.1.2 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK2017BD3E5D074025B2
   7B107065B
31D99
From: sathia <sip:8323@10.10.1.29>;tag=851025480
To: <sip:20597@10.10.1.2>
Contact: <sip:8323@10.10.1.29:5060>
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 INVITE
Max-Forwards: 70
Content-Type: application/sdp
User-Agent: X-Lite release 1103m
Content-Length: 293

v=0
o=8323 1402407484 1402407500 IN IP4 10.10.1.29
s=X-Lite
c=IN IP4 10.10.1.29
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

18-SENT To 10.10.1.29:5060 at 16016
SIP/2.0 100 Trying
To: <sip:20597@10.10.1.2>;tag=33963e90
From: sathia <sip:8323@10.10.1.29>;tag=851025480
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 INVITE
Contact: 20597<sip:20597@10.10.1.47:5060>
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK2017BD3E5D074025B2
   7B107065B
```

```
31D99
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


19-SENT To 10.10.1.29:5060 at 16017
SIP/2.0 180 Ringing
To: <sip:20597@10.10.1.2>;tag=33963e90
From: sathia <sip:8323@10.10.1.29>;tag=851025480
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 INVITE
Contact: 20597<sip:20597@10.10.1.47:5060>
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK2017BD3E5D074025B2
   7B107065B
31D99
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


20-RECEIVED From 10.10.1.2:5060 at 16018


21-RECEIVED From 10.10.1.14:5060 at 16018


22-RECEIVED From 10.10.1.29:5060 at 16020


23-SENT To 10.10.1.14:5060 at 16022
INVITE sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 2 INVITE
User-Agent: Excel_CSP/83.10.75
Contact: 20597<sip:20597@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uac
Min-SE: 300
Content-Type: application/sdp
Content-Length: 221

v=0
o=sip 1122578408 1122578408 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 0.0.0.0
t=0 0
m=audio 8000 RTP/AVP 0 8 3 101
```

```
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

24-RECEIVED From 10.10.1.14:5060 at 16022
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 2 INVITE
Server: X-Lite release 1103m
Content-Length: 0


25-RECEIVED From 10.10.1.14:5060 at 16022
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 2 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 292

v=0
o=35751 292794390 292802031 IN IP4 10.10.1.14
s=X-Lite
c=IN IP4 10.10.1.14
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

26-SENT To 10.10.1.14:5060 at 16022
ACK sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 2 ACK
```

```
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


27-SENT To 10.10.1.2:5060 at 16022
INVITE sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
From: <sip:35751@10.10.1.14>;tag=45103e7b
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35319 INVITE
User-Agent: Excel_CSP/83.10.75
Contact: 35751<sip:35751@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uac
Min-SE: 300
Content-Type: application/sdp
Content-Length: 221

v=0
o=sip 1122578408 1122578408 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 0.0.0.0
t=0 0
m=audio 8002 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

28-RECEIVED From 10.10.1.2:5060 at 16022
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.1.47
From: <sip:35751@10.10.1.14>;tag=45103e7b
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35319 INVITE
Server: X-Lite release 1103m
Content-Length: 0


29-RECEIVED From 10.10.1.2:5060 at 16022
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: <sip:35751@10.10.1.14>;tag=45103e7b
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35319 INVITE
```

```
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 290

v=0
o=20597 722096328 722096343 IN IP4 10.10.1.2
s=X-Lite
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

30-SENT To 10.10.1.2:5060 at 16022
ACK sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
From: <sip:35751@10.10.1.14>;tag=45103e7b
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35319 ACK
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


31-SENT To 10.10.1.2:5060 at 16026
INVITE sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
From: <sip:35751@10.10.1.14>;tag=45103e7b
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35320 INVITE
User-Agent: Excel_CSP/83.10.75
Contact: 35751<sip:35751@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uac
Min-SE: 300
Content-Type: application/sdp
Content-Length: 224

v=0
o=sip 1122578412 1122578412 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 10.10.1.29
t=0 0
m=audio 8000 RTP/AVP 0 8 3 101
```

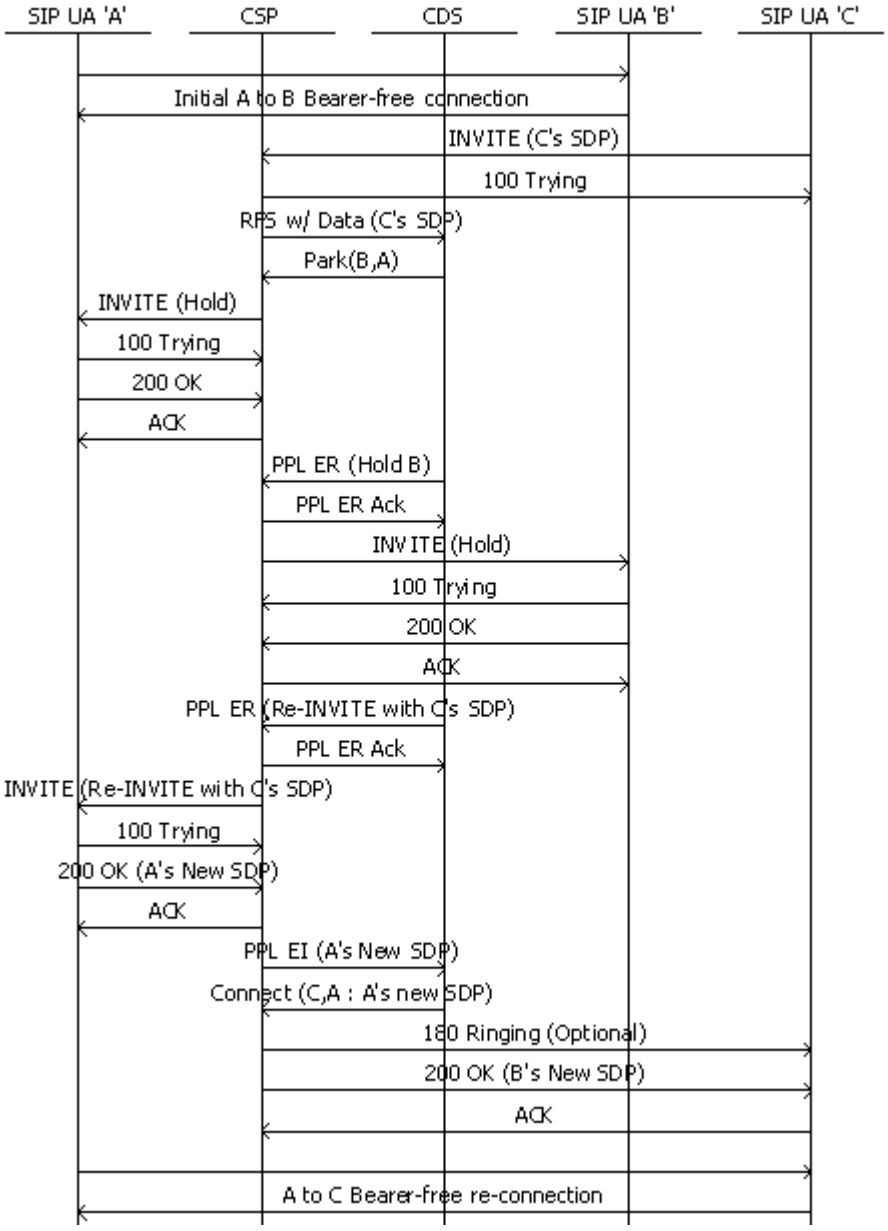
```
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

32-RECEIVED From 10.10.1.2:5060 at 16026
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.1.47
From: <sip:35751@10.10.1.14>;tag=45103e7b
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35320 INVITE
Server: X-Lite release 1103m
Content-Length: 0


33-RECEIVED From 10.10.1.2:5060 at 16026
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: <sip:35751@10.10.1.14>;tag=45103e7b
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35320 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 290

v=0
o=20597 722096328 722096343 IN IP4 10.10.1.2
s=X-Lite
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

34-SENT To 10.10.1.2:5060 at 16026
ACK sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
From: <sip:35751@10.10.1.14>;tag=45103e7b
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35320 ACK
```

```
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


35-SENT To 10.10.1.29:5060 at 16027
SIP/2.0 200 OK
To: <sip:20597@10.10.1.2>;tag=33963e90
From: sathia <sip:8323@10.10.1.29>;tag=851025480
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 INVITE
Contact: 20597<sip:20597@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK2017BD3E5D074025B2
   7B107065B
31D99
User-Agent: Excel_CSP/83.10.75
Content-Type: application/sdp
Content-Length: 223

v=0
o=sip 1122578413 1122578413 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

36-RECEIVED From 10.10.1.29:5060 at 16027
ACK sip:20597@10.10.1.47:5060 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK93925BBE56494E7A8A
   D16EDBDB4
711EB
From: sathia <sip:8323@10.10.1.29>;tag=851025480
To: <sip:20597@10.10.1.2>;tag=33963e90
Contact: <sip:8323@10.10.1.29:5060>
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 ACK
Max-Forwards: 70
Content-Length: 0


37-RECEIVED From 10.10.1.2:5060 at 16028
```

```
38-RECEIVED From 10.10.1.14:5060 at 16029


39-RECEIVED From 10.10.1.29:5060 at 16030


40-SENT To 10.10.1.14:5060 at 16031
BYE sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 3 BYE
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


41-RECEIVED From 10.10.1.14:5060 at 16031
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 3 BYE
Server: X-Lite release 1103m
Content-Length: 0


42-RECEIVED From 10.10.1.2:5060 at 16039
```

Third Party Connected to Calling Party

The call flow below shows a scenario where the initial conversation between two SIP endpoints (A/B) is put on hold and a third party (C) is connected to the calling party of the initial conversation (A).



API Message

```
X->H
    [01 1d 00 2d 00 08 0c 00 01 0d 03 00 64 0c 00 33 01 03
   00 33
     01 09 00 10 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
     19 00 06 33 35 37 35 31 00 29 1b 00 0b 31 30 2e 31 30
     2e 31 2e 31 34 00 29 1c 00 04 00 00 13 c4 29 23 00 06
     32 30 35 39 37 00 29 25 00 0a 31 30 2e 31 30 2e 31 2e
     32 00 29 26 00 04 00 00 13 c4 29 2d 00 06 32 30 35 39
     37 00 29 2f 00 0a 31 30 2e 31 30 2e 31 2e 32 00 29 30
     00 04 00 00 13 c4 29 33 00 01 01 27 18 00 08 02 00 00
     00 05 20 59 70 27 17 00 06 02 00 05 35 75 10 29 ff 00
     75 2a 0e 00 04 0a 0a 01 02 2a 01 00 69 2a 03 00 01 00
     2a 07 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 02
     2a 09 00 01 02 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a
     08 00 02 00 01 2a 09 00 01 01 2a 0b 00 04 00 00 1f 40
     2a 02 00 13 2a 08 00 02 00 16 2a 09 00 01 16 2a 0b 00
     04 00 00 1f 40 2a 02 00 13 2a 08 00 02 01 65 2a 09 00
     01 32 2a 0b 00 04 00 00 1f 40 29 16 00 01 01]

H->X
    [00 0c 00 2d 00 08 0c 00 01 0d 03 00 64 0c]

H->X
    [00 0c 00 2d 00 08 0c 00 01 0d 03 00 64 0c]

H->X
    [01 40 00 e8 00 00 0c 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 71 00 0f 00
     01 0b 00 71 00 02 00 00 03 00 33 01 11 00 11 27 4e 00
     02 00 05 27 7e 00 03 08 00 00 29 19 00 06 33 35 37 35
     31 00 29 1b 00 0b 31 30 2e 31 30 2e 31 2e 31 34 00 29
     1c 00 04 00 00 13 c4 29 23 00 06 32 30 35 39 37 00 29
     25 00 0a 31 30 2e 31 30 2e 31 2e 32 00 29 26 00 04 00
     00 13 c4 29 2d 00 06 32 30 35 39 37 00 29 2f 00 0a 31
     30 2e 31 30 2e 31 2e 32 00 29 30 00 04 00 00 13 c4 29
     33 00 01 01 27 18 00 08 02 00 00 00 05 20 59 70 27 17
     00 06 02 00 05 15 75 80 2a 00 00 75 2a 0e 00 04 0a 0a
     01 02 2a 01 00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f
     40 2a 02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b
     00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09
     00 01 01 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00
     02 00 16 2a 09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02
     00 13 2a 08 00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00
     00 1f 40 29 16 00 01 01 2a 0e 00 04 0a 0a 01 02]

X->H
```

```
    [00 14 00 e8 00 00 0c 00 10 01 02 1e 09 00 01 00 39 00
   03 00
     64 0d]

X->H
    [00 11 00 43 00 0a 0c 00 01 0d 03 00 64 0d 00 a7 00 24
   00]

H->X
    [00 05 00 43 00 0a 0c]

H->X
    [00 05 00 43 00 0a 0c]

X->H
    [00 91 00 43 00 0b 0c 00 01 0d 03 00 64 0d 00 a7 00 20
   01 03
     00 33 00 7b 00 01 29 ff 00 75 2a 0e 00 04 0a 0a 01 0e
     2a 01 00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f 40 2a
     02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04
     00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01
     01 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00
     16 2a 09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02 00 13
     2a 08 00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00 00 1f
     40]

H->X
    [00 05 00 43 00 0b 0c]

H->X
    [00 05 00 43 00 0b 0c]

H->X
    [00 9f 00 05 00 00 0c 00 02 0d 03 00 64 0c 0d 03 00 64
     0d 01 02 03 00 1e 00 07 00 01 01 16 00 01 01 03 00 33
     00 7b 00 01 2a 00 00 75 2a 0e 00 04 0a 0a 01 0e 2a 01
     00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f 42 2a 02 00
     13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04 00 00
     1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01 01 2a
     0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 16 2a
     09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08
     00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00 00 1f 40]

X->H
    [00 07 00 05 00 00 0c 00 10]

X->H
    [01 1b 00 2d 00 09 0c 00 01 0d 03 00 64 0e 00 33 01 03
   00 33
     01 07 00 10 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
```

```
     19 00 06 32 30 35 39 37 00 29 1b 00 0a 31 30 2e 31 30
     2e 31 2e 32 00 29 1c 00 04 00 00 13 c4 29 23 00 05 38
     33 32 33 00 29 25 00 0b 31 30 2e 31 30 2e 31 2e 32 39
     00 29 26 00 04 00 00 13 c4 29 2d 00 05 38 33 32 33 00
     29 2f 00 0b 31 30 2e 31 30 2e 31 2e 32 39 00 29 30 00
     04 00 00 13 c4 29 33 00 01 01 27 18 00 07 02 00 00 00
     04 83 23 27 17 00 06 02 00 05 20 59 70 29 ff 00 75 2a
     0e 00 04 0a 0a 01 1d 2a 01 00 69 2a 03 00 01 00 2a 07
     00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 02 2a 09
     00 01 02 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00
     02 00 01 2a 09 00 01 01 2a 0b 00 04 00 00 1f 40 2a 02
     00 13 2a 08 00 02 00 16 2a 09 00 01 16 2a 0b 00 04 00
     00 1f 40 2a 02 00 13 2a 08 00 02 01 65 2a 09 00 01 32
     2a 0b 00 04 00 00 1f 40 29 16 00 01 01]

H->X
    [00 0c 00 2d 00 09 0c 00 01 0d 03 00 64 0e]

H->X
    [00 0c 00 2d 00 09 0c 00 01 0d 03 00 64 0e]

H->X
    [00 11 00 bf 00 00 0c 00 02 0d 03 00 64 0e 0d 03 00 64
     0e]

X->H
    [00 07 00 bf 00 00 0c 00 10]

H->X
    [00 11 00 bf 00 00 0c 00 02 0d 03 00 64 0c 0d 03 00 64
     0d]

X->H
    [00 07 00 bf 00 00 0c 00 10]

X->H
    [00 0e 00 42 00 64 0c 00 01 0d 03 00 64 0d 04 00]

H->X
    [00 1d 00 44 00 00 0c 00 01 0d 03 00 64 0c 00 a7 00 1e
     01 03 00 33 00 07 00 01 27 b3 00 01 03]

H->X
    [00 05 00 42 00 64 0c]

H->X
    [00 05 00 42 00 64 0c]

X->H
    [00 07 00 44 00 00 0c 00 10]
```

```
H->X
    [00 91 00 44 00 00 0c 00 01 0d 03 00 64 0c 00 a7 00 24
     01 03 00 33 00 7b 00 01 2a 00 00 75 2a 0e 00 04 0a 0a
     01 1d 2a 01 00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f
     40 2a 02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b
     00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09
     00 01 01 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00
     02 00 16 2a 09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02
     00 13 2a 08 00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00
     00 1f 40]

X->H
    [00 07 00 44 00 00 0c 00 10]

X->H
    [00 91 00 43 00 0c 0c 00 01 0d 03 00 64 0c 00 a7 00 20
   01 03
     00 33 00 7b 00 01 29 ff 00 75 2a 0e 00 04 0a 0a 01 02
     2a 01 00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f 40 2a
     02 00 13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04
     00 00 1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01
     01 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00
     16 2a 09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02 00 13
     2a 08 00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00 00 1f
     40]

H->X
    [00 05 00 43 00 0c 0c]

H->X
    [00 05 00 43 00 0c 0c]

H->X
    [00 9f 00 05 00 00 0c 00 02 0d 03 00 64 0e 0d 03 00 64
     0c 01 02 03 00 1e 00 07 00 01 01 16 00 01 01 03 00 33
     00 7b 00 01 2a 00 00 75 2a 0e 00 04 0a 0a 01 02 2a 01
     00 69 2a 03 00 01 00 2a 07 00 04 00 00 1f 40 2a 02 00
     13 2a 08 00 02 00 02 2a 09 00 01 02 2a 0b 00 04 00 00
     1f 40 2a 02 00 13 2a 08 00 02 00 01 2a 09 00 01 01 2a
     0b 00 04 00 00 1f 40 2a 02 00 13 2a 08 00 02 00 16 2a
     09 00 01 16 2a 0b 00 04 00 00 1f 40 2a 02 00 13 2a 08
     00 02 01 65 2a 09 00 01 32 2a 0b 00 04 00 00 1f 40]

X->H
    [00 07 00 05 00 00 0c 00 10]

H->X
    [00 11 00 08 00 00 0c 00 02 0d 03 00 64 0d 0d 03 00 64
     0d]
```

```
X->H
    [00 07 00 08 00 00 0c 00 10]

X->H
    [00 0c 00 49 00 0a 0c 00 01 0d 03 00 64 0d]

H->X
    [00 05 00 49 00 0a 0c]
```

SIP messages

```
1 -RECEIVED From 10.10.1.2:5060 at 15995
INVITE sip:35751@10.10.1.14 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bK1FDA7D2876B04953942
   90D381FB3
389C
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
To: <sip:35751@10.10.1.14>
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 INVITE
Max-Forwards: 70
Content-Type: application/sdp
User-Agent: X-Lite release 1103m
Content-Length: 290

v=0
o=20597 722096328 722096343 IN IP4 10.10.1.2
s=X-Lite
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

2 -SENT To 10.10.1.2:5060 at 15995
SIP/2.0 100 Trying
To: <sip:35751@10.10.1.14>;tag=45103e7b
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 INVITE
Contact: 35751<sip:35751@10.10.1.47:5060>
```

```
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bK1FDA7D2876B04953942
   90D381FB3
389C
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


3 -SENT To 10.10.1.14:5060 at 15996
INVITE sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751<sip:35751@10.10.1.14:5060>
From: 20597<sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
Contact: 20597<sip:20597@10.10.1.47:5060>
User-Agent: Excel_CSP/83.10.75
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 223

v=0
o=sip 1122578382 1122578382 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

4 -RECEIVED From 10.10.1.14:5060 at 15996
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


5 -RECEIVED From 10.10.1.14:5060 at 15996
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
```

```
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 1 INVITE
Server: X-Lite release 1103m
Content-Length: 0


6 -RECEIVED From 10.10.1.2:5060 at 15998


7 -RECEIVED From 10.10.1.14:5060 at 15998


8 -RECEIVED From 10.10.1.29:5060 at 15999


9 -RECEIVED From 10.10.1.14:5060 at 16004
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 1 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 292

v=0
o=35751 292794390 292802031 IN IP4 10.10.1.14
s=X-Lite
c=IN IP4 10.10.1.14
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

10-SENT To 10.10.1.14:5060 at 16004
ACK sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751<sip:35751@10.10.1.14:5060>;tag=4107074591
From: 20597<sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 1 ACK
```

```
Content-Length: 0


11-RECEIVED From 10.10.1.2:5060 at 16008


12-RECEIVED From 10.10.1.14:5060 at 16008


13-SENT To 10.10.1.2:5060 at 16009
SIP/2.0 180 Ringing
To: <sip:35751@10.10.1.14>;tag=45103e7b
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 INVITE
Contact: 35751<sip:35751@10.10.1.47:5060>
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bK1FDA7D2876B04953942
   90D381FB3
389C
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


14-SENT To 10.10.1.2:5060 at 16009
SIP/2.0 200 OK
To: <sip:35751@10.10.1.14>;tag=45103e7b
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 INVITE
Contact: 35751<sip:35751@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bK1FDA7D2876B04953942
   90D381FB3
389C
User-Agent: Excel_CSP/83.10.75
Content-Type: application/sdp
Content-Length: 224

v=0
o=sip 1122578395 1122578395 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 10.10.1.14
t=0 0
m=audio 8002 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
```

```
a=rtpmap:101 telephone-event/8000

15-RECEIVED From 10.10.1.2:5060 at 16009
ACK sip:35751@10.10.1.47:5060 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.2:5060;rport;branch=z9hG4bKDC857736F60944AEB97
   C9B1D843C
7181
From: Subbu <sip:20597@10.10.1.2>;tag=803647026
To: <sip:35751@10.10.1.14>;tag=45103e7b
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35318 ACK
Max-Forwards: 70
Content-Length: 0


16-RECEIVED From 10.10.1.29:5060 at 16010


17-RECEIVED From 10.10.1.29:5060 at 16016
INVITE sip:20597@10.10.1.2 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK2017BD3E5D074025B2
   7B107065B
31D99
From: sathia <sip:8323@10.10.1.29>;tag=851025480
To: <sip:20597@10.10.1.2>
Contact: <sip:8323@10.10.1.29:5060>
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 INVITE
Max-Forwards: 70
Content-Type: application/sdp
User-Agent: X-Lite release 1103m
Content-Length: 293

v=0
o=8323 1402407484 1402407500 IN IP4 10.10.1.29
s=X-Lite
c=IN IP4 10.10.1.29
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

```
18-SENT To 10.10.1.29:5060 at 16016
SIP/2.0 100 Trying
To: <sip:20597@10.10.1.2>;tag=33963e90
From: sathia <sip:8323@10.10.1.29>;tag=851025480
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 INVITE
Contact: 20597<sip:20597@10.10.1.47:5060>
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK2017BD3E5D074025B2
   7B107065B
31D99
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


19-SENT To 10.10.1.29:5060 at 16017
SIP/2.0 180 Ringing
To: <sip:20597@10.10.1.2>;tag=33963e90
From: sathia <sip:8323@10.10.1.29>;tag=851025480
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 INVITE
Contact: 20597<sip:20597@10.10.1.47:5060>
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK2017BD3E5D074025B2
   7B107065B
31D99
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


20-RECEIVED From 10.10.1.2:5060 at 16018


21-RECEIVED From 10.10.1.14:5060 at 16018


22-RECEIVED From 10.10.1.29:5060 at 16020


23-SENT To 10.10.1.14:5060 at 16022
INVITE sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 2 INVITE
User-Agent: Excel_CSP/83.10.75
Contact: 20597<sip:20597@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uac
```

```
Min-SE: 300
Content-Type: application/sdp
Content-Length: 221

v=0
o=sip 1122578408 1122578408 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 0.0.0.0
t=0 0
m=audio 8000 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

24-RECEIVED From 10.10.1.14:5060 at 16022
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 2 INVITE
Server: X-Lite release 1103m
Content-Length: 0


25-RECEIVED From 10.10.1.14:5060 at 16022
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 2 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 292

v=0
o=35751 292794390 292802031 IN IP4 10.10.1.14
s=X-Lite
c=IN IP4 10.10.1.14
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
```

```
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

26-SENT To 10.10.1.14:5060 at 16022
ACK sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 2 ACK
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


27-SENT To 10.10.1.2:5060 at 16022
INVITE sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
From: <sip:35751@10.10.1.14>;tag=45103e7b
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35319 INVITE
User-Agent: Excel_CSP/83.10.75
Contact: 35751<sip:35751@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uac
Min-SE: 300
Content-Type: application/sdp
Content-Length: 221

v=0
o=sip 1122578408 1122578408 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 0.0.0.0
t=0 0
m=audio 8002 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

28-RECEIVED From 10.10.1.2:5060 at 16022
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.1.47
From: <sip:35751@10.10.1.14>;tag=45103e7b
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35319 INVITE
Server: X-Lite release 1103m
Content-Length: 0
```

```
29-RECEIVED From 10.10.1.2:5060 at 16022
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: <sip:35751@10.10.1.14>;tag=45103e7b
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35319 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 290

v=0
o=20597 722096328 722096343 IN IP4 10.10.1.2
s=X-Lite
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

30-SENT To 10.10.1.2:5060 at 16022
ACK sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
From: <sip:35751@10.10.1.14>;tag=45103e7b
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35319 ACK
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


31-SENT To 10.10.1.2:5060 at 16026
INVITE sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
From: <sip:35751@10.10.1.14>;tag=45103e7b
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35320 INVITE
User-Agent: Excel_CSP/83.10.75
Contact: 35751<sip:35751@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uac
```

```
Min-SE: 300
Content-Type: application/sdp
Content-Length: 224

v=0
o=sip 1122578412 1122578412 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 10.10.1.29
t=0 0
m=audio 8000 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

32-RECEIVED From 10.10.1.2:5060 at 16026
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.10.1.47
From: <sip:35751@10.10.1.14>;tag=45103e7b
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35320 INVITE
Server: X-Lite release 1103m
Content-Length: 0


33-RECEIVED From 10.10.1.2:5060 at 16026
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: <sip:35751@10.10.1.14>;tag=45103e7b
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
Contact: <sip:20597@10.10.1.2:5060>
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35320 INVITE
Content-Type: application/sdp
Server: X-Lite release 1103m
Content-Length: 290

v=0
o=20597 722096328 722096343 IN IP4 10.10.1.2
s=X-Lite
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 98 97 101
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:3 gsm/8000
a=rtpmap:98 iLBC/8000
a=rtpmap:97 speex/8000
```

```
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

34-SENT To 10.10.1.2:5060 at 16026
ACK sip:20597@10.10.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: Subbu <sip:20597@10.10.1.2>;tag=803647026
From: <sip:35751@10.10.1.14>;tag=45103e7b
Call-ID: 1493360C-C36D-4A22-945A-6A8D06BB5BB8@10.10.1.2
CSeq: 35320 ACK
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


35-SENT To 10.10.1.29:5060 at 16027
SIP/2.0 200 OK
To: <sip:20597@10.10.1.2>;tag=33963e90
From: sathia <sip:8323@10.10.1.29>;tag=851025480
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 INVITE
Contact: 20597<sip:20597@10.10.1.47:5060>
Supported: timer
Session-Expires: 1800; refresher=uas
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK2017BD3E5D074025B2
   7B107065B
31D99
User-Agent: Excel_CSP/83.10.75
Content-Type: application/sdp
Content-Length: 223

v=0
o=sip 1122578413 1122578413 IN IP4 10.10.1.47
s=SIP_Call
c=IN IP4 10.10.1.2
t=0 0
m=audio 8000 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000

36-RECEIVED From 10.10.1.29:5060 at 16027
ACK sip:20597@10.10.1.47:5060 SIP/2.0
Via: SIP/2.0/UDP
   10.10.1.29:5060;rport;branch=z9hG4bK93925BBE56494E7A8A
   D16EDBDB4
711EB
From: sathia <sip:8323@10.10.1.29>;tag=851025480
To: <sip:20597@10.10.1.2>;tag=33963e90
```

```
Contact: <sip:8323@10.10.1.29:5060>
Call-ID: D9DBF498-035A-4676-8223-B989D91D92E4@10.10.1.29
CSeq: 6321 ACK
Max-Forwards: 70
Content-Length: 0


37-RECEIVED From 10.10.1.2:5060 at 16028


38-RECEIVED From 10.10.1.14:5060 at 16029


39-RECEIVED From 10.10.1.29:5060 at 16030


40-SENT To 10.10.1.14:5060 at 16031
BYE sip:35751@10.10.1.14:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.47
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 3 BYE
User-Agent: Excel_CSP/83.10.75
Content-Length: 0


41-RECEIVED From 10.10.1.14:5060 at 16031
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 10.10.1.47
From: 20597 <sip:20597@10.10.1.2:5060>;tag=410941543e7c
To: 35751 <sip:35751@10.10.1.14:5060>;tag=4107074591
Contact: <sip:35751@10.10.1.14:5060>
Call-ID: EXCEL-CSP12.100d.15996.360@10.10.1.47
CSeq: 3 BYE
Server: X-Lite release 1103m
Content-Length: 0


42-RECEIVED From 10.10.1.2:5060 at 16039
```

PPL Changes The following PPL change supports this feature.

PPL Event Request to Generate RE-INVITE message

PPL Event Request (0x0024)

The PPL Event Request (0x0024) is used to generate a RE-INVITE message to the SIP endpoint connected to the channel indicated in the channel AIB.

This PPL Event Request will work only if the call is in bearer-free mode or else it will NACK with a value 0x130C (Call is not in bearer-free mode.)

The PPL Event Request is added to the SIP UA (0x00A7) component. Refer to *PPL Information: SIP UA 0x00A7 (5-14)*

When the PPL Event Request generates a RE-INVITE message, the 200 OK SIP message received from the other end having the SDP information will be reported to the host application using the PPL Event Indication (0x0020) by the SIP UA component.

The data in the PPL Event Indication could be used within the subsequent *Connect with Data* (0x0005) message. The PPL Event Request is ACKED if the data is valid.

Important! The PPL Event Request should contain a Media Remote End Point Info (0x2A00) TLV.

Call Flows

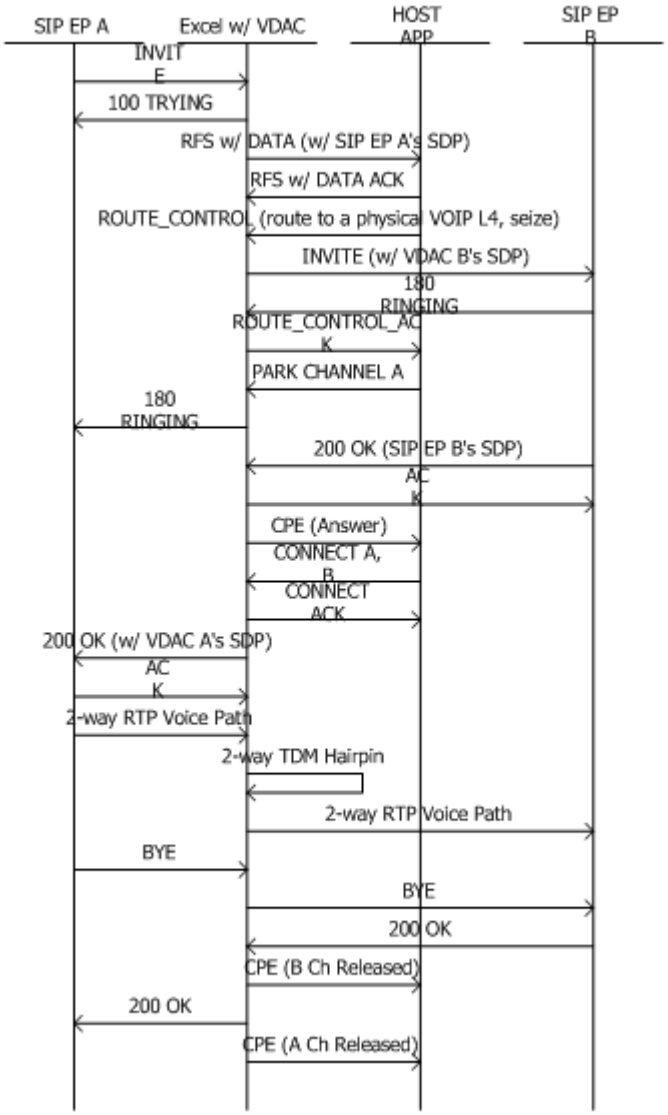
Overview The call flows in this section cover the following scenarios:

- Traditional Switching SIP-to-SIP Call
- Call Agent Switching with Host Application Control
- Dual Switching Scenario
  Includes diagram of example
- Dual Switching DSP-ONE Based Prompt and Collect Scenarios
  Includes diagram of example
- Mid-Call RFC 2833 DTMF Signaling by Near-End
  Includes diagrams of three examples

Traditional Switching SIP-to-SIP Call

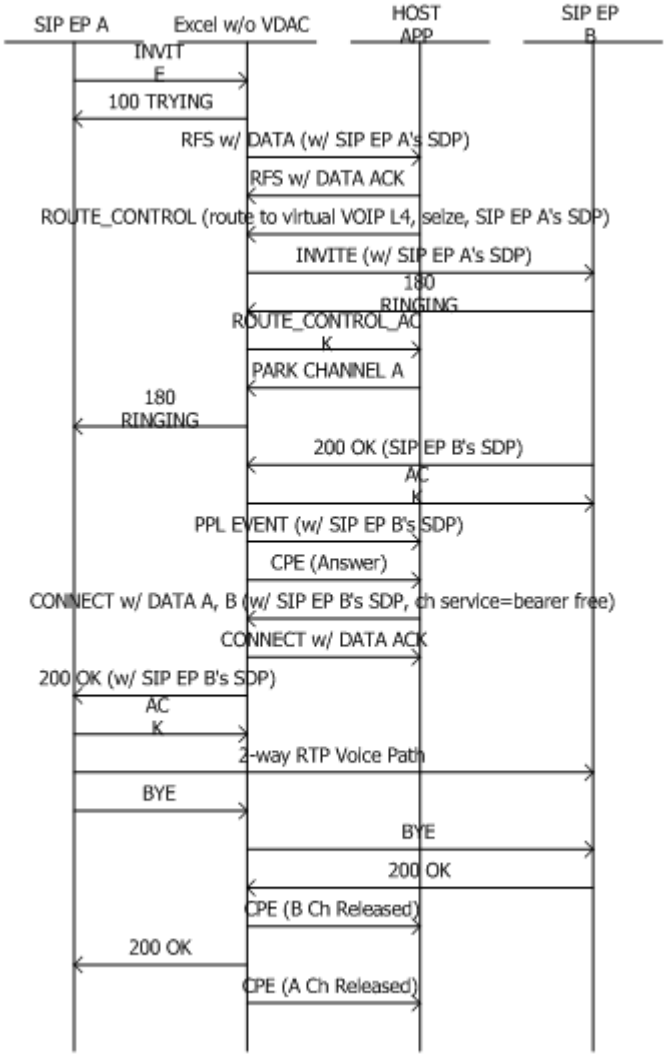
Traditional switching is supported in this release and in previous releases.

The following call flow shows how a traditional bearer-switched SIP-to-SIP call is established through the CSP. The CSP acts as a back-to-back user agent in that the SIP end-points signal with the CSP only, and connect to VDAC RTP ports. The VDAC RTP ports are internally connected over the Time Slot Interchange (TSI) fabric for an end-to-end speech path. This internal connection is the “hairpin.”



Call Agent Switching with Host Application Control

The following call flow shows Call Agent switching with host application control. This call flow is a typical example of how a host application controls and participates in network signaling data transfer from one side to the other. From the OAM&P perspective, all Call Agent calls require host applications to configure virtual span/channels in the system. Virtual span/channels are essentially span/channels assigned to virtual slots that contain virtual VoIP cards. Additionally, you must configure the router so that virtual span/channel-based resource groups are defined and associated with appropriate route lookup rules.



Dual Switching Scenario

The following call flow and *Figure 6-5, Dual Switching Scenario for Pre-Paid Application with External IVR (6-97)* depicts a normal dual switching scenario. The caller is first connected to an Interactive Voice Response (IVR) system through the bearer-switched path provided by the CSP. When the host application releases the IVR call leg, the caller is connected to the long-haul called party, bypassing the bearer switching facilities of the CSP. Overall, the caller's connection to the IVR system is an intermediary step that can be repeated any number of times; however, the caller's connection to the IVR typically alternates with the connection to the long-haul party.

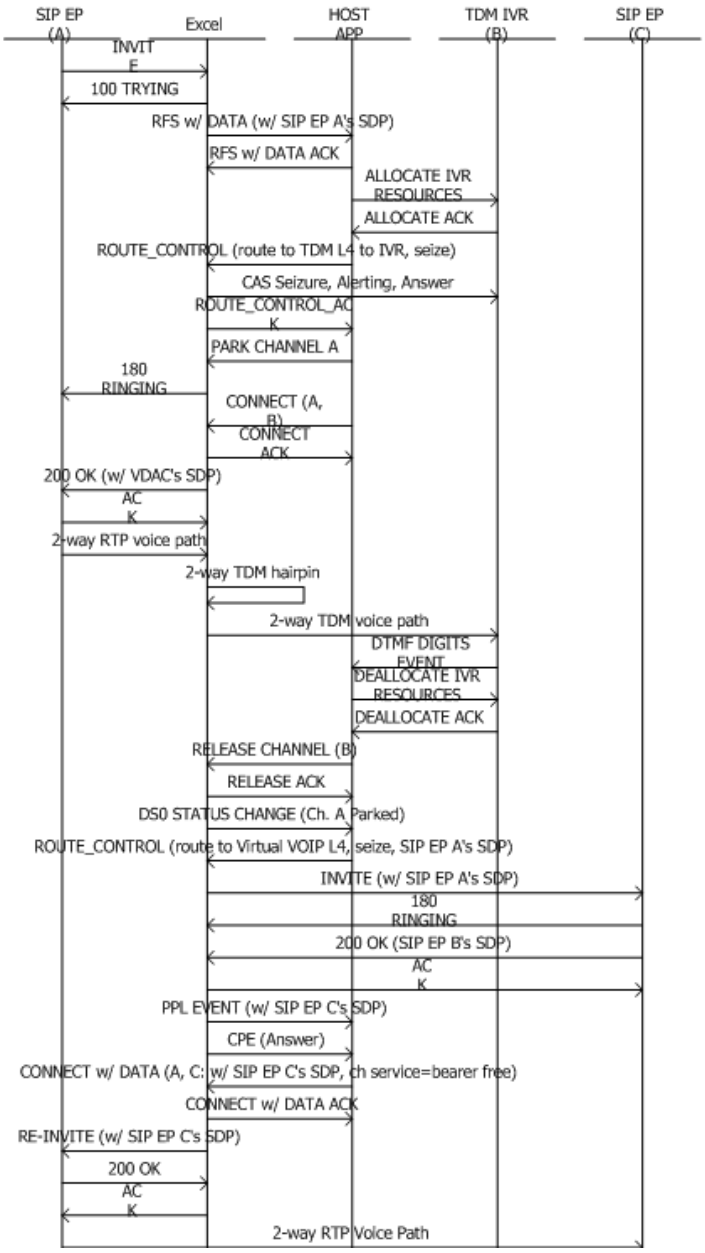
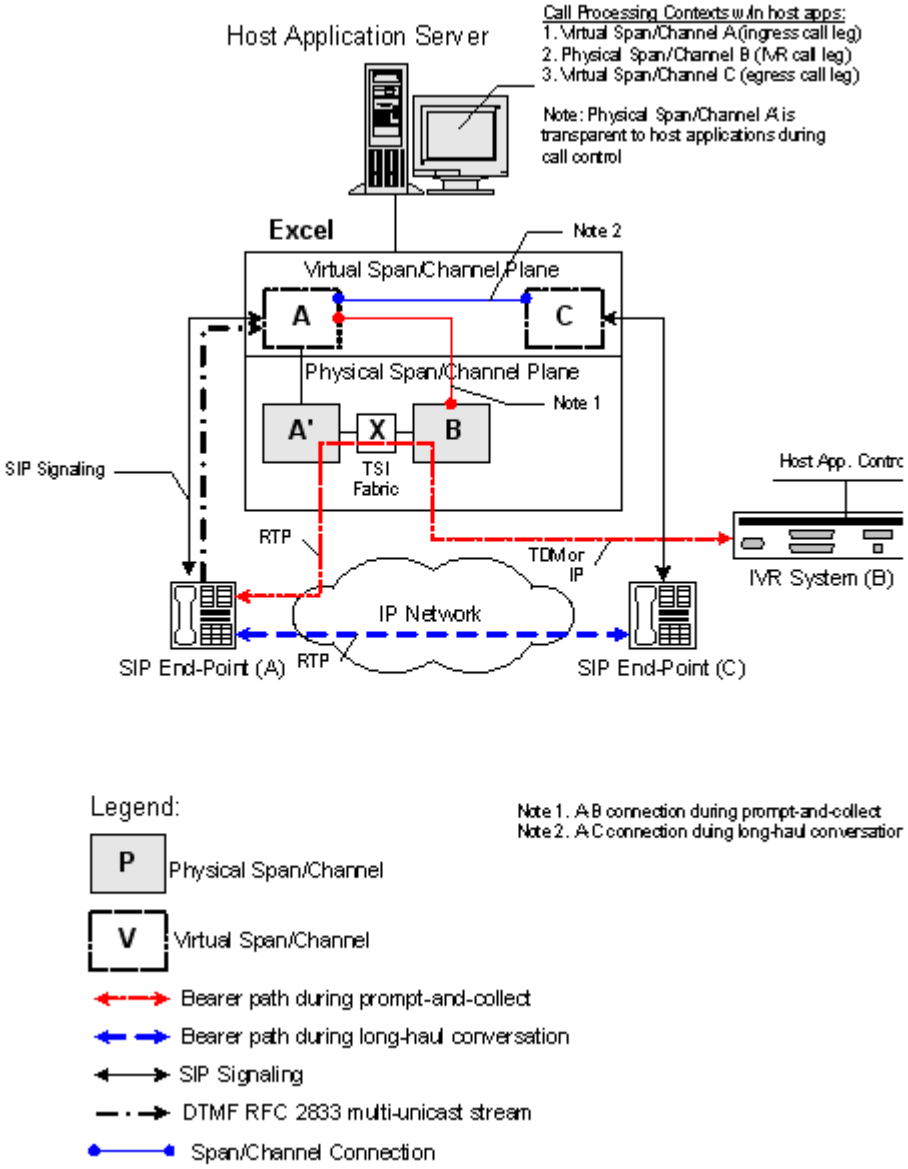

SIP EP (A)
Excel
HOST APP
TDM IVR (B)
SIP EP (C)
INVITE
100 TRYING
RFS w/ DATA (w/ SIP EP A's SDP)
RFS w/ DATA ACK
ALLOCATE IVR RESOURCES
ALLOCATE ACK
ROUTE_CONTROL (route to TDM L4 to IVR, seize)
CAS Seizure, Alerting, Answer
ROUTE_CONTROL_ACK
PARK CHANNEL A
180 RINGING
CONNECT (A, B)
CONNECT ACK
200 OK (w/ VDAC's SDP)
ACK
2-way RTP voice path
2-way TDM hairpin
2-way TDM voice path
DTMF DIGITS EVENT
DEALLOCATE IVR RESOURCES
DEALLOCATE ACK
RELEASE CHANNEL (B)
RELEASE ACK
DS0 STATUS CHANGE (Ch. A Parked)
ROUTE_CONTROL (route to Virtual VOIP L4, seize, SIP EP A's SDP)
INVITE (w/ SIP EP A's SDP)
180 RINGING
200 OK (SIP EP B's SDP)
ACK
PPL EVENT (w/ SIP EP C's SDP)
CPE (Answer)
CONNECT w/ DATA (A, C: w/ SIP EP C's SDP, ch service=bearer free)
CONNECT w/ DATA ACK
RE-INVITE (w/ SIP EP C's SDP)
200 OK
ACK
2-way RTP Voice Path

Figure 6-5 Dual Switching Scenario for Pre-Paid Application with External IVR



Dual Scenario DSP-ONE Based Prompt and Collect Scenarios

It is transparent to Call Agent software whether media services are run on an external IVR or an internal DSP-ONE card. The call flow below shows a dual switching scenario where a DSP-ONE card is used for media services. Note that only A and B channels are shown because the DSP-ONE services are requested on channel A itself. This scenario is illustrated in *Figure 6-6, Dual Switching Scenario for Pre-Paid Application with DSP-ONE Card (6-99)* following this call flow.

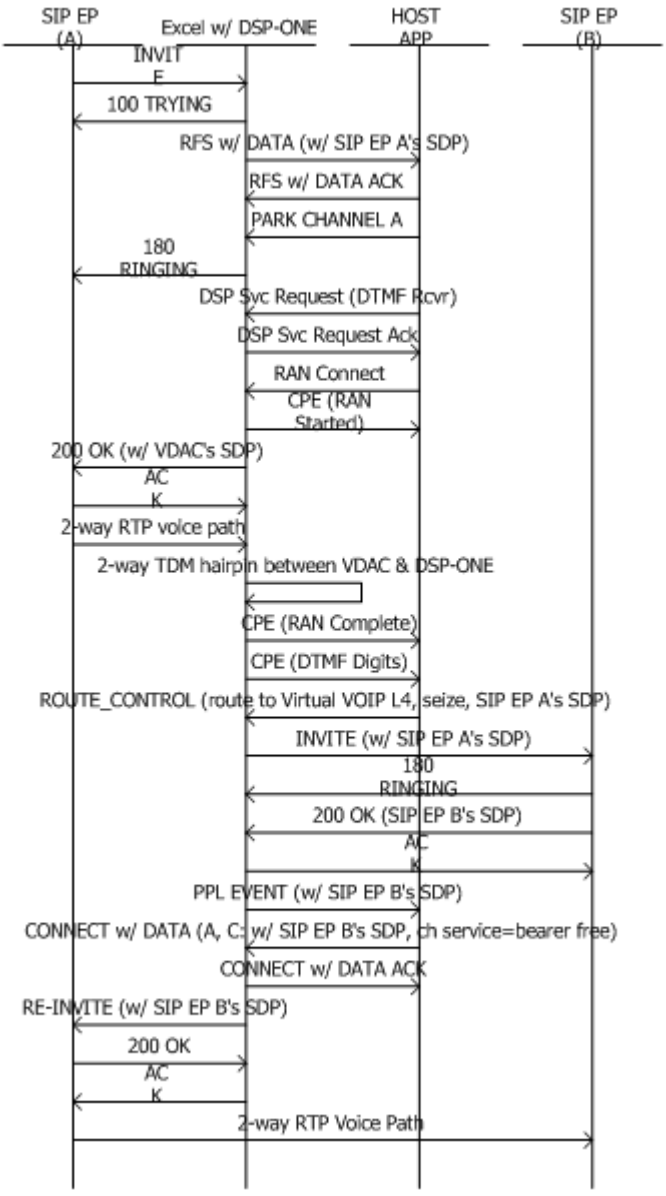
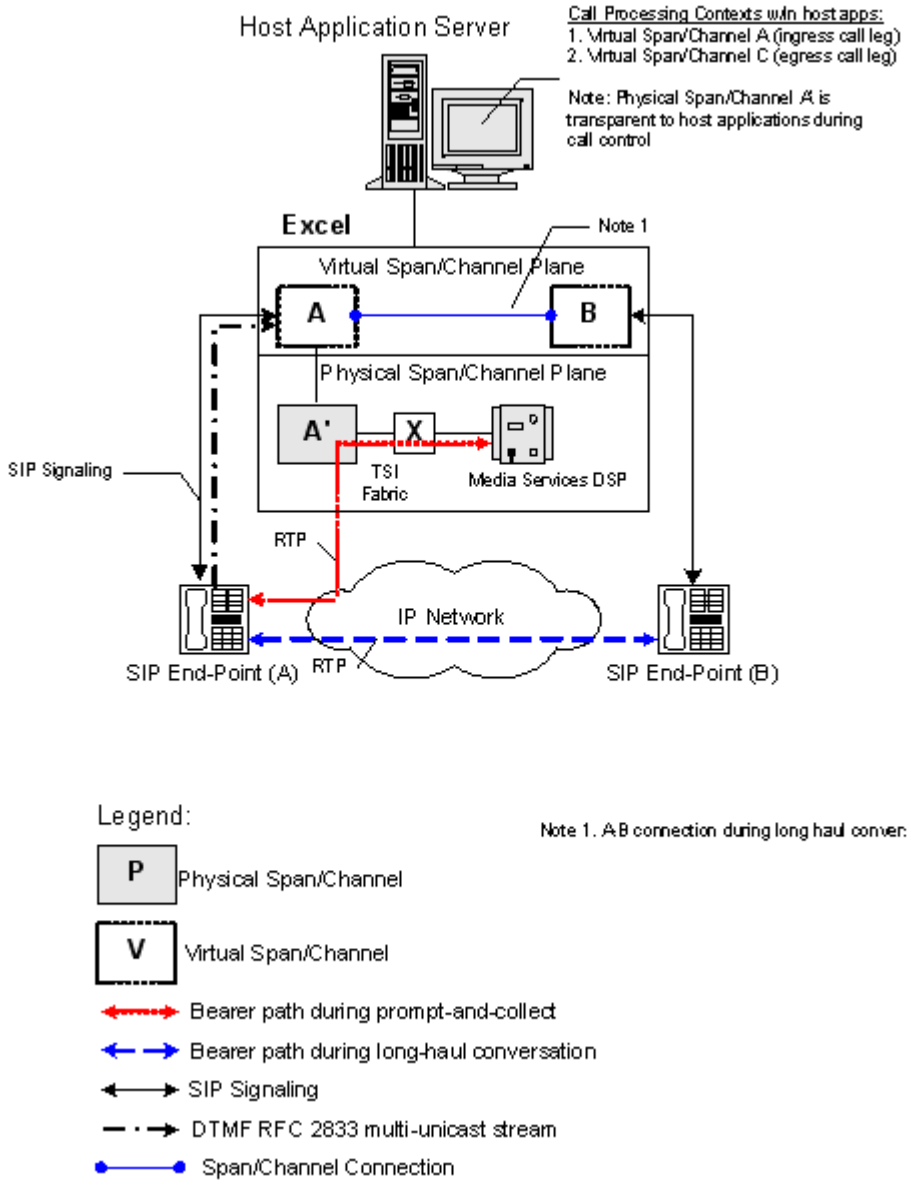


Figure 6-6 Dual Switching Scenario for Pre-Paid Application with DSP-ONE Card



Mid-Call RFC 2833 DTMF Signaling by Near-End

The call flow below shows the Mid-call RFC 2833 DTMF signaling by near-end scenario. In this scenario, the caller specifically instructs the CSP for long-distance call termination and reconnection with an IVR. The CSP uses a special sequence of DTMF digits (such as long # sequence) to communicate and decipher the caller's intent. Following this call flow are three diagrams showing three different environments using RFC 2833 multi-unicast.

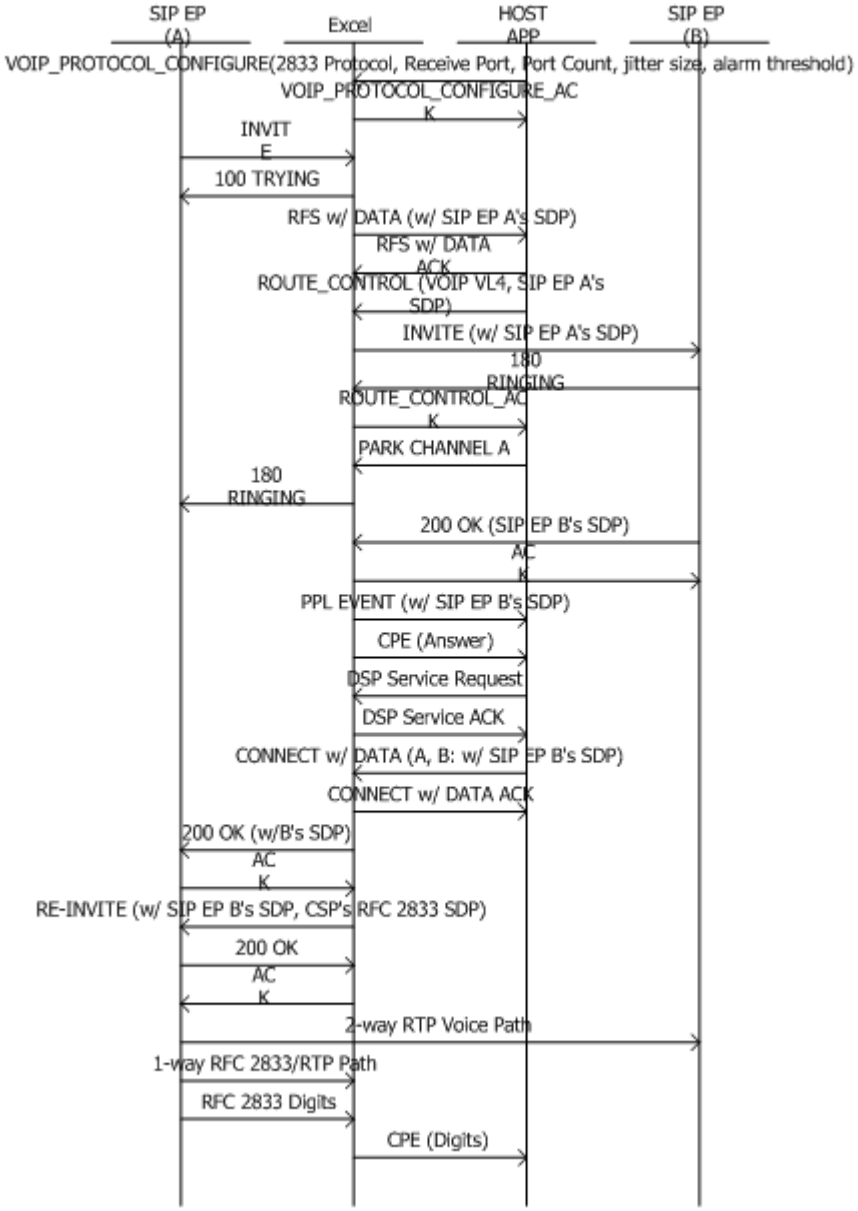

SIP EP (A)
Excel
HOST APP
SIP EP (B)
VOIP_PROTOCOL_CONFIGURE(2833 Protocol, Receive Port, Port Count, jitter size, alarm threshold)
VOIP_PROTOCOL_CONFIGURE_ACK
INVITE
100 TRYING
RFS w/ DATA (w/ SIP EP A's SDP)
RFS w/ DATA ACK
ROUTE_CONTROL (VOIP VL4, SIP EP A's SDP)
INVITE (w/ SIP EP A's SDP)
180 RINGING
ROUTE_CONTROL_ACK
PARK CHANNEL A
180 RINGING
200 OK (SIP EP B's SDP)
ACK
PPL EVENT (w/ SIP EP B's SDP)
CPE (Answer)
DSP Service Request
DSP Service ACK
CONNECT w/ DATA (A, B: w/ SIP EP B's SDP)
CONNECT w/ DATA ACK
200 OK (w/B's SDP)
ACK
RE-INVITE (w/ SIP EP B's SDP, CSP's RFC 2833 SDP)
200 OK
ACK
2-way RTP Voice Path
1-way RFC 2833/RTP Path
RFC 2833 Digits
CPE (Digits)

Figure 6-7 Call Agent RFC 2833 multi-unicast in SIP Softswitch Environment

In the following diagram, the CSP uses an SDP message to re-invite the Media Gateway (via the softswitch) to offer two media streams:

- audio to/from the remote end-point
- telephone event - RFC 2833 multi-unicast

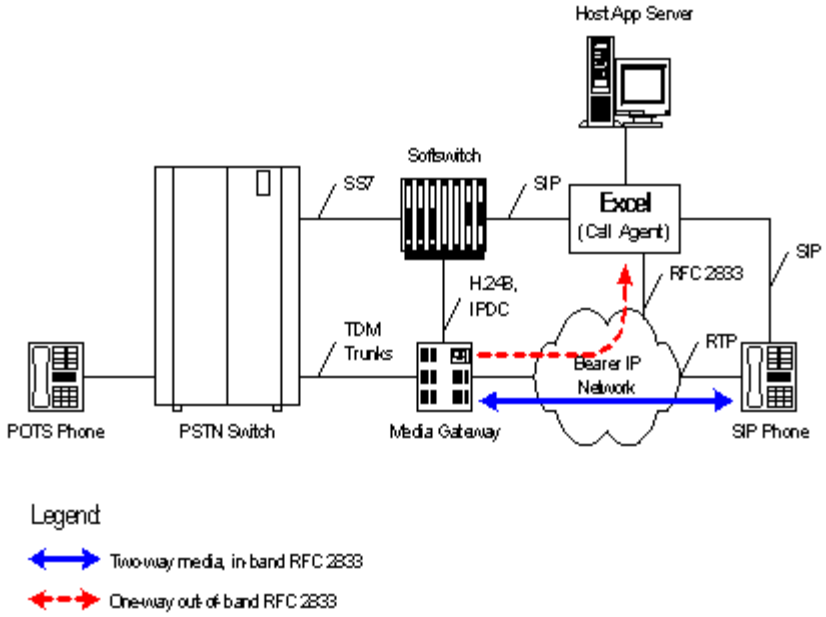


Figure 6-8 Call Agent RFC 2833 multi-unicast in SIP Gateway Environment

In the following diagram, the CSP uses an SDS message to re-invite the SIP Gateway to offer two media streams:

- audio to/from the remote end-point

- telephone event - RFC 2833 multi-unicast

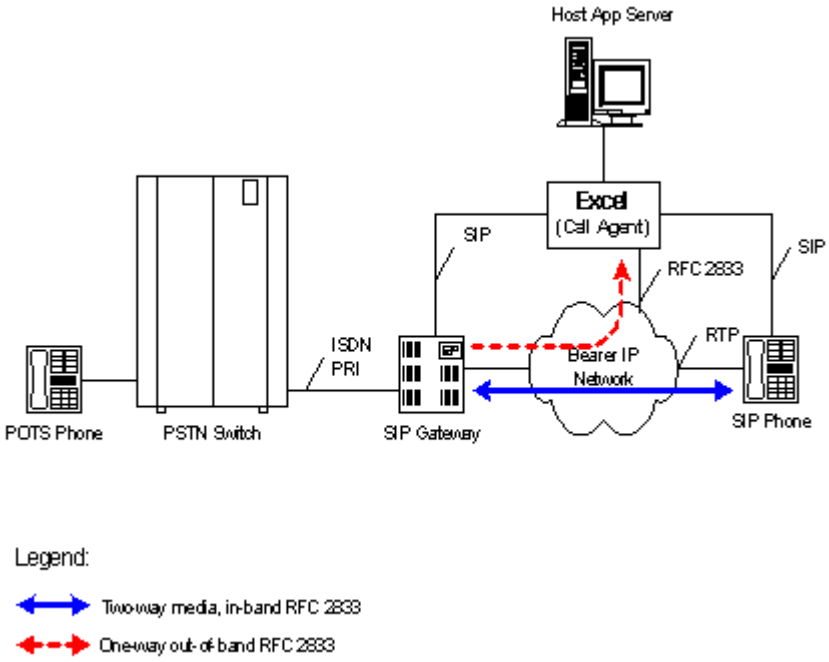


Figure 6-9 Call Agent RFC 2833 multi-unicast in SIP End-Point Environment

In the following diagram, the CSP uses an SDS message to re-invite the ingress SIP end-point to offer two media streams:

- audio to/from the remote end-point

- telephone event - RFC 2833 multi-unicast

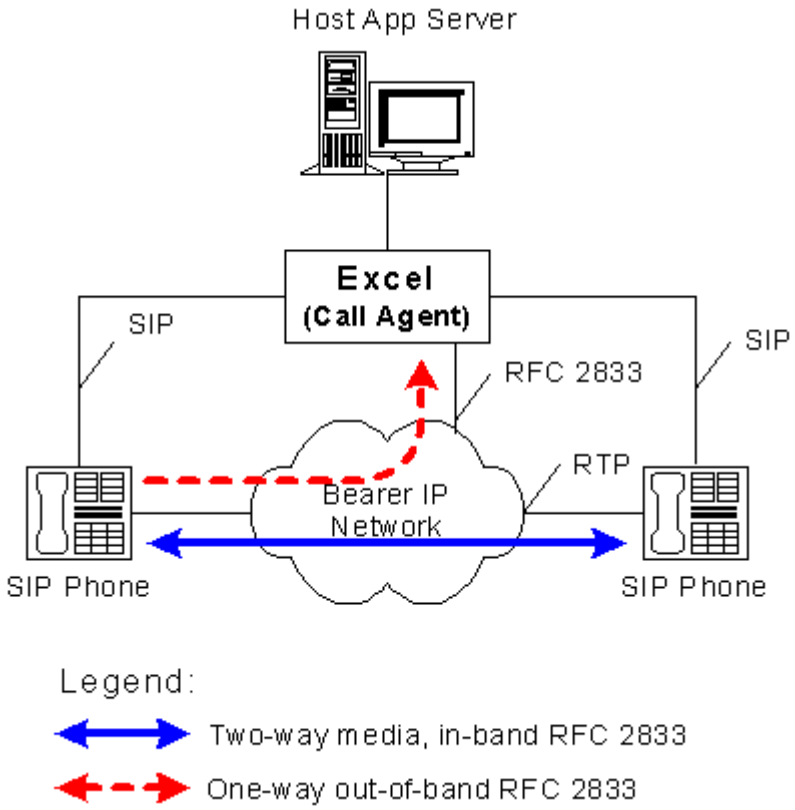
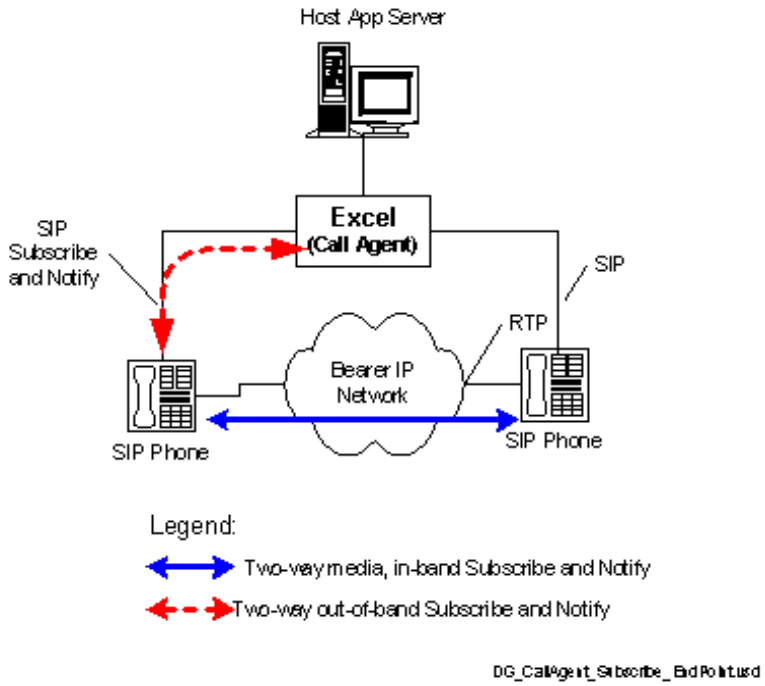


Figure 6-10 Call Agent Subscribe and Notify in SIP End-Point Environment

In the following diagram, the CSP uses an SDS message to re-invite the ingress SIP end-point to offer two media streams:

- audio to/from the remote end-point

- telephone event - Subscribe and Notify. The CSP is the subscriber and the remote end point is the notifer.



DG_CallAgent_Subscribe_EndPoint.vsd

7 Interworking

Purpose This chapter provides information about Dialogic's implementation of Interworking. If the information in this chapter does not address your specific Interworking issues, please contact Dialogic Technical Support.

Introduction to Interworking

Definition of Interworking

Interworking is often defined as the controlled transfer of signaling information between different signaling systems.

In the Dialogic environment, interworking is the ability of a call to enter the CSP using one network protocol and to leave the CSP using a different network protocol with correct end-to-end operation.

Mechanisms

The following are the major mechanisms designed to facilitate interworking:

- Network Protocol Data Intelligence (NPDI) is a call processing feature that analyzes, generates, and converts network signaling information.
- Universal Protocol Data Format (UPDF) is a pre-defined protocol-independent data representation that facilitates the translation of layer protocols.

Responsibilities of Host Application Provider

While the CSP provides mechanisms to facilitate interworking, the host application provider or system integrator must do the following:

- identify the interworking requirements
- implement required translations/mapping
- verify the correct end-to-end operation

Configuring Interworking

Overview Follow the steps below to configure two CSPs for interworking. These steps assume knowledge of SS7 and IP configuration.

The following term is used below.

Universal Flavor - Indicates that the Universal Protocol is not using any user defined TLVs to communicate any network-side protocol information which is not represented in the Universal Protocol specifications.

CSP 1

1. Assign the Logical Node ID.
2. De-assign spans.
3. Send the *Product License Download* message for the appropriate license (for example SIP).
4. Assign the IP Network Interface card's spans and channels.
5. Configure interworking for all the designated SS7 CICs by setting the L4 Host Interworking configuration byte 0x0A as follows:

```
00 17 00 D7 00 00 FF 01 02 0D 03 00 01 00 0D 03 00 05 17
ØØ 61 Ø1 Ø1 ØA Ø1
```

Where:

00 61 = L4 Channel Management

01 = Entity

01 = Number of configuration bytes

0A = L4 Host Interworking

01 = Enable

See CH - Channel Management (0x61) in the *API Developer's Guide: Overview* for more detail.

6. Configure Layer 4 interworking conversion from SS7 to Universal format by setting configuration bytes for Host/Network Protocol ID and flavor in the L4 Interworking component 0x84.

```
00 1D 00 D7 00 00 FF 01 02 0D 03 00 01 00 0D 03 00 05 17
00 84 01 04 03 01 04 02 14 05 15 01
```

Where:

00 84 = L4 Interworking Component

01 = Entity

04 = Number of Configuration Bytes

03 = Local Network Protocol ID

01 = SS7

04 = Local Network Protocol Flavor

02 = ISUP ITU

14 = Host Protocol ID (host interface to mimic for host interworking)

05 = Universal Protocol

15 = Host Protocol Flavor (host interface to mimic for host interworking)

01 = Universal Flavor (indicates that the Universal Protocol is not using any user defined TLVs to communicate any network side protocol information which is not represented in the Universal Protocol specifications.)

See *PPL Component for Interworking (0x84)* in the *API Developer's Guide: Overview* for more detail.

7. Configure the SS7 stack, link set, links, routes, and CICs.
8. Send the route tables.
9. Bring spans, SS7 links, and channels in service.
10. Using the *VoIP Protocol Configure* 0x00EE configure the appropriate TLVs to enable SIP, UPDF tunneling, and remote host (set as CSP 2).

CSP 2

1. Assign Logical Node ID.
2. Deassign spans.
3. Send the *Product License Download* message for the appropriate license (for example SIP).

4. Configure interworking for all the designated SS7 CICs by setting the L4 Host Interworking configuration byte 0x0A as follows:

   00 17 00 D7 00 00 FF 01 02 0D 03 00 01 00 0D 03 00 05 17 00 61 01 01 0A 01

5. Configure Layer 4 interworking conversion from SS7 to Universal format by setting the configuration bytes for Host/Network Protocol ID and flavor in the L4 Interworking component 0x84.

   00 1D 00 D7 00 00 FF 01 02 0D 03 00 01 00 0D 03 00 05 17 00 84 01 04 03 01 04 02 14 05 15 01

6. Configure the SS7 stack, link set, links, route, and CICs.
7. Send route tables.
8. Bring spans, SS7 links, and channels in service.
9. Using the *VoIP Protocol Configure (0x00EE)* message configure the appropriate TLVs to enable SIP, UPDF tunneling, and remote host.

SIP-UPDF Tunneling

Overview SIP-UPDF tunneling carries PSTN signaling across two CSPs with SIP. The *Request for Service* message presented to the host contains TLVs in UPDF format. SIP UPDF is proprietary to Dialogic as it uses proprietary Multipurpose Internet Mail Extensions (MIME) multipart.

Sample SS7 ISUP information is encapsulated within the SIP messages that flow across each CSP. The following is a sample of the message encapsulation:

```
Content-Type: Application/Multipart
Content-Length: 482

Content-Type: multipart/mixed; boundary="de5=_78=_e5d"

--"de5=_78=_e5d"
Content-Type: application/sdp
v=0
o=sip 14999 14999 IN IP4 135.119.55.42
s=SIP_Call
c=IN IP4 135.119.55.48
t=0 0
m=audio 14980 RTP/AVP 18
a=rtpmap:18 G729/8000

--"de5=_78=_e5d"
Content-Type: application/SS7 ISUP; version:EXCEL-UPDF
Content-Transfer-Encoding: binary
03 00 33 00 67 00 0c 27 4e 00 02 00 05 27 25 00 02 21 00
   27
46 00 01 02 27 17 00 08 01 01 0a 86 25 00 00 12 27 18 00
   0a
01 10 81 0a 0a 31 35 55 12 12 27 7d 00 07 00 01 00 06 00
   01
00 27 61 00 08 00 00 00 04 01 00 00 00 27 92 00 04 87 77
   37
30 27 93 00 04 00 00 3a 84 27 b0 00 02 01 12 27 b1 00 02
   01
02 27 7e 00 03 08 00 00
--"de5=_78=_e5d"--
```

Tunneling UPDF TLVs

Two Methods for Tunneling

Two methods are available to tunnel the underlying UPDF TLVs from one CSP to another:

- Configure the SIP stack to tunnel all outbound INVITE(s) to include the TLVs.
- Request the UPDF TLV to be included on a per-outbound-call basis.

The first method involves using the *VoIP Configure* message. The second method involves adding the SIP Tunnel Type TLV (0x2936) to the *Route Control* message.

All Calls Contain UPDF TLVs

This example shows how to form a *VoIP Configure* message to enable tunneling of UPDF data for all SIP calls.

***VoIP Configure* message:**

```
H->X
    [00 27 00 ee 00 0f ff 00 00 00 00 05 01 c8 00 01 04 02
   6a 00 01 04 02 6b 00 01 02 02 64 00 04 0a 0a 0c 21  02
   70 00 02 00 02]
```

Example: SIP message from the SIP stack:

```
INVITE sip:10.10.12.33:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.12.32
To: 3600<sip:3600@10.10.12.33:5060>
From: 3300<sip:3300@10.10.12.32:5060>;tag=5186492ec
Contact: 3300<sip:3300@10.10.12.32:5060>
User-Agent: Excel/8.0.76
Call-ID: EXCEL-CSP255.206.236.250@10.10.12.32
CSeq: 1 INVITE
Content-Type: Application/Multipart
Content-Length: 435

Content-Type: multipart/mixed; boundary="ec=_10e=_1fa"

--"ec=_10e=_1fa"
Content-Type: application/SDP
v=0
o=sip 482 482 IN IP4 10.10.12.32
s=SIP_Call
c=IN IP4 10.10.12.36
t=2208989036 0
m=audio 10072 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

```
--"ec=_10e=_1fa"
Content-Type: application/SS7 ISUP; version:EXCEL-UPDF
Content-Transfer-Encoding: binary
03 00 33 00 39 00 07 27 17 00 05 10 00 04 36 00 27 18 00
   07 10 00 04 37 04 33 00 27 92 00 04 0a 0a 0c 24 27 93
   00 04 00 00 27 58 27 b0 00 02 01 02 27 b1 00 02 01 01
   27 7e 00 03 08 00 00
```

Important! In the above MIME encoded message, the Hex ASCII data that appears after the `Content-Transfer-Encoding:` binary line is actually transmitted in binary, but is listed here as hex ASCII so you can read it.

Route Control message sent from the host:

The following is the actual call being placed with a *Route Control* message, and received with the RFS.

```
H->X
00 4c 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00 19
   00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00 01
   0b 00 65 00 02 00 00 03 00 33 00 1d 00 03 27 7e 00  03
   08 00 01 27 17 00 05 10 00 04 36 00 27 18 00 07 10  00
   04 37 04 33 00
```

Example: SIP message from the SIP stack:

```
INVITE sip:10.10.12.33:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.12.32
To: 3600<sip:3600@10.10.12.33:5060>
From: 3300<sip:3300@10.10.12.32:5060>;tag=5188342109d
Contact: 3300<sip:3300@10.10.12.32:5060>
User-Agent: Excel/8.0.76
Call-ID: EXCEL-CSP255.206.4253.370@10.10.12.32
CSeq: 1 INVITE
Content-Type: Application/Multipart
Content-Length: 455

Content-Type: multipart/mixed; boundary="109d=_186=_1223"

--"109d=_186=_1223"
Content-Type: application/SDP
v=0
o=sip 4379 4379 IN IP4 10.10.12.32
s=SIP_Call
c=IN IP4 10.10.12.36
t=2208993053 0
m=audio 10072 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

```
--"109d=_186=_1223"
Content-Type: application/SS7 ISUP; version:EXCEL-UPDF
Content-Transfer-Encoding: binary
03 00 33 00 3f 00 08 27 17 00 05 10 00 04 36 00 27 18 00
   07
10 00 04 37 04 33 00 29 36 00 02 00 02 27 92 00 04 0a 0a
   0c 24 27 93 00 04 00 00 27 58 27 b0 00 02 01 02 27 b1
   00 02 01 01 27 7e 00 03 08 00 00
```

Important! In the above MIME encoded message the Hex ASCII data that appears after the "Content-Transfer-Encoding: binary" line is actually transmitted in binary, but listed here as Hex ASCII so you can read it.

Received RFS message with UPDF encoded data

All TLVs are in UPDF format.

X->H

```
00 a5 00 2d 00 04 ff 00 01 0d 03 00 01 02 00 33 01 03
00 33
00 91 00 10 27 4e 00 02 00 05 27 7e 00 03 08 0000 29
19 00 05 33 36 30 30 00 29 1b 00 0c 31 30 2e 31 30 2e
31 32 2e 33 33 00 29 1c 00 04 00 00 13 c4 29 23 00 05
33 33 30 30 00 29 25 00 0c 31 30 2e 31 30 2e 31 32 2e
33 32 00 29 26 00 04 00 00 13 c4 27 18 00 07 10 00 04
37 04 33 00 27 17 00 05 10 00 04 36 00 27 94 00 04 0a
0a 0c 24 27 95 00 04 00 00 27 58 27 b0 00 02 01 02 27
b1 00 02 01 01 27 92 00 04 0a 0a 0c 24 27 93 00 04 00
00 27 58
```

Enabling UPDF TLVs per Call

The following is an example of a Per Call request of a UPDF encoded message.

VoIP Configure message

This message does not configure the UPDF tunnel mode.

H->X

```
00 21 00 ee 00 0f ff 00 00 00 00 04 01 c8 00 01 04 02
6a 00 01 04 02 6b 00 01 02 02 64 00 04 0a 0a 0c 21
```

Route Control message from the host

At the end of the sample *Route Control* message below the TLV to enable UPDF is 0x2936 and value is 0x02.

H->X

```
00 52 00 e8 00 00 ff 00 01 29 02 ff fe 02 03 00 1e 00
19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
01 0b 00 65 00 02 00 00 03 00 33 00 23 00 04 27 7e 00
03 08 00 01 27 17 00 05 10 00 04 36 00 27 18 00 07 10
00 04 37 04 33 00 29 36 00 02 00 02
```

This message results in the following SIP message sent from the SIP stack with MIME encoded Session Description Protocol (SDP).

```
INVITE sip:10.10.12.33:5060 SIP/2.0
Via: SIP/2.0/UDP 10.10.12.32
To: 3600<sip:3600@10.10.12.33:5060>
From: 3300<sip:3300@10.10.12.32:5060>;tag=5188342109d
Contact: 3300<sip:3300@10.10.12.32:5060>
User-Agent: Excel/8.0.76
Call-ID: EXCEL-CSP255.206.4253.370@10.10.12.32
CSeq: 1 INVITE
Content-Type: Application/Multipart
Content-Length: 455

Content-Type: multipart/mixed; boundary="109d=_186=_1223"

--"109d=_186=_1223"
Content-Type: application/SDP
v=0
o=sip 4379 4379 IN IP4 10.10.12.32
s=SIP_Call
c=IN IP4 10.10.12.36
t=2208993053 0
m=audio 10072 RTP/AVP 0
a=rtpmap:0 PCMU/8000

--"109d=_186=_1223"
Content-Type: application/SS7 ISUP; version:EXCEL-UPDF
Content-Transfer-Encoding: binary

03 00 33 00 3f 00 08 27 17 00 05 10 00 04 36 00 27 18 00
   07
10 00 04 37 04 33 00 29 36 00 02 00 02 27 92 00 04 0a 0a
   0c
24 27 93 00 04 00 00 27 58 27 b0 00 02 01 02 27 b1 00 02
   01
01 27 7e 00 03 08 00 00
--"109d=_186=_1223"--
```

Important! In the above MIME encoded message the Hex ASCII data which appears after the “Content-Transfer-Encoding: binary”

line is actually transmitted in binary but listed here as Hex ASCII so you can read it.

Received RFS message with UPDF encoded data.

```
X->H
      00 ab 00 2d 00 00 ff 00 01 0d 03 00 01 00 00 33 01 03
      00 33
      00 97 00 11 27 4e 00 02 00 05 27 7e 00 03 08 00 00 29
      19 00 05 33 36 30 30 00 29 1b 00 0c 31 30 2e 31 30 2e
      31 32 2e 33 33 00 29 1c 00 04 00 00 13 c4 29 23 00 05
      33 33 30 30 00 29 25 00 0c 31 30 2e 31 30 2e 31 32 2e
      33 32 00 29 26 00 04 00 00 13 c4 27 18 00 07 10 00 04
      37 04 33 00 27 17 00 05 10 00 04 36 00 27 94 00 04 0a
      0a 0c 24 27 95 00 04 00 00 27 58 27 b0 00 02 01 02 27
      b1 00 02 01 01 29 36 00 02 00 02 27 92 00 04 0a 0a 0c
      24 27 93 00 04 00 00 27 58
```

UPDF Example

Tunnel SS7 via Ethernet

In the following example, the objective is to tunnel SS7 signaling information from one Signaling Point to another via Ethernet. SIP envelops all SS7 ISUP messages in MIME encoded SDP.

The CSP can handle this scenario because it supports traditional PSTN protocols and VoIP protocols such as SIP.

Figure 7-1 Network for Example

In this network, you can configure SS7 channels on the CSP to report incoming SS7 ISUP calls to either:

- Report an *Request for Service* message to the host
- Perform Layer 4 routing to the SIP channels

In either case, you should configure the NPDI Interworking on the SS7 channels on the CSP #1. This way, if a call is reported to the host, the RFS will have UPDF TLVs.

If the host is required to route the call to the SIP side, the host needs to:

- Examine the *Request for Service* message and determine that it is an SS7 ISUP call.
- Issue a *Route Control* message that will have all pertinent UPDF TLVs related to SS7 ISUP parameters so that all those TLVs can carry MIME encoded DSP in the *Invite* message

Sample RFS for Incoming Call

The following sample *Request for Service* message for an incoming call assumes NPDI Interworking is enabled on SS7 channels.

```
 00 5f 00 2d 00 01 01 00 01 0d 03 00 04 01 01 33 01 03 00 33
00 4b 00 08 27 4e 00 02 00 05 27 25 00 02 21 00 27 46
00 01 02 27 17 00 08 01 01 0a 86 25 00 00 12 27 18 00
0a 01 10 81 0a 0a 31 35 55 12 12 27 7d 00 07 00 01 00
06 00 01 00 27 61 00 08 00 00 00 04 01 00 00 00 27 7e
00 03 01 02 02
```

Sample Invite Message

The host needs to issue a *Route Control* message that embeds all of the TLVs required for this RFS. The UPDF tunneled *Invite* message goes out to the other CSP.

The corresponding *Invite* message appears as follows:

```
1 -SENT To 135.119.55.45:5060 at 3557
INVITE sip:8625000012@135.119.55.45:5060 SIP/2.0
Via: SIP/2.0/UDP 135.119.55.42
To: 8625000012<sip:8625000012@135.119.55.45:5060>
From:
   3135551212<sip:3135551212@135.119.55.42:5060>;tag=1745
   2195de5
Call-ID: EXCEL-CSP1.6d1.3557.120@135.119.55.42
Contact: 3135551212<sip:3135551212@135.119.55.42:5060>
User-Agent: Excel/82.0.10
Supported: timer
Session-Expires: 1800
Min-SE: 300
CSeq: 1 INVITE
Content-Type: Application/Multipart
Content-Length: 482

Content-Type: multipart/mixed; boundary="de5=_78=_e5d"

--"de5=_78=_e5d"
Content-Type: application/sdp
v=0
o=sip 14999 14999 IN IP4 135.119.55.42
s=SIP_Call
c=IN IP4 135.119.55.48
t=0 0
m=audio 14980 RTP/AVP 18
a=rtpmap:18 G729/8000

--"de5=_78=_e5d"
Content-Type: application/SS7 ISUP; version:EXCEL-UPDF
Content-Transfer-Encoding: binary

03 00 33 00 67 00 0c 27 4e 00 02 00 05 27 25 00 02 21 00
   27
46 00 01 02 27 17 00 08 01 01 0a 86 25 00 00 12 27 18 00
   0a
```

```
01 10 81 0a 0a 31 35 55 12 12 27 7d 00 07 00 01 00 06 00
   01
00 27 61 00 08 00 00 00 04 01 00 00 00 27 92 00 04 87 77
   37
30 27 93 00 04 00 00 3a 84 27 b0 00 02 01 12 27 b1 00 02
   01
02 27 7e 00 03 08 00 00

--"de5=_78=_e5d"--
```

8 Routing VoIP Calls

Overview

Purpose This chapter provides information to help you configure the CSP to route VoIP calls.

The following basic scenarios are explained in this chapter:

- Routing VoIP calls controlled by SIP and H.323 signaling
  - Using the *Route Control* (0x00E8) message
  - Using the *Outseize Control* (0x002C) message
- Clear Channel VoIP calls (without SIP or H.323 signaling)

For information on routing Time Division Multiplex (TDM) calls, refer to the *Call Routing* chapter in the *Developer's Guide: Overview.*

Channel Management PPL Component for VDAC - 0x0061

The Channel Management (CH) component interfaces with the CSP Signaling layer (Layer 3) for call control, and to the host (Layer 5) for call processing. All communication between the host and a channel passes through the CH component. A CH state machine is statically instantiated once for each channel.

PPL Configuration Bytes

The table below shows the PPL Configuration Byte values for the CH component that apply to VoIP Routing.

Byte	Description	Values
2	Answer Supervision	0x00 - Propagate Answer to Distant End 0x01 - Notify Host of Answer 0x02 - Propagate Answer to Distant End and Notify Host of Answer 0x03 - No Answer Supervision - no propagation of answer or notification (default = 0xFF, Use L3-provided value)
3	Local End Release Mode	0x00 - Release 0x01 - Park (default = 0xFF, Use L3-provided value)
4	Distant End Release Mode	0x00 - Release 0x01 - Park (default = 0xFF, Use L3-provided value)

Routing SIP and H.323 Calls Using *Route Control* Message

Overview

This section explains routing VoIP calls that are controlled by SIP and H.323 where your host application uses the *Route Control* message to route the outbound side of the call. Using the *Route Control* message allows the host application to leverage Layer 4 Router functionality thereby relieving the host application from making the span/channel assignments on the IP Network Interface card modules.

Router Component

The Router component (0x64) is a PPL state machine that references a database of resources (the Route Table) to route calls. It is separated from the call processing logic in the CSP. Refer to *RTR Router (0x64)* in the *Developer's Guide: Overview* for PPL information including configuration bytes and general purpose registers.

Route Tables

A Route Table is a special type of PPL table (Type 3) containing only routing criteria data. The router accesses the table to obtain a destination address or criteria data type (0x08).

Route Table entries are arranged into route groups that include one criteria type.

Resource Group Table

A Resource Group table is a special type of PPL table (Type 4) containing only Resource Group data. When the Route Table entry data yields a specific Resource Group ID, the router accesses the table to obtain a destination address.

Resource groups include the span/channel and the IP address of the IP Network Interface card.

API Messages

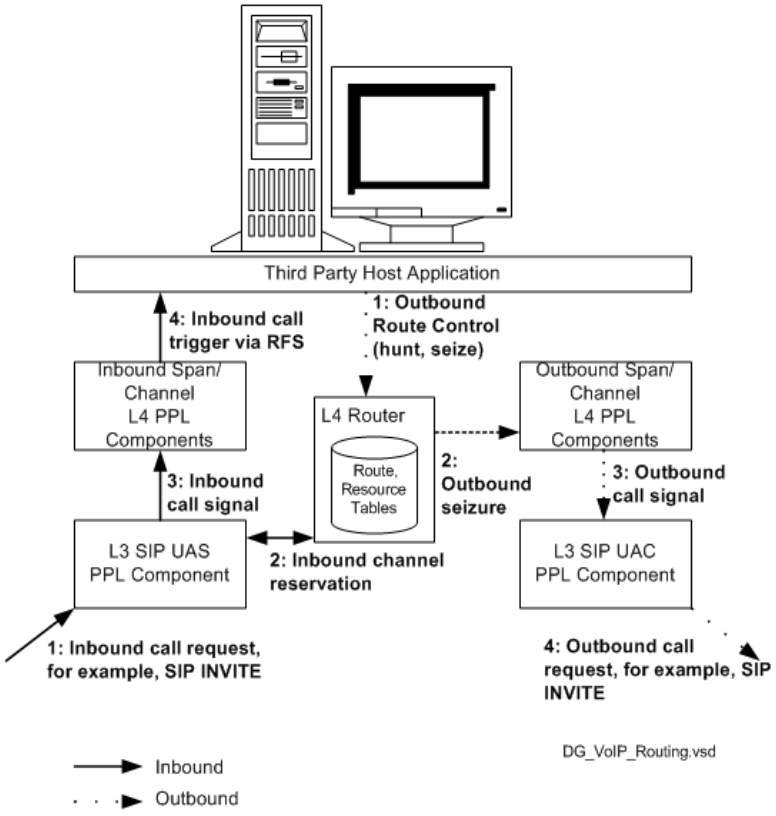
Use the following API messages to download the Route Tables and Resource Group tables to the CSP with a .cfg file:

- *PPL Table Initiate Download* message (0x00D5)
- *PPL Table Download* message (0x00D6)

Routing Process

The following diagram shows a call controlled by SIP signaling and identifies each step in the routing process. Each step is further explained after the diagram.

Figure 8-1 Routing Overview - Inbound and Outbound



Steps - Inbound

1. The INVITE message comes into the Layer 3 SIP User Agent Server on the CSP.
2. The SIP stack interfaces with the Router component to reserve the inbound channel.
3. The SIP User Agent Server sends the inbound call signal to the Inbound Span/Channel Layer 4 PPL components.
4. The Layer 4 PPL components send an inbound call trigger to the host application in the *Request for Service with Data* message. The inbound call is completed.

Steps - Outbound

1. The host application sends the *Route Control* message to the Router component to hunt and seize the outbound resources.
2. The Router component sends an outbound seizure to the outbound span/channel Layer 4 PPL components.
3. The Layer 4 PPL components send the outbound call signal to the Layer 3 SIP User Agent Client.

4. The SIP User Agent Client initiates the outbound call using a SIP INVITE message.

Examples This section provides examples of how the CSP routes a SIP or H.323 call including:

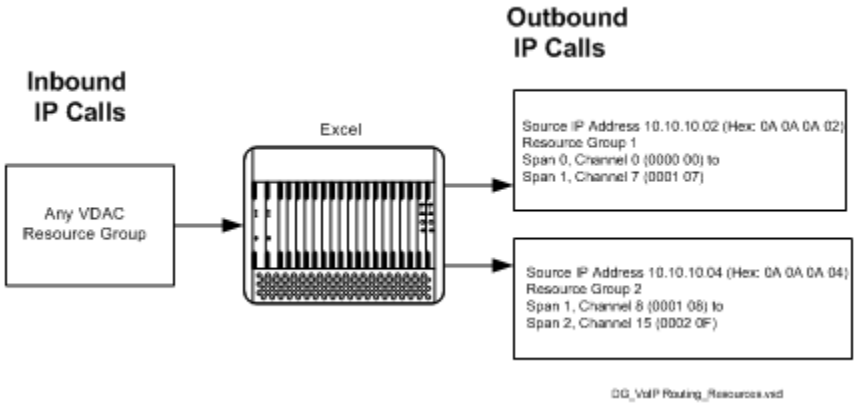
- A high-level diagram of the CSP Routing
- An example of Route Tables and Resource Groups created using the Converged Services Administrator (CSA) and downloaded to the CSP with the *PPL Download* message. It is a simplified case in which the CSP has only one VDAC-ONE card with two VoIP modules.
- A *Route Control* message with the relevant portions annotated
- Detailed steps of how the *Route Control* message uses the Route Table and Resource groups. These steps refer to specific lines in the Route Table and Resource Group.
- Additional examples of Route Tables and Resource Groups that show the various configurations available. Annotations call out the Resource Groups and the Route Tables.

Using Resource Groups

The following diagram shows, at a high level, how both inbound and outbound calls use Resources Groups. The information in this diagram corresponds to the information in the example that follows. That example contains the specific Route Tables and Resource Groups used to configure this scenario.

Following that is the *Route Control* message as well as the steps that the message completes to select the resources.

Figure 8-2 Resource Groups



One VDAC-ONE Card with Two VoIP Modules

In the following example, two Resource Groups are set up to correspond to the two VoIP modules on the VDAC-ONE card. Annotations are included to make it easier to read.

Resource Group

```
Line 1
00 21 00 d5 00 00 01 00 64 04 01 00 00 00 44 72 65 73 2f 67 72 6f 75 70
20 74 61 62 6c 65 00 00 00 00 00
```

Resource Group ID

```
Line 2
00 2b 00 d6 00 00 01 00 64 04 01 00 01 00 01 00 01 00 01 00 0c
01 02 0d 03 00 00 00 0d 03 00 01 07 00 01 00 08 01 68 00 04 0a 0a 0a 02
```

Span and Channel Range AIB

VoIP Module IP Address

Resource Group ID

```
Line 3
00 2b 00 d6 00 00 01 00 64 04 01 00 02 00 01 00 01 00 01 00 0c
01 02 0d 03 00 01 08 0d 03 00 02 0f 00 01 00 08 01 68 00 04 0a 0a 0a 04
```

Span and Channel Range AIB

VoIP Module IP Address

Route Table

```
Line 4
00 21 00 d5 00 00 01 00 64 03 01 00 00 00 83 72 6f 75 74 65 72 5f 74 61
62 6c 65 00 00 00 00 00 00 00 00
```

Universal Criteria Type
Source IP Address

```
Line 5
00 35 00 d6 00 00 01 00 64 03 01 00 01 c0 01 00 01 00 01 27 92 01 00 0e
01 00 00 00 00 00 0a 0a 0a 02 ff ff ff ff 00 02 00 0d 00 15 00 02 00 01
00 63 00 03 01 00 01
```

Universal Criteria Type
Source IP Address

```
Line 6
00 35 00 d6 00 00 01 00 64 03 01 00 02 c0 01 00 01 00 01 27 92 01 00 0e
01 00 00 00 00 00 0a 0a 0a 04 ff ff ff ff 00 02 00 0d 00 15 00 02 00 02
00 63 00 03 01 00 02
```

VoIP Physical

```
Line 7
00 34 00 d6 00 00 01 00 64 03 01 00 03 c0 01 00 01 00 01 00 65 01 00 02
00 00 00 04 00 18 00 62 00 01 01 00 15 00 02 00 01 00 15 00 02 00 02 00
63 00 03 01 00 03
```

Resource Group TLV

ID

Resource Group TLV

ID

Route Control Message

The *Route Control* message uses two or more ICBs to make an outbound SIP or H.323 call. The Generic PPL ICB (0x001E) controls the span-channel and VoIP module selection through the Router component. Below is an example of a *Route Control* message. The first ICB, which is explained, could be used for a SIP or H.323 call.

```
00 66 00 e8 00 00 01
00 01 29 02 ff fe - Router AIB - Route Handle FF FE
02 03 00 1e 00 19 - extended ICB with a length of 0x19
00 04              - 4 TLVs to follow
00 13 00 02 00 08 - Routing Method = Criteria Type 0x0008
00 08 00 02 00 65 - Criteria Type TLV = use physical VoIP
   channel
00 0f 00 01 0b    - Router protocol ID = 0x0b
00 65 00 02 00 00 - Physical VoIP Channel

03 00 33 00 37
00 04
29 19 00 08 31 38 37 30 30 30 31 00
29 23 00 0b 42 6f 74 74 6f 6d 2d 43 53 50 00
29 1b 00 0f 31 33 35 2e 31 31 39 2e 35 31 2e 31 38 37 00
27 7e 00 03 08 00 00
```

How the *Route Control* Message Uses Route Tables

The following steps explain how the *Route Control* message uses the Route Tables and Resource Groups to route a call.

1. The AIB (0x29) in the *Route Control* message directs the call to the Router component.
2. The Routing Method TLV (0x0013) specifies to use a criteria type. The Criteria Type TLV specifies the criteria to use to search for a route.
3. The *Route Control* message directs the Router component to search the Route Table for any resources that are mapped to physical VoIP channels (value - 0x0065 VoIP Physical).
4. The Router component finds Criteria Type 00 65 in Line 7 of the Route Table. Note that Lines 5 and 6 are not matched because they use different criteria types. There are two Terminating Resource Group TLVs (0x0015) in Line 7. The first contains the Resource Group ID 0001 and the second contains Resource Group ID 00002.
5. These IDs map to Resource Group IDs 00 01 and 00 02 respectively in Lines 2 and 3 of the Resource Group.

6. Line 2 maps VoIP Module IP Address 0a 0a 0a 02 to span 0000, channel 00 through span 0001, channel 07. Line 3 maps the VoIP Module IP Address 0a 0a 0a 04 to span 0001, channel 08 through span 0002, channel 0f.
7. The Router component routes the outbound call based on the VoIP module and Span/Channel.

Additional Examples — One VDAC-ONE Card with Four VoIP Modules

```
Resource Group
00 21 00 d5 00 00 01 00 64 04 01 00 00 00 88 72 65 73 2f
   67 72 6f 75 70 20 74 61 62 6c 65 00 00 00 00 00

00 2b 00 d6 00 00 01 00 64 04 01 00 01 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 00 00 0d 03 00 01 07 00 01 00 08
   01 68 00 04 0a 0a 0a 02

00 2b 00 d6 00 00 01 00 64 04 01 00 02 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 01 08 0d 03 00 02 0f 00 01 00 08
   01 68 00 04 0a 0a 0a 03

00 2b 00 d6 00 00 01 00 64 04 01 00 03 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 02 10 0d 03 00 03 17 00 01 00 08
   01 68 00 04 0a 0a 0a 04

00 2b 00 d6 00 00 01 00 64 04 01 00 04 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 03 18 0d 03 00 04 1f 00 01 00 08
   01 68 00 04 0a 0a 0a 05

Route Table
00 21 00 d5 00 00 01 00 64 03 01 00 00 00 e7 72 6f 75 74
   65 72 5f 74 61 62 6c 65 00 00 00 00 00 00 00 00

00 35 00 d6 00 00 01 00 64 03 01 00 01 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 02 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 01 00 63 00 03 01 00 01

00 35 00 d6 00 00 01 00 64 03 01 00 02 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 03 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 02 00 63 00 03 01 00 02

00 35 00 d6 00 00 01 00 64 03 01 00 03 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 04 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 03 00 63 00 03 01 00 03

00 35 00 d6 00 00 01 00 64 03 01 00 04 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 05 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 04 00 63 00 03 01 00 04
```

```
00 40 00 d6 00 00 01 00 64 03 01 00 05 c0 01 00 01 00 01
   00 65 01 00 02 00 00 00 06 00 24 00 62 00 01 01 00 15
   00 02 00 01 00 15 00 02 00 02 00 15 00 02 00 03 00 15
   00 02 00 04 00 63 00 03 01 00 05
```

One Virtual VDAC-ONE Card

```
Resource Group
00 21 00 d5 00 00 01 00 64 04 01 00 00 00 1a 72 65 73 2f
   67 72 6f 75 70 20 74 61 62 6c 65 00 00 00 00 00

00 23 00 d6 00 00 01 00 64 04 01 00 01 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 00 00 0d 03 00 1f 1f 00 00 00 00

Route Table
00 21 00 d5 00 00 01 00 64 03 01 00 00 00 25 72 6f 75 74
   65 72 5f 74 61 62 6c 65 00 00 00 00 00 00 00 00

00 2e 00 d6 00 00 01 00 64 03 01 00 02 c0 01 00 01 00 01
   00 71 01 00 02 00 00 00 03 00 12 00 62 00 01 01 00 15
   00 02 00 01 00 63 00 03 01 00 02
```

Two VDAC-ONE Cards with Four Modules

```
Resource Group
00 21 00 d5 00 00 01 00 64 04 01 00 00 01 10 72 65 73 2f
   67 72 6f 75 70 20 74 61 62 6c 65 00 00 00 00 00

00 2b 00 d6 00 00 01 00 64 04 01 00 01 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 00 00 0d 03 00 01 07 00 01 00 08
   01 68 00 04 0a 0a 0a 02

00 2b 00 d6 00 00 01 00 64 04 01 00 02 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 01 08 0d 03 00 02 0f 00 01 00 08
   01 68 00 04 0a 0a 0a 03

00 2b 00 d6 00 00 01 00 64 04 01 00 03 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 02 10 0d 03 00 03 17 00 01 00 08
   01 68 00 04 0a 0a 0a 04

00 2b 00 d6 00 00 01 00 64 04 01 00 04 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 03 18 0d 03 00 04 1f 00 01 00 08
   01 68 00 04 0a 0a 0a 05

00 2b 00 d6 00 00 01 00 64 04 01 00 05 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 05 00 0d 03 00 06 07 00 01 00 08
   01 68 00 04 0a 0a 0a 0b
```

```
00 2b 00 d6 00 00 01 00 64 04 01 00 06 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 06 08 0d 03 00 07 0f 00 01 00 08
   01 68 00 04 0a 0a 0a 0c

00 2b 00 d6 00 00 01 00 64 04 01 00 07 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 07 10 0d 03 00 08 17 00 01 00 08
   01 68 00 04 0a 0a 0a 0d

00 2b 00 d6 00 00 01 00 64 04 01 00 08 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 08 18 0d 03 00 09 1f 00 01 00 08
   01 68 00 04 0a 0a 0a 0e

Route Table
00 21 00 d5 00 00 01 00 64 03 01 00 00 01 af 72 6f 75 74
   65 72 5f 74 61 62 6c 65 00 00 00 00 00 00 00 00

00 35 00 d6 00 00 01 00 64 03 01 00 01 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 02 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 01 00 63 00 03 01 00 01

00 35 00 d6 00 00 01 00 64 03 01 00 02 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 03 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 02 00 63 00 03 01 00 02

00 35 00 d6 00 00 01 00 64 03 01 00 03 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 04 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 03 00 63 00 03 01 00 03

00 35 00 d6 00 00 01 00 64 03 01 00 04 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 05 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 04 00 63 00 03 01 00 04

00 35 00 d6 00 00 01 00 64 03 01 00 05 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 0b ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 05 00 63 00 03 01 00 05

00 35 00 d6 00 00 01 00 64 03 01 00 06 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 0c ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 06 00 63 00 03 01 00 06

00 35 00 d6 00 00 01 00 64 03 01 00 07 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 0d ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 07 00 63 00 03 01 00 07
00 35 00 d6 00 00 01 00 64 03 01 00 08 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 0e ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 08 00 63 00 03 01 00 08

00 58 00 d6 00 00 01 00 64 03 01 00 09 c0 01 00 01 00 01
   00 65 01 00 02 00 00 00 0a 00 3c 00 62 00 01 01 00 15
   00 02 00 01 00 15 00 02 00 02 00 15 00 02 00 03 00 15
   00 02 00 04 00 15 00 02 00 05 00 15 00 02 00 06 00 15
   00 02 00 07 00 15 00 02 00 08 00 63 00 03 01 00 09
```

One IP Network Interface Series 2 Card with Two VoIP Modules

```
Resource Group
00 21 00 d5 00 00 01 00 64 04 01 00 00 00 44 72 65 73 2f
   67 72 6f 75 70 20 74 61 62 6c 65 00 00 00 00 00

00 2b 00 d6 00 00 01 00 64 04 01 00 01 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 00 00 0d 03 00 07 1f 00 01 00 08
   01 68 00 04 0a 0a 0a 02

00 2b 00 d6 00 00 01 00 64 04 01 00 02 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 08 00 0d 03 00 0f 1f 00 01 00 08
   01 68 00 04 0a 0a 0a 03

Route Table
00 21 00 d5 00 00 01 00 64 03 01 00 00 00 83 72 6f 75 74
   65 72 5f 74 61 62 6c 65 00 00 00 00 00 00 00 00

00 35 00 d6 00 00 01 00 64 03 01 00 01 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 02 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 01 00 63 00 03 01 00 01

00 35 00 d6 00 00 01 00 64 03 01 00 02 c0 01 00 01 00 01
   27 92 01 00 0e 01 00 00 00 00 00 0a 0a 0a 03 ff ff ff
   ff 00 02 00 0d 00 15 00 02 00 02 00 63 00 03 01 00 02

00 34 00 d6 00 00 01 00 64 03 01 00 03 c0 01 00 01 00 01
   00 65 01 00 02 00 00 00 04 00 18 00 62 00 01 01 00 15
   00 02 00 01 00 15 00 02 00 02 00 63 00 03 01 00 03
```

Two Virtual IP Network Interface Series 2 Cards

```
Resource Group
00 21 00 d5 00 00 01 00 64 04 01 00 00 00 34 72 65 73 2f
   67 72 6f 75 70 20 74 61 62 6c 65 00 00 00 00 00

00 23 00 d6 00 00 01 00 64 04 01 00 01 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 00 00 0d 03 00 1f 1f 00 00 00 00

00 23 00 d6 00 00 01 00 64 04 01 00 02 00 01 00 01 00 01
   00 0c 01 02 0d 03 00 32 00 0d 03 00 51 1f 00 00 00 00

Route Table
00 21 00 d5 00 00 01 00 64 03 01 00 00 00 2b 72 6f 75 74
   65 72 5f 74 61 62 6c 65 00 00 00 00 00 00 00 00

00 34 00 d6 00 00 01 00 64 03 01 00 02 c0 01 00 01 00 01
   00 71 01 00 02 00 00 00 04 00 18 00 62 00 01 01 00 15
   00 02 00 01 00 15 00 02 00 02 00 63 00 03 01 00 02
```

Routing SIP and H.323 Calls Using *Outseize Control* Message

Overview

VoIP applications can use the *Outseize Control* (0x002C) message for outbound SIP and H.323 signaling. This method is required for application developers who:

- are migrating TDM applications to the VoIP environment if the TDM applications already use the *Outseize Control* message
- want to control the selection of spans/channels

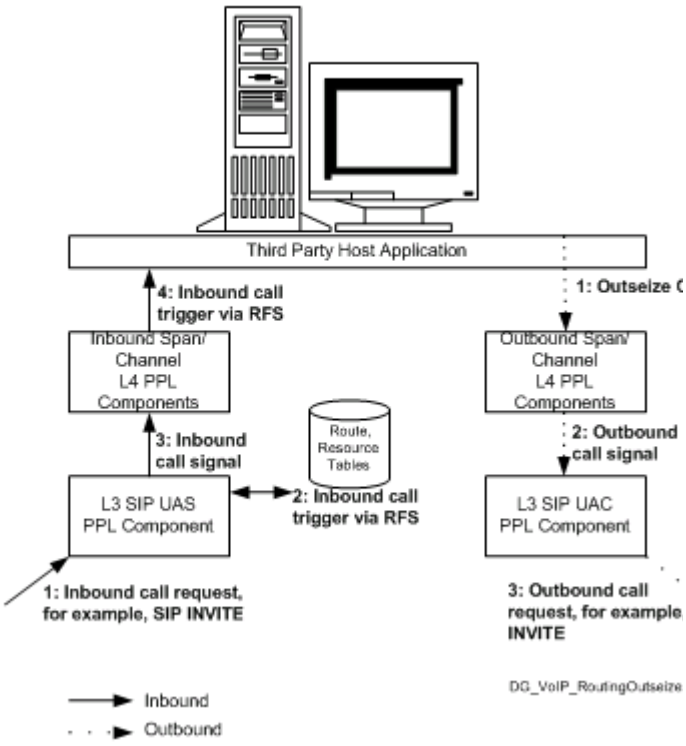
Routing Process

The host application is responsible for selecting the span and channel for each outbound call on a per call basis. The host application also selects the source IP address and port.

> **Important!** If you do not want the host application to be responsible for the channel management, you can use the *Route Control* message instead. Refer to *Routing SIP and H.323 Calls Using Route Control Message.*

The figure below provides an overview of the routing process. Note that the inbound side of the call is the same as described in the previous section. The outbound side involves the *Outseize Control* message rather than the *Route Control* message. The differences are pointed out in the steps following the figure.

Figure 8-3 Routing Process with *Outseize Control* Message



Steps - Outbound

1. The host application sends the *Outseize Control* message to the outbound span/channel Layer 4 PPL components. (Note that the Layer 4 Router component is not used.)

   The *Outseize Control* message includes the span/channel, source IP address and port required to route the call. (For details, refer to Sample *Outseize Control* message following these steps.)
2. The Layer 4 PPL components send the outbound call signal to the Layer 3 SIP User Agent Client.
3. The SIP User Agent Client initiates the outbound call using a SIP INVITE message.

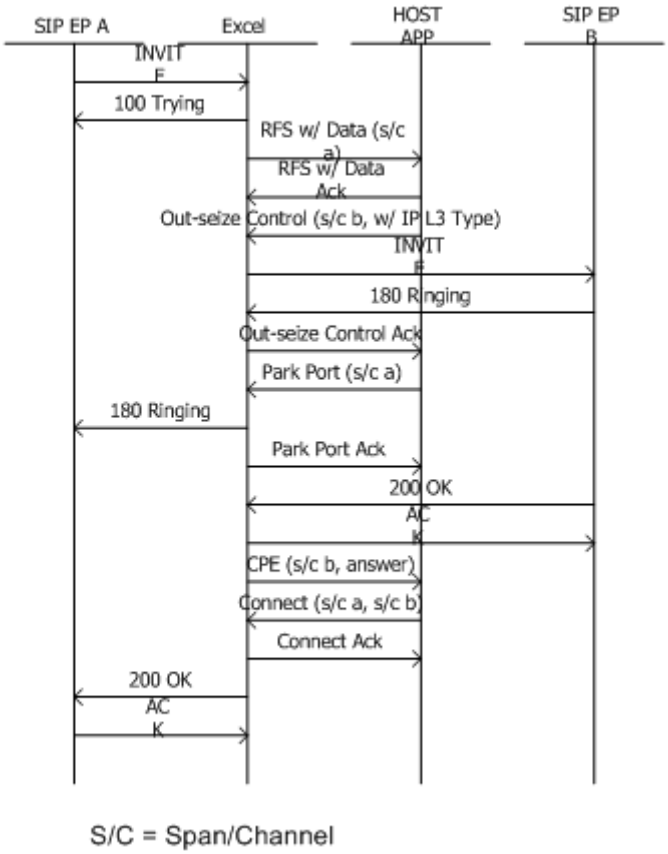
Sample *Outseize Control* Message

In the following example, the *Outseize Control* message contains the AIB that specifies the span and channel used to route this call. The NPDI Universal ICB contains six TLVs that specify information about this call as indicated below. You must include the Remote Side Protocol TLV (0x277E) so the call can be handed to the appropriate Layer 3 protocol. For H.323 calls, you must include the IP Signaling Series 3 Card ID TLV (0x27C1) so the call can be delivered to one of the H.323 IP Signaling Series 3 cards in the CSP.

Important! You must use the NPDI TLVs rather than the legacy TLVs for the *Outseize Control* message to work with SIP and H.323 signaling because they are NPDI-only oriented.

```
00 40 00 2C 00 00 $node_id
00         - address method
01         - number of address elements
0d         - address type = span/channel
03         - data length
00 06 06   - span/channel
01         - number of ICBs to follow
03         - icb type = extended
00 33      - icb subtype
00 2E      - icb data length
00 06      - number of TLVs
27 92 00 04 0a 0a 01 25 - Source IP Address
27 93 00 04 00 00 10 80 - Source IP Port
29 19 00 05 33 33 33 33 00 - SIP To Username
27 7E 00 03 08 00 00 - Remote Side Protocol, SIP
27 B0 00 02 01 01 - RTP Payload Type - Egress, G.711 A-
   Law
27 B1 00 02 01 01 - RTP Payload Size - Egress, 0x01 use
   Basic Packet Rate
```

Figure 8-4 Call Flow - VoIP Routing with *Outseize Control* Message



Routing Clear Channel VoIP Calls

Overview

The following examples show the host application initiating Clear Channel VoIP Calls using the VDAC-ONE card but without running SIP or H.323 software.

CSP Initiated

The following are two examples of API messages that cause the CSP to initiate a clear channel VoIP call. Even though the examples use different API messages, they perform the same function.

***Route Control* message**

```
00 4c 00 e8 00 00 ff 00 01 29 02 ff fe 02 02 1e 16 00 03
   00 13 00 02 00 09 00 0f 00 01 0b 00 09 00 05 0d 03 00
   40 03 03 00 33 00 22 00 04 27 92 00 04 87 77 33 b2 27
   93 00 04 00 00 17 70 27 94 00 04 87 77 33 b7 27 95 00
   04 00 00 17 78
```

***Outseize Control* message**

```
00 34 00 2c 00 00 ff 00 01 0d 03 00 40 03 01 03 00 33 00
   22 00 04 27 92 00 04 87 77 33 b2 27 93 00 04 00 00 17
   84 27 94 00 04 87 77 33 b7 27 95 00 04 00 00 17 88
```

These messages require the host to manage the IP call signaling and to map the VDAC-ONE spans and channels to the correct IP resources (VDAC-ONE IP addresses and ports).

IP Endpoint Signaling

The following example involves host-to-host signaling (also called IP endpoint signaling).

1. A call from the PSTN comes into CSP A, which sends a *Request for Service with Data* message to Host A.
2. Host A notifies Host B (or an endpoint directly) of a call request using any protocol.
3. The call request contains the IP address and port number of a resource on the VDAC-ONE card (those IP addresses and ports must be managed by the host in this example).
4. The Host B side responds to the call request with the Host B side parameters: IP address and port.
5. Host A issues a *Route Control* or *Outseize Control* message to CSP A supplying the local and remote VDAC-ONE IP and port address (as well as the correct span/channel) to initiate a VoIP clear channel call from the VDAC-ONE card.

6. Host B issues a *Route Control* or *Outseize Control* message to CSP B to complete the IP call.

IP Based Routing

Overview

IP Based Routing allows the CSP to separate its physical VoIP resources into resource groups that share a common attribute such as the IP subnet. Previously, all VoIP channels had to act as a single resource pool and assigned connections without regard to the IP address of the media port or remote endpoint. (Throughout this section, “IP media port” refers to either a port on the VDAC-ONE card or IP Network Interface Series 2 card).

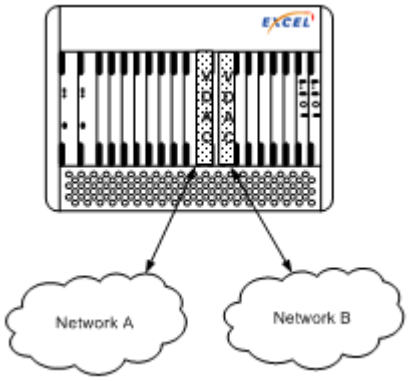
With IP Based Routing, you can prevent a situation where a VoIP port, pulled from a common pool, cannot reach an endpoint because it is on the wrong IP subnet.

We recommend that you use two cards, in any of the following combinations:

- Two IP Network Interface Series 2 cards
- Two VDAC-ONE cards
- One IP Network Interface Series 2 card and one VDAC-ONE card

With IP based routing you can control the resources across the two cards to determine how calls come into and out of the CSP.

Figure 8-5 Two VDAC Cards

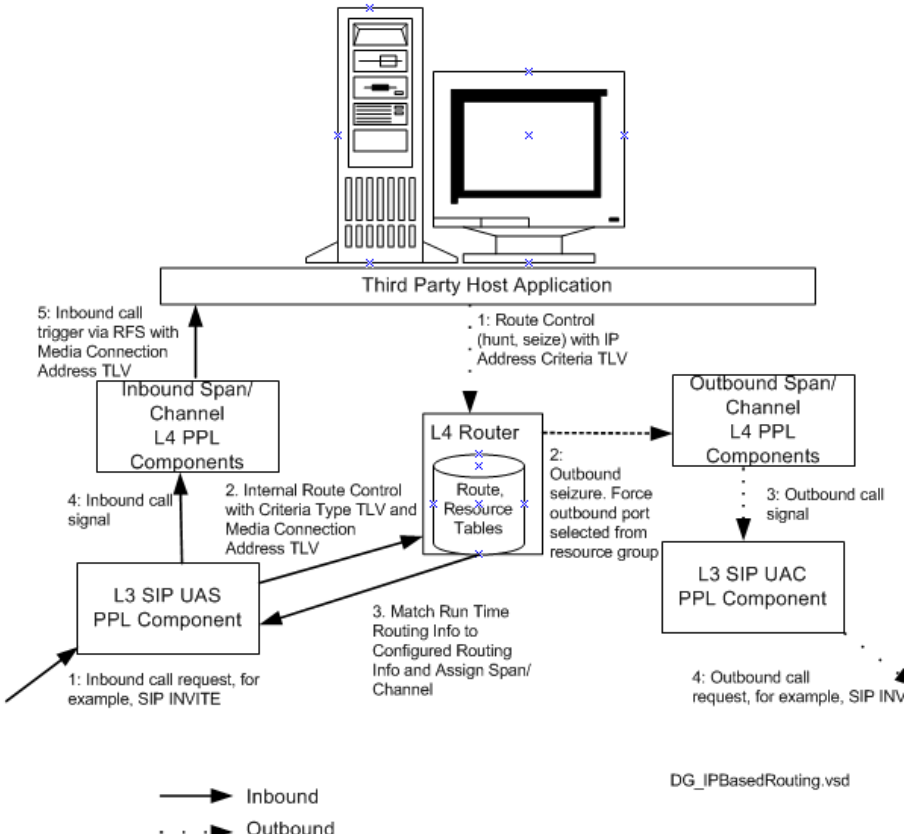


This feature is useful in the carrier environment where different carriers are on different subnets or on different IP networks. You can assign a range of spans to carriers so one carrier does not consume a disproportionate number of available spans.

Routing Process

The following diagram provides an overview of the IP Based Routing process. Each step is explained following the diagram. These steps apply to SIP and H.323 calls and the differences are called out.

Figure 8-6 IP Based Routing Process



Steps - Inbound

1. For SIP calls, the INVITE message comes into the Layer 3 SIP User Agent Server on the CSP. For H.323, the H.225 Setup message comes into Layer 3.
2. The following TLVs in the *VoIP Protocol Configure* (0x00EE) message determine how the SIP stack operates. The combination of TLVs depends on which routing method is used.

Method 1

- Routing Method (0x0013)
  - Value: Criteria Type 0x0008
- Criteria Type (0x0008)
  - Value: 0x2A0E Media Connection Address
- Router Protocol ID (0x000F)
  - Value: 0x0B (default)
- Media Connection Address (0x2A0E).
  Value automatically added by SIP stack from SDP

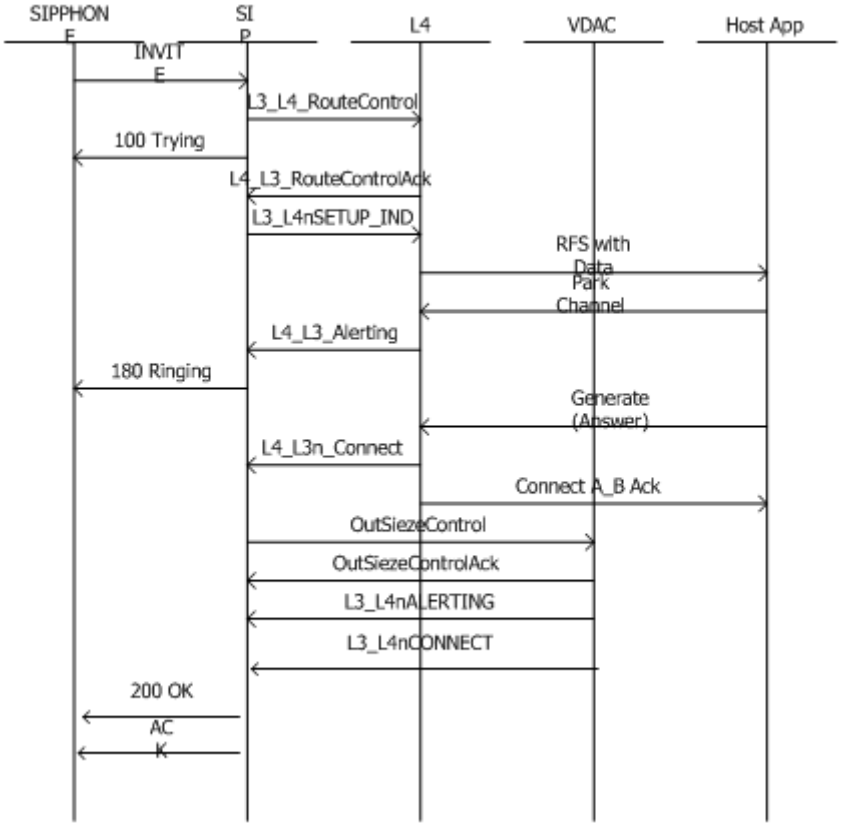
Method 2

- Routing Method (0x00013)
  -Value: Route Group ID 0x0006
- Route Group ID (0x0006)
  - Value: Route Group ID
- Router Protocol ID (0x000F)
  - Value: Router Protocol ID
- Criteria Type (0x0008)
  - Value: 0x2A0E Media Connection Address
- Media Connection Address (0x2A0E)
  Value automatically added by SIP stack from SDP

3. The SIP stack sends the internal *Route Control* message to the Layer 4 Router component. The message uses the Routing Method of Criteria Type. The criteria type is Media Connection Address. The SIP stack extracts the IP address from the “cline” parameter from the SDP of the INVITE message and automatically puts in the Media Connection Address TLV (0x2A0E).
4. The Router component compares this run time information to the configured routing information. If there is a match, the appropriate resource group is hunted. The router returns a VoIP port with the same subnet to Layer 3.
5. The SIP User Agent Server sends the inbound call signal to the Inbound Span/Channel Layer 4 PPL components.
6. The Layer 4 PPL components send an inbound call trigger to the host application in the *Request for Service with Data* message. This message can contain the media IP address of the endpoint if the *VoIP Protocol Configure* (0x00EE) message was sent with the SIP Message Information Mask TLV 0x027F set to include Media Connection Address. For H.323 calls, this TLV is H.323 Message Information Mask TLV (0x02D4). The inbound call is completed.

Figure 8-7 Call Flow - Inbound SIP Call

The following call flow diagram shows an inbound SIP call. H.323 calls are essentially the same. Note that the Criteria type is Media Connection Address.



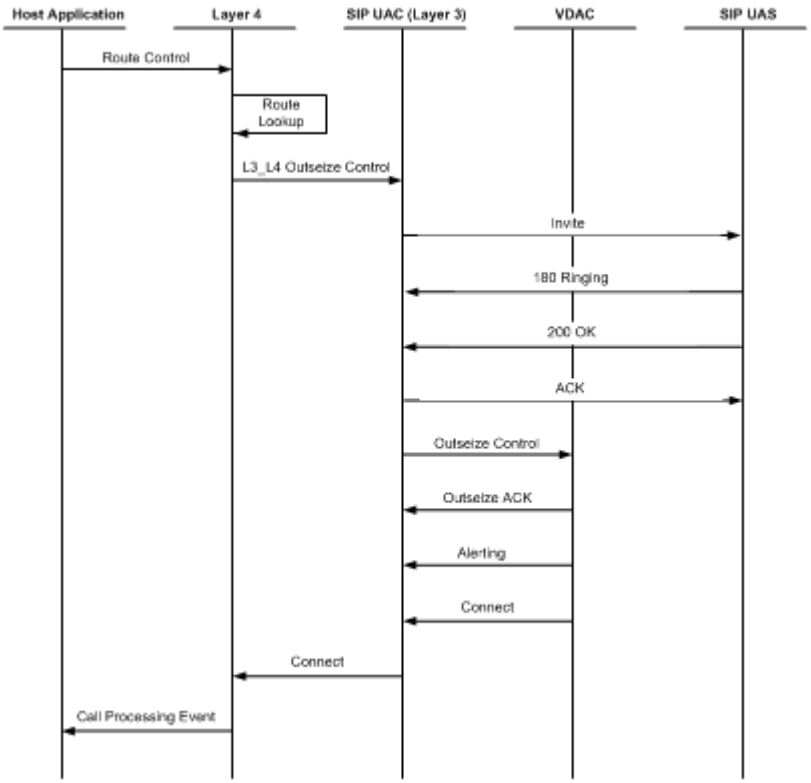
Steps - Outbound

1. The host application sends the *Route Control* message to the Router component to hunt and seize the outbound resources. The host selects routing information (including subnet) based on a host-based SIP registration database or something similar.
2. The host includes the IP Address Criteria TLV (0x27B4) as the Criteria Type and the IP address that matches the appropriate routing information. The host selects the VoIP port from resource groups that are configured to be on the matching subnet.
3. The Router component sends an outbound seizure to the outbound span/channel Layer 4 PPL components.
4. The Layer 4 PPL components send the outbound call signal to the Layer 3 SIP User Agent Client.

5. The SIP User Agent Client initiates the outbound call using a SIP INVITE message. For H.323 calls, Layer 3 sends a SETUP message.

Call Flow - Outbound SIP Call

The following call flow diagram shows an outbound SIP call. H.323 calls are essentially the same.


Host Application
Layer 4
SIP UAC (Layer 3)
VDAC
SIP UAS
Route Control
Route Lookup
L3_L4 Outseize Control
Invite
180 Ringing
200 OK
ACK
Outseize Control
Outseize ACK
Alerting
Connect
Connect
Call Processing Event

Sample *Route Control* Message

The following is a sample *Route Control* message for the outbound side of a call using IP Based Routing.

```
00 Address Method
01 Number of Address Elements
29 02 ff fe Router AIB - Route Handle FF FE
02 ICB Count
03 ICB Type, Extended
00 1e Generic PPL TLV ICB
00 13  ICB data length
00 03  Number of TLVs
00 13 00 02 00 08 Routing Method = Criteria Type 0x0008
00 08 00 02 27 B4 Criteria Type TLV = Use IP Address
00 0F 00 01 0B Router protocol ID = 0x0b
03 ICB type 03=extended
00 33 NPDI Universal ICB
00 1a ICB Data Length
00 03  Number of TLVs
27 7E 00 03 08 00 00 Remote Side Protocol 08=SIP
27 B4 00 04 IP Address Criteria
27 17 00 05 10 00 04 22 11 Called Party Number
```

Call Agent Calls

Steps - Inbound

1. The INVITE message comes into the Layer 3 Call Agent Server on the CSP.
2. The SIP stack sends the internal *Route Control* message to the Layer 4 Router component. Because this call is a call agent call, the CSP does not yet know the IP address.
3. The SIP Call Agent Server sends the inbound call signal to the Inbound Span/Channel Layer 4 PPL components.
4. The Layer 4 PPL components send an inbound call trigger to the host application in the *Request for Service with Data* message.
5. The host application sends the *Connect with Data* (0x0005) message which allows the router to couple a virtual channel to a physical IP media channel. The *Connect with Data* message includes the Criteria Type TLV with the IP Address Criteria 0x27B4 as the routing criteria.

Steps - Outbound

1. The Router component sends an outbound seizure to the outbound span/channel Layer 4 PPL components.
2. The Layer 4 PPL components send the outbound call signal to the Layer 3 SIP Call Agent Client.
3. The SIP Call Agent Client initiates the outbound call using a SIP INVITE message.

9 H.323 Software

Purpose This chapter provides information about Dialogic's implementation of the H.323 protocol.

H.323 Protocol Overview

Protocol Standard

The H.323 protocol is an International Telecommunication Union (ITU-T) standard that describes transmission for packet-based video, audio, and data conferencing. The current ITU-T recommendation, known as Version 2.0, was ratified by Study Group 16 of the Telecommunications Sector. H.323 is an umbrella standard that refers to these other standards:

- The H.225.0 standard includes the following:
  - Q.931 for call signaling (setup, teardown)
  - RAS for Registration, Admission, and Status
  - RTP/RTCP for media stream packetization and synchronization
- H.235 for security
- H.245 for call control (capabilities, master/slave, logical channel)
- H.261 and H.263 video codecs
- H.450 for supplementary services
- G.711, G.728, G.729, G.723 audio codecs

Allows Multimedia

The H.323 protocol is used for multimedia (audio, video, and data) conferencing over packet networks. The goal of H.323 is to allow multimedia communication devices to interoperate regardless of the type of network those devices are connected to.

H.323 in Packet Networks

H.323 can be used on top of any packet network transport layer, such as Ethernet, IP/TCP/UDP, ATM, and Frame Relay. H.323 uses the Internet Protocol (IP) for internetwork conferencing.

CSP H.323 Offering

Overview

This section summarizes the Hardware Requirements and Software Overview for the H.323 offering.

Hardware Requirements

The following hardware is required to implement the H.323 Gateway feature and supporting software applications in the CSP.

- CSP Matrix Series 3 Card
- IP Signaling Series 3 card
- Chassis - Excel 2040, 2090 or 2110
- IP Network Interface Series 2 card or VDAC-ONE card for RTP streaming
- DSP-2 card or DSP-ONE card for DTMF tone detection and generation on the PSTN.

Other Prerequisites

Before you can configure your CSP to accommodate H.323 VoIP gateway functionality, you must first have:

- BOOTP Server, which is required for configuring the IP Address, Gateway IP Address, and Subnet Mask on the IP Signaling Series 3 card. See *Installing and Configuring a BOOTP Server* in the *Developer Guide: Overview.*
- An established, working IP network
- VDAC-ONE or IP Network Interface Series 2 card configured

Software Overview

VoIP Gateway

The H.323 VoIP Gateway complies with H.323 Version 2.0, and therefore supports all gateway Registration, Admission, and Status (RAS) functions.

The H.323 VoIP Gateway provides an interface between the PSTN and IP network for voice calls. IP-to-IP calls are also supported on the gateway. The H.323 stack is implemented on the specially-configured IP Signaling Series 3 card. The IP Signaling Series 3 card has a special software load that includes IP Signaling interface support and the H.323 stack.

Carrier-Class Voice Quality

Assuming adequate network conditions, the CSP delivers carrier-class voice quality over H.323.

End-To-End Solution

The CSP H.323 offering on the CSP provides both signaling and media capability, so application developers do not need to add these features separately. The H.323 signaling is provided by the IP Signaling Series 3 card and media is provided by the VDAC-ONE or IP Network Interface Series 2 card.

Seamless PSTN-IP Signaling Interworking: SIP/H.323/SS7/ISDN

This H.323 implementation uses Universal Protocol Data Format (UPDF) to provide seamless protocol interworking, with no need for protocol-specific messages. This interworking includes calls from PSTN-to-PSTN, IP-to-IP, PSTN-to-IP, and IP-to-PSTN.

User Input Indication Message

The User Input Indication (UII) message is available in the CSP to transmit inputs from the user interface to the receiver. The inputs could be a button that the user presses on the PSTN side of the call such as a DTMF keypad or hookswitch information. The UII message transmits the hookflash and DTMF signal information on the IP side of the call.

Hookflash Relay

The hookflash is a common input indicating a brief on-hook condition that occurs during a call. For example, during a call, the phone user quickly depresses and then releases the hook on their telephone. It is not long enough in duration to be interpreted as a signal to disconnect the call; however, telephone switches and PBXs are frequently programmed to intercept hookflash indications and use them to allow a

user to invoke supplemental services. For example, your local service provider may allow users to enter a hookflash to switch between calls if the user subscribes to a call waiting service.

To enable the CSP to transmit the PPL Event Indication message to the host, you must set Configuration Byte 1 for PPL Component L3P H.225 (0xA1) to 0x01. To disable, set the configuration byte to 0x00.

See the following PPL Component in this chapter.

PPL Component for H.225 - 0x00A1 (9-38)

The following data ICBs support the User Input Indication message. See the following in the ICB Chapter in the *API Reference*:

- *0x5B IP Signaling Series 3 Card ID*
- *0x5C H.323 Hookflash Received* (no data)
- *0x5D H.323 DTMF Signal Input Received*
- *0x5E H.323 Signal Update Input Received*
- *0x5F H.323 Alphanumeric Input Received*
- *0x62 Remote Endpoints UII Capabilities*

Alias Addressing

The CSP supports multiple alias addresses such as E.164, H.323Id, URL, and e-mail for the IP Signaling Series 3 card and remote H.323 endpoints.

The following TLVs support alias addressing:

- 0x02C1 Vendor ID
- 0x02C4 Gateway E164
- 0x02C6 Gateway URL ID
- 0x02C7 Gateway E-mail ID
- 0x27C4 Source IP
- 0x27C5 Source Port
- 0x27D8 Source H.323 ID
- 0x27D9 Source URL
- 0x27DA Source E-mail
- 0x27DC Remote H.323 ID
- 0x27DD Remote URL

- 0x27DE Remote E-mail

Gatekeeper Support

The CSP interfaces to gatekeepers from leading companies. This feature requires the CSP to send a gateway technology prefix in the registration request (RRQ). Use this prefix to configure the gateway's routing table or database in the gatekeeper. Include the TLV below in the *VoIP Protocol Configure* message to create and send the prefix.

0x02D3 Gateway Technology Prefix

Supported Specifications

In this CSP H.323 implementation, the following specifications are supported:

- H.225.0 (RAS, Q.931, RTP/RTCP)
- H.245 and audio codecs

RTP Streams

The VDAC-ONE or IP Network Interface Series 2 card provides the RTP stream for the H.323 call.

The H.323 software gets the RTP Payload Type and Payload size from the Route Table or *Route Control* message. If the type and size are not specified in either of those, the H.323 software uses the defaults.

Call Progress and Alerting Messages

The Progress message is an optional Q.931 call setup message. It can be sent by an H.323 gateway to indicate the progress of a call when interworking with a Switched Circuit Network (SCN). This message can also be sent by an H.323 endpoint before the Connect message is sent, depending on the supplementary service interaction.

On an H.323 call, Call Alerting and Progress messages can contain a Progress Indicator Information Element (IE) to describe an event that occurred during the life of the call. The CSP will establish a voice path when required.

The CSP does the following:

- Processes an optional Progress message on an outgoing call.
- Determines if the Q.931 Progress Indicator IE is present in the Alerting or Progress messages on a call.
- Sends a PPL event indication to the host when the CSP receives an Alerting or Progress message. A message is sent to the VDAC-ONE (or IP Network Interface Series 2) card to establish the voice path.

- Receives a PPL Event Request of Alerting or Progress from the host with the progress indicator.
- Transmits a Progress message to network.
- Transmits the appropriate Release Complete Reason in Release Complete message to the network. The Release Cause Codes are listed in the Release Cause Code TLV (0x27E3) in the *Tag Length Value Blocks* chapter in the *API Reference.*

This functionality works in gateway or normal (non-gateway) mode.

Dual Ethernet Port

You can configure the second Ethernet port on the IP Signaling Series 3 card to separate the H.323 signaling traffic from the host control traffic. Dialogic recommends having the host-to-CSP traffic carried on Ethernet port A and the H.323 signaling traffic carried on the Ethernet port B.

The BOOTP server is required to configure Ethernet port B.

The additional information is carried in the vendor specific area of the BOOTP response. You modify the BOOTP configuration to include new entries to carry the IP Address, Gateway IP Address, and Subnet Mask for the second physical Ethernet port.

Configuring the second Ethernet port is optional but if you enter an IP Address for Ethernet port B you must enter an associated Subnet Mask.

> **Important!** Although a gateway IP Address may be configured for both Port A and Port B, only one is utilized. If two gateway IP addresses are configured, one for Port A and one for Port B the one specified for Port A will be used. Therefore, Dialogic recommends that you configure only one gateway IP Address and that it be on Port B.

If you do not configure the second ethernet port, it is internally set to the same values configured for the first ethernet port.

For the associated procedure, refer to *Configuring Dual Ethernet Ports on IP Signaling Card* in the *Application Development* chapter in the *Developer Guide: Overview.*

Codec List in H.323-Initiated Offers

The CSP can send a list of up to five supported codecs in an H.323 offer to an endpoint when establishing a media session. The receiving endpoint selects one codec from that list and reports the selection back in the answer message to the CSP.

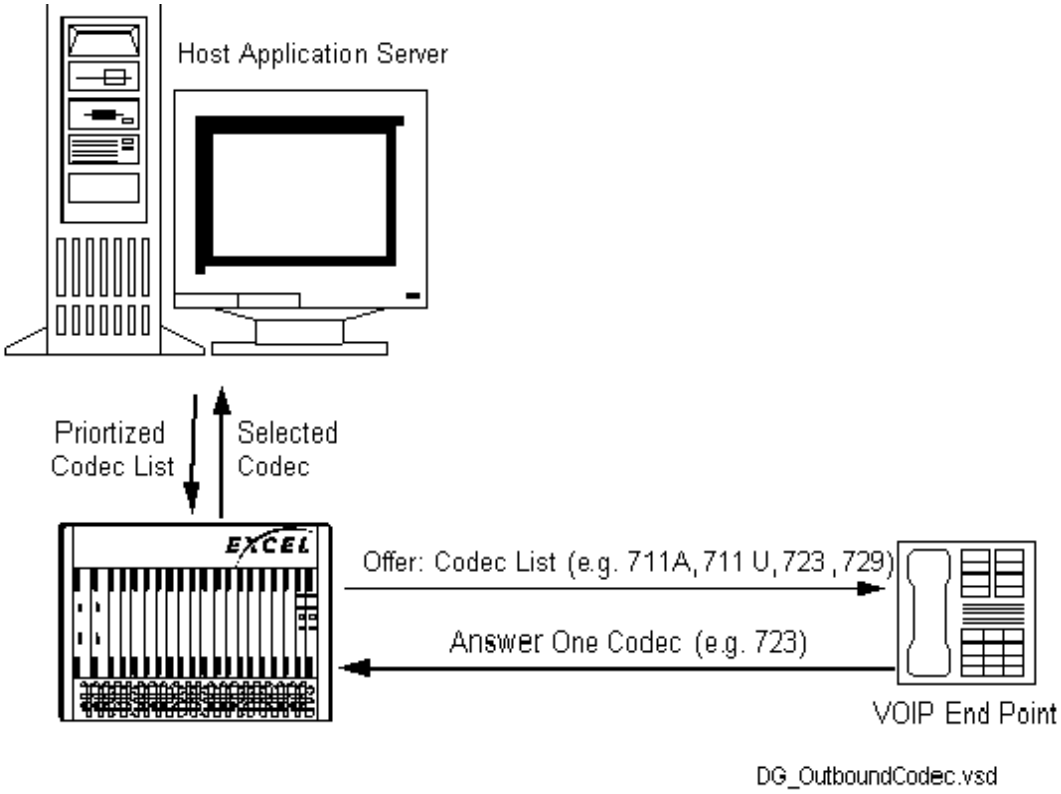
The following applies to this feature:

- The codec list is applicable for calls that use SIP or H.323 signaling and is not for clear-channel VoIP calls. For SIP signaling refer to *Host Control Method of SIP 180 Provisional Response Generation (5-166)*.
- The IP Network Interface card must use Profile 2.
- The VDAC-ONE card must use Profile 0.

Without this feature, the call might not get established because the CSP could include a codec type that endpoint will not accept.

The figure below provides an overview of this process.

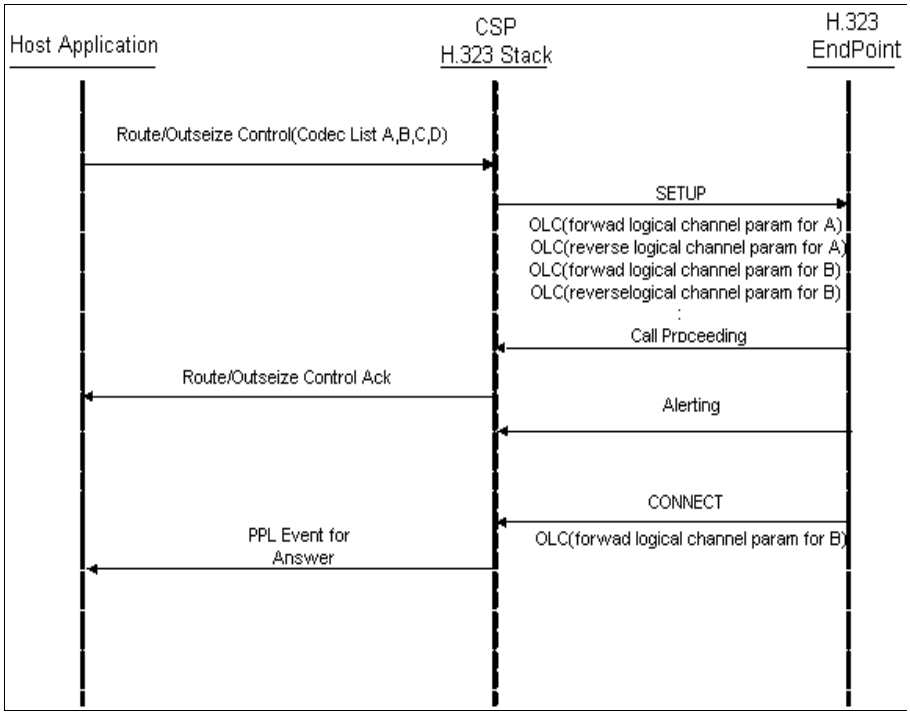
Figure 9-1 Codec List in Offer



Outseize Control messages

The host application provides the codec list and sends the list in the *Outseize Control* message as shown in Figure 9-2:

Figure 9-2 Host Provides Codec List



TLVs Used

Use the following TLVs in the *Outseize Control* message to include the codec types:

- 0x29FF - Local End-Point Media Info
- 0x2A0E - Media Connection Address
- 0x2A01 - Per Media Stream Information
- 0x2A03 - Media Type
- 0x2A07 - Media Port
- 0x2A02 - Per Codec Info

The *Outseize Control Message Trace* shows how to use these TLVs in the *Outseize Control* message. Note that the *Outseize Control* message cannot exceed 260 bytes.

Important! DO NOT use the following TLVs for these codec types:

- 0x0100 - RTP Payload Type

- 0x0101 - RTP Payload Size
- 0x27B0 - RTP Payload Type
- 0x27B1 - RTP Payload Size

Outseize Control Message Trace

The following trace for the *Outseize Control* message shows the nested TLVs containing the codec information.

```
00 78 00 2c 00 00 00
00 01 0d 03 00 64 03                'Span Channel AIB
01
03 00 33 00 66 00 06
27 7e 00 01 09                      'protocol type H323
27 c1 00 02 00 01                   'device server id
27 c2 00 04 0a 0a 1e 0a             'remote signalling ip
27 17 00 05 10 00 04 10 10          'called party #
27 18 00 07 10 00 04 0a 04 40 40    'calling party
29 ff 00 3d                         'Local End point Media INFO
   2a 0e 00 04 0a 0a 1e 34          'Media Per Stream Information
   2a 01 00 31                      'Media Connection Address
     2a 07 00 04 00 00 00 56        'Media Port
     2a 03 00 01 00                 'Media Type
     2a 02 00 0e                    'Media Per Codec Information
           2a 08 00 02 00 02        'Media Payload Type
           2a 0a 00 04 00 00 00 1e  'Media Payload Size
     2a 02 00 0e
           2a 08 00 02 00 01
           2a 0a 00 04 00 00 00 1e
```

H.323 Host Control Direct and Gatekeeper Routed Calls

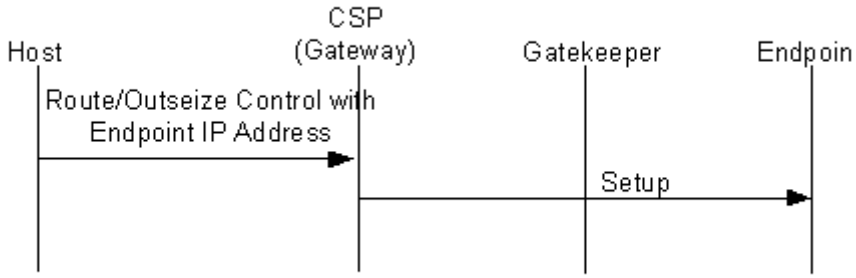
The H.323 protocol resident in the IP Signaling Series 3 card enables the CSP (gateway) to communicate with other H.323 gateways and gatekeepers in the call server network architecture.

Prior to this feature, the Gateway (CSP) supported either gateway (direct calls) or gatekeeper routed calls at any one time. When the Gateway (CSP) was registered with a gatekeeper it would process all the calls as Gatekeeper routed calls.

If the CSP is registered with a Gatekeeper, the CSP may send an ARQ (Admission Request) message to the Gatekeeper to request information about a remote endpoint. If the CSP chooses to send an ARQ to the gatekeeper, when the gatekeeper responds with the ACF (Access Confirmed), the response will determine if the CSP sends the call to the gatekeeper or directly to the remote endpoint.

Direct Routed

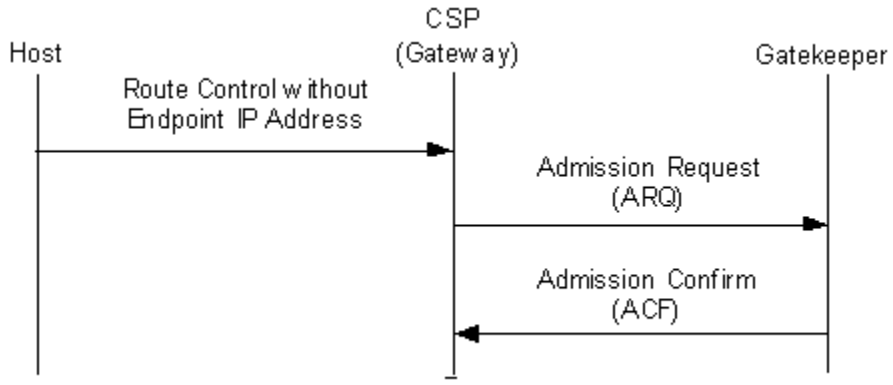
If the remote endpoint IP address TLV 0x27C2 is included in the *Route Control/Outseize* message, then the CSP will send the call to the remote endpoint IP address that is specified in the TLV 0x27C2. The CSP will not send the ARQ to the gatekeeper.



Gatekeeper Routed

If the remote endpoint IP Address TLV 0x27C2 is omitted from the *Route Control or Outseize Control* message, then the CSP sends the ARQ message to the Gatekeeper. When the ACF is returned, the CSP looks in the message to determine where to send the call. The CSP will send the call either to the remote endpoint IP address provided by the gatekeeper or to the gatekeeper as specified in the ACF message.

The end result is the CSP does not actually modify the gatekeepers' behavior. The CSP acts differently based upon data included in the *Route Control/ Outseize Control* message.



Invoking

To invoke this feature use the *Route Control* (0x00E8) message with the Remote End Signaling IP Address (Q.931) (0x 27C2) TLV. This TLV provides the remote endpoint IP address for the call. This TLV will be used for each call to determine whether the ARQ and ACF message exchanged is implemented for that call.

> **Important!** The Remote End Signaling Port (Q.931) (0x27C3) TLV specifies the remote signaling port. The defaults are: port number 1719 for gatekeeper routed calls and port number 1720 for gateway routed calls. This TLV is only necessary when using non-standard values.

H.245 Tunneling

The H.245 is a control signaling protocol in the H.323 multimedia communication architecture used to exchange end-to-end H.245 messages such as:

- master/slave determination
- capability exchange
- logical channel procedures

This configurable CSP H.245 Tunneling feature permits the encapsulation of H.245 messages within H.225 message between H.323 endpoints.

Enabling

To enable H.245 Tunneling for the IP Signaling card, use the H.245 Tunneling Configure TLV (0x02D5) in the *VoIP Protocol Configure* message (0x00EE). Refer to the TLV chapter in the *API Reference*.

Inbound and Outbound calls

The IP Signaling cards can then accept incoming calls with H.245 tunneling. With this feature enabled on the IP Signaling card, H.245 tunneling can be used on a per outbound call basis by using the H.245 Outbound Tunneling TLV (0x27E4) in *Route Control or Outseize Control* messages.

Note that if tunneling is not enabled on the IP Signaling card using the H.245 Tunneling Configure TLV (0x02D5) in the *VoIP Protocol Configure* message, then the CSP cannot offer outbound tunneling calls.

Configuring with CSA

Refer to the *Configuring H.323* in the *Converged Services Administrator User's Guide*.

H.323 Digit Collection

The H.323 Digit Collection feature enables you to collect Dual Tone Multi-Frequency (DTMF) digits using the H.323 protocol in the CSP. DTMF digits can be received and sent by the H323 protocol and can be sent in-band or out-of-band. The DTMF digits, routed to Layer 4 Dialing Plan (L4 DP) using the IP network, are either in signaling or in Real Time Protocol (RTP) format. These digits are decoded for the customer as part of their dialing plan.

When this feature is disabled, the DTMF digits are internally propagated and bypassed directly to the PSTN side using the *DSP Service Request* message. The customer is unable to initiate the collecting of digits.

Instead of bypassing the direct propagation of digits to the PSTN side, the configuration bytes listed below provide enhanced digit routing in order to support digit collection. These configuration bytes support both signal and alphanumeric type applications.

In-Band and Out-of-Band Methods for Routing Digits

The following in-band or out-of-band methods are used for routing digits using the H.323 protocol.

In-Band

- In-Band DTMF:
  The DTMF digit collection is provided by the IP Network Interface Series 2 and DSP Series 2 cards and then sent to L4.
- In-Band RFC 2833:
  RFC 2833 digit collection is provided by the IP Network Interface Series 2 and DSP Series 2 cards and then sent to L4.

Out-of-Band

- User Input Indication (UII):
  UII digits are sent out-of-band using the H.323 protocol. Digits can be sent as a H.245 signal, signal update or alphanumeric.
- Keypad Information Elements (IE):
  Allows the H.323 stack to pass the digits collected through the Keypad IE to L4.

PPL Components and Configuration Bytes

The PPL components listed below provide the configuration bytes required for H.323 Digit Collection.

PPL Component for H.225 (0x00A1)

- Enable Digit Routing to L4 CH Received in Keypad IE (0x04)

  This configuration byte allows the H.323 stack to pass the digits collected through Keypad IE to L4.

PPL Component for L3P H.245 (0x00A2)

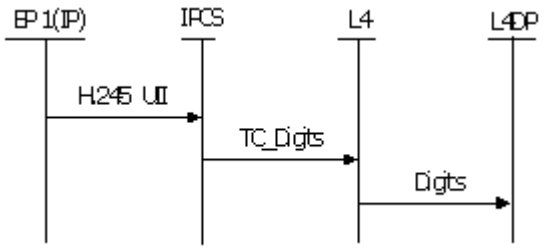
- Enable Digit Routing to L4 Channel Component (CH) Received in UII (0x06)

  This configuration byte allows the H.323 stack to pass the digits collected through UII to L4.

UII Digit Routing

The H.323 Stack passes UII digits to Layer 4 (L4)

The Digit Routing to L4 CH Received in UII (0x06) configuration byte sends a Tone Control (TC_Digit) request to L4, so that the digits will be collected in the L4 Dialing Plan (DP).



The DTMF signal is received in the H.323 in two parts.

- The Signal message will be received as digits either as the default duration or actual duration.
- The optional Signal Update message follows the Signal message to specify the override default duration. The Signal Update message is currently handled in H.323 as a Signal Update event in the H.245 PPL.

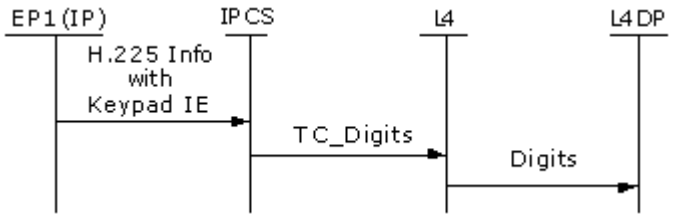
When this feature is enabled and the Signal Update message is received the Signal message is ignored.

Keypad IE Digit Routing

The CSP routes digits from Keypad IE to Layer 4 (L4)

Currently, the H.323 software does not sense the Keypad IE information when it arrives in the H.225 Information message. This feature provides that the H.225 Information message will be decoded in the H.323 stack. When the message includes a Keypad IE it will be extracted for the digits.

The Enable Digit Routing to L4 CH Received in Keypad IE (0x04) configuration byte allows the H.323 stack to pass the digits collected through the Keypad IE to L4. The Keypad IE information will be extracted and stored if it is available in the message. The digits are decoded in the IE that is being sent and propagated to L4 as if the DSP has received a digit. This in turn will be routed to L4DP.



The H.225 Information message is managed by the H.225 PPL Atomic Function AF 94 which provides for digit processing and routing the digits to L4.

RFC 2833 Digit Routing

The CSP routes digits coming from RFC 2833 to L4

The RFC 2833 is an IETF standard used to transmit digits over VoIP networks such as H.323. This is an In-band method of sending digits within the Real Time Protocol (RTP).

To use RFC 2833, the The IP Network Interface Series 2 card, VoIP modules need to be configured. This VoIP profile can be enabled for the RFC 2833 in the host. The IPCS card queries each VoIP module once, when the first call is sent. This information is stored in H.323 software database for further reference. The H.323 stack will be allowed to route digits from the RFC 2833 to L4.

The Resource Attribute Query (0x00E4) message queries the database using the following TLVs:

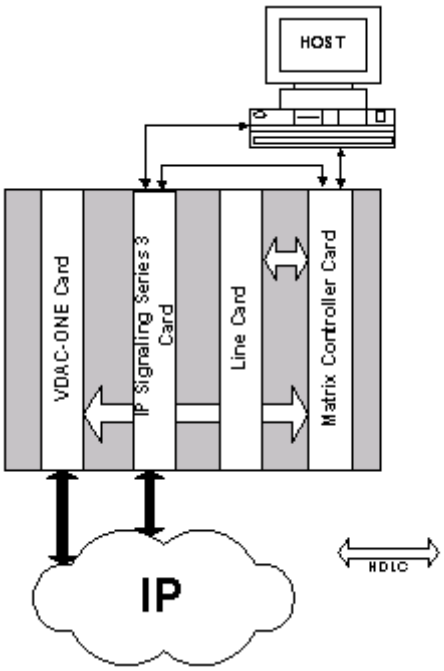
- RFC 2833 Enable (0x01E2) TLV indicates whether the RFC 2833 is enabled or not.
- RFC 2833 Dynamic Payload Type (0x01E2) TLV indicates the RFC 2833 payload size.

When enabled RFC 2833 outsiezes the VoIP information along with other parameters.

Internal Architecture

The line cards T-ONE, E-ONE, J-ONE, DS3, and VDAC-ONE (or IP Network Interface Series 2) card communicate with the CSP Matrix Series 3 Card over the internal HDLC bus. In addition, the IP Signaling Series 3 card and the VDAC-ONE (or IP Network Interface Series 2) card connect directly to the IP Network. The Host, IP Signaling Series 3 Card, and CSP Matrix Series 3 Card communicate over the Ethernet LAN.

Figure 9-3 The CSP chassis



IP Call Processing and Translation

The CSP processes H.323 calls as follows:

1. Any H.323-related message from the IP network is received on the IP Signaling Series 3 card and processed. If necessary, the message is translated and sent to the CSP Matrix Series 3 Card.
2. Any host message destined for the IP network is sent first to the CSP Matrix Series 3 Card, where it is processed and translated, then sent to the IP Signaling Series 3 card.
3. The IP Signaling Series 3 card then forwards the message to the IP Network.

4. The CSP Matrix Series 3 Card also handles all communication between the IP Signaling Series 3 card and the VDAC-ONE (or IP Network Interface Series 2) card.
5. The VDAC-ONE (or IP Network Interface Series 2) card is responsible for transmitting and receiving the IP RTP data stream.

The CSP as H.323 Protocol Converter

The CSP acts as an H.323 gateway by connecting the PSTN to an IP network. In the CSP, any inbound PSTN call (for example, T1) is connected to the outbound IP network by connecting the VoIP channels and T1 channels. For example:

1. A caller with a traditional analog telephone dials the number of an IP telephone.
2. The signal comes into the CSP.
3. The CSP Matrix Series 3 Card sends a *Request for Service* message to the host computer.
4. The host sends a *Route Control* message through the CSP Matrix Series 3 Card to the IP Signaling Series 3 card.
5. The IP Signaling Series 3 card sets up the call to the IP phone.
6. The host connects the channels.

Figure 9-4 PSTN-to-IP call

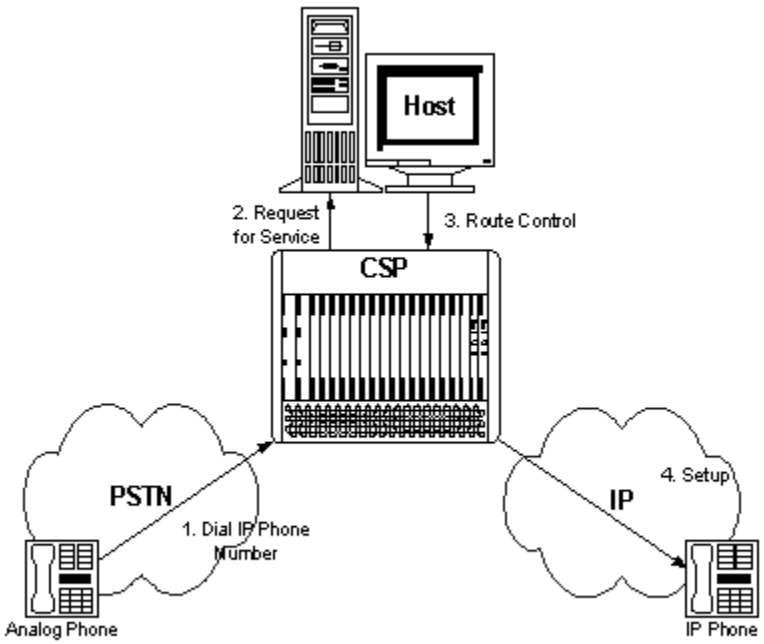
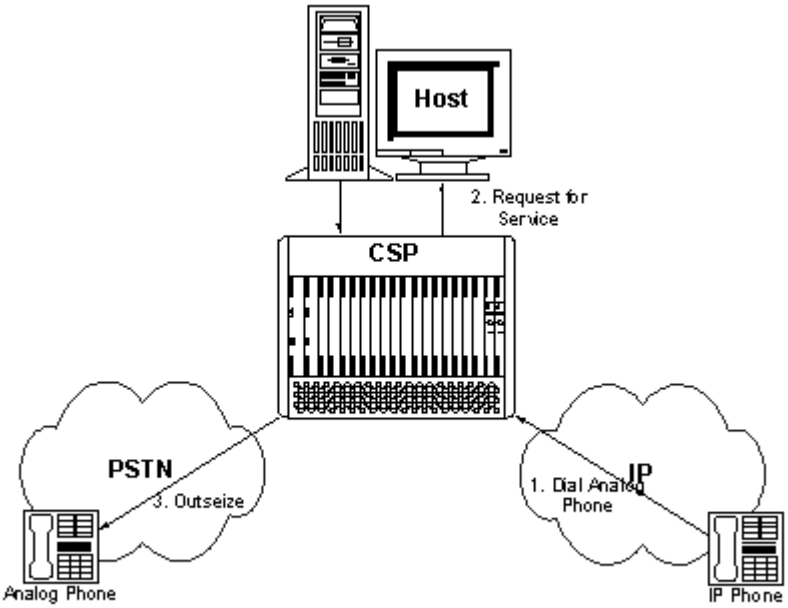


Figure 9-5 IP-to-PSTN call



H.245 Tunneling

The H.245 messages are encapsulated and carried between H.225 controlled endpoints within H.225 messages. This way of “piggy-backing” an H.245 message to an H.225 message is the H.245 Tunneling feature.

H.245 Tunneling is optional and negotiable between communicating H.323 endpoints. If both endpoints support this option, usually the H.245 Media Controlled messages are exchanged with the tunneling mechanism.

The CSP can accept H.245 Tunneling for all inbound calls even if outbound calls do not offer this capability. However, H.245 Tunneling for outbound H.323 calls is available only when it is enabled for inbound H.323 calls.

Configuring H.245 Tunneling for Inbound H.323 Calls

Send the following TLV to the CSP in the *VoIP Protocol Configure* message (0x00EE).

H.225 Tunneling Configure TLV (0x02D5)

Configuring H.245 Tunneling for Outbound H.323 Calls

Send the following TLV to the CSP with either the *Route Control* (0x00E8) message or *Outseize Control* (0x002C) message.

H.245 Outbound Tunneling TLV (0x27E4)

Configuring with CSA

Refer to *Configuring H.323* in the *SwitchKit Converged Services Administrator User's Guide*.

Possible Modifications to Existing Applications

EXS API Messages

You might need to add or modify the following API messages to your application to implement H.323 functionality:

- *Matrix Configure* (0x007D)
  Informs a CSP Matrix Series 3 Card about an IP Signaling Series 3 card. It sets up all of the information on the CSP Matrix Series 3 Card side to allow for the connection to the IP Signaling Series 3 card.
- *Matrix Query* (0x0097)
  Sent by the Host to query details of set configuration data and the status of the IP Signaling Series 3 interface.
- *IP Signaling Series 3 Card Configure* (0x0100)
  Inform the IP Signaling Series 3 card about a CSP Matrix Series 3 Card. It sets up all of the information on the IP Signaling Series 3 side that allows for the connection to the CSP Matrix Series 3 Card.
- *IP Signaling Series 3 Card Query* (0x0101)
  Used by the host to query the details of the configuration data and the status of the IP Signaling Series 3 interface.
- *IP Signaling Series 3 Card Status Report* (0x0105)
  This message is sent automatically (without being solicited) from the IP Signaling Series 3 card to the host in different scenarios. This scenario is generally caused by some Matrix interaction of which the host requires notification.
- *IP Signaling Series 3 Card Host Poll* (0x0104)
  This message is sent automatically (without being solicited) from the IP Signaling Series 3 card to the host on a periodic basis to allow the host to monitor the status of the IP Signaling Series 3 card. The default poll interval is one second.
- *VoIP Protocol Configure* (0x00EE)
  This message is used to configure H.323 and Session Initiation Protocol (SIP) software.
- *VoIP Protocol Query* (0x00EF)
  Use this message to query the VoIP configuration.
- *Request For Service With Data* (0x002D)
  This message is used to report incoming calls from H.323 endpoints.

- *Channel Released With Data* (0x0069)
  Used to include Network Protocol Data Intelligence (NPDI) Universal TLV for H.323 Release Code 0x27E3.
- *Release Channel With Data* (0x0036)
  Used to include NPDI TLV for H.323 Release Code to reject an incoming H.323 call.
- *PPL Event Request* (0x0044)
  For RAS PPL Component (0xa0) and H.245 PPL Component (0x0a2).
- *PPL Event Indication* (0x0043)
  For RAS PPL Component (0xa0) and H.245 PPL Component (0x0a2).

Important! Please note that the *Card Status Report* message contains no card-specific or other hardware-specific information for the IP Signaling Series 3 card other than card type and status.

- *Route Control* (0x00E8)
  Used to initiate an outseize on a channel. Replaces the *Outseize Control* message in existing applications.

  Within the *Route Control* message you must specify the remote endpoint alias address by including the following TLV(s) in the NPDI Universal ICB:

  - 0x27C2 Remote End Signaling IP Address
    0x27C3 Remote End Signaling Port (Q.931) (optional)
    or
  - 0x27DC Remote H.323 ID
    or
  - 0x27DD Remote URL
    or
  - 0x27DE Remote e-mail
    or
  - any combination of the alias addresses

If the CSP is registered with a gatekeeper, you do not need to include the IP address and port in the *Route Control* message.

Configuration

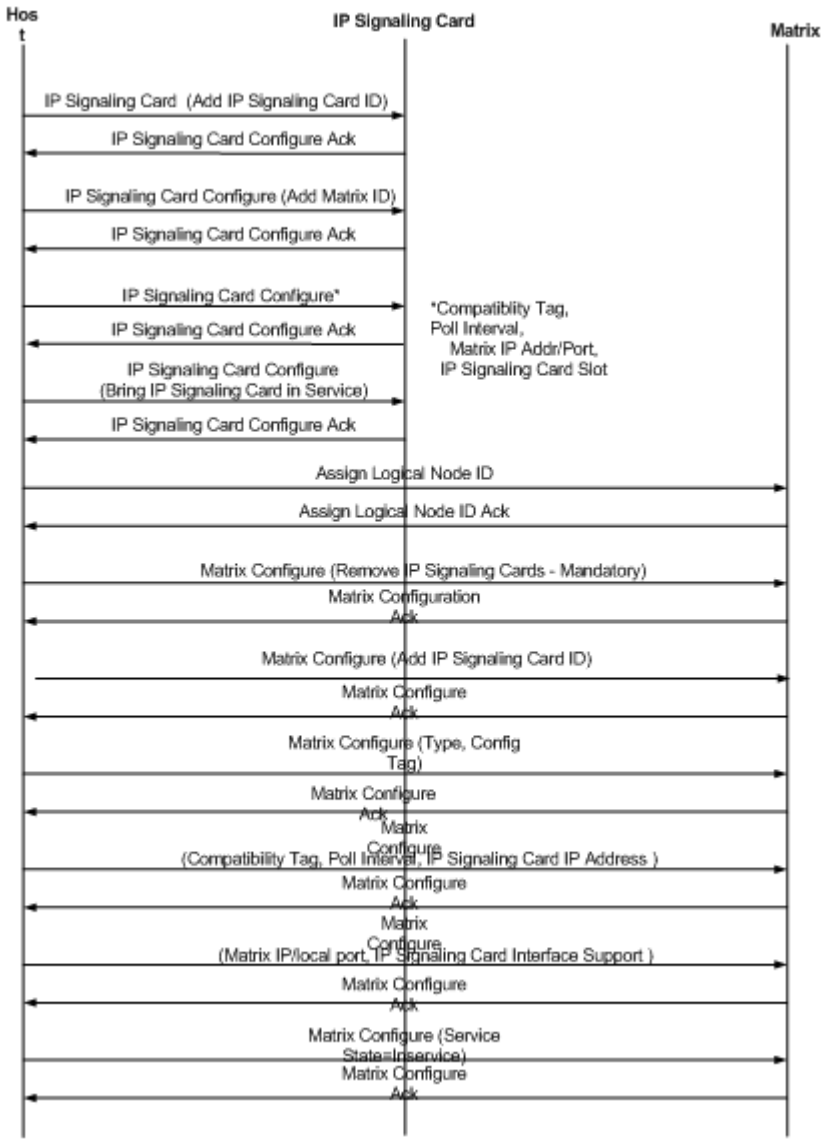
The CSP Matrix Series 3 Card must receive configuration information to recognize an IP Signaling Series 3 card. The CSP Matrix Series 3 Card recognizes the IP Signaling Series 3 card on initialization, power-up, reset, or insertion.

Important! You can receive a “board dead” message for the IP Signaling Series 3 card if it is not configured and brought into service before the relevant line card timer expires. The IP Signaling Series 3 card comes into service only when configured for the first time by the host and, thereafter, when you perform a push-button reset.

1. Use the BOOTP server to configure the IP address, Gateway IP address, and Subnet mask of the IP Signaling Series 3 card.
2. Download the system software to the IP Signaling Series 3 card and CSP Matrix Series 3 Card. TFTP is used for the download. Separate tftp.cfg files are required for the CSP Matrix Series 3 Card and the IP Signaling Series 3 card; however, the same load tag (EXCPU_LOAD) is used for both the cards.
3. The host must configure the IP Signaling Series 3 cards’ interface for communicating over Ethernet with the CSP Matrix Series 3 Card. The host performs this configuration by using the *IP Signaling Series 3 Card Configure (0x0100)* message and its associated TLVs.
4. Assign Logical Node IDs.
5. Configure the CSP Matrix Series 3 Card using the *Matrix Configure (0x007D)* message. The CSP 2000 Matrix can then recognize the IP Signaling Series 3 card. The CSP Matrix Series 3 Card then sends a *Card Status Report* for the IP Signaling Series 3 card to the host. The first TLV sent should be the *0x0002 Remove IP Signaling Series 3 Card*.
6. When the IP Signaling Series 3 card is in service, all communication occurs over the TCP/IP connection to the CSP Matrix Series 3 Card.
7. Assign VDAC-ONE (or IP Network Interface Series 2) card IP Addresses.
8. Assign Logical Span IDs for VDAC-ONE (or IP Network Interface Series 2) card.

9. Configure Answer Supervision for Notify Host of Answer using the *PPL Configure* message.
10. Bring the spans in to service.
11. Bring the channels in to service.
12. Configure Optional VDAC-ONE (or IP Network Interface Series 2) card attributes.
13. Configure Routing.
14. Configure H.323 attributes.

The configuration message flow from the host to the CSP is illustrated below:


Host
IP Signaling Card
Matrix
IP Signaling Card (Add IP Signaling Card ID)
IP Signaling Card Configure Ack
IP Signaling Card Configure (Add Matrix ID)
IP Signaling Card Configure Ack
IP Signaling Card Configure*
IP Signaling Card Configure Ack
*Compatiblity Tag,
Poll Interval,
Matrix IP Addr/Port,
IP Signaling Card Slot
IP Signaling Card Configure
(Bring IP Signaling Card in Service)
IP Signaling Card Configure Ack
Assign Logical Node ID
Assign Logical Node ID Ack
Matrix Configure (Remove IP Signaling Cards - Mandatory)
Matrix Configuration Ack
Matrix Configure (Add IP Signaling Card ID)
Matrix Configure Ack
Matrix Configure (Type, Config Tag)
Matrix Configure Ack
Matrix Configure
(Compatibility Tag, Poll Interval, IP Signaling Card IP Address)
Matrix Configure Ack
Matrix Configure
(Matrix IP/local port, IP Signaling Card Interface Support)
Matrix Configure Ack
Matrix Configure (Service State=Inservice)
Matrix Configure Ack

Setting up H.323 without Using SwitchKit

Configuring the IP Signaling Series 3 card

```
'Add IP Signaling Series 3 Card ID
00 18 01 00 00 00 fe 00 02 52 02 ff ff 53 04 00 20 ff ff
   01 00 01 00 02 00 3c


'Add matrix ID
00 18 01 00 00 00 fe 00 02 52 02 00 3c 53 04 00 20 ff ff
   01 00 03 00 02 00 01

'IP Signaling Series 3 Card CFG
00 3a 01 00 00 00 fe 00 02 52 02 00 3c 53 04 00 20 00 01
   05 \
00 08 00 02 01 01 \ 'Compatibility tag
00 0a 00 04 00 64 00 0A \ 'Poll Interval
00 0c 00 06 87 77 37 b4 36 ff \ 'Matrix IP, port
00 05 00 02 00 0c 'Slot Number

'Service State Configure
00 18 01 00 00 00 fe 00 02 52 02 00 3c 53 04 00 20 00 01
   01 00 0e 00 02 00 01
```

Configuring the Matrix card

```
'Assign Logical Node Id
00 0e 00 10 00 00 ff 00 01 10 05 00 00 20 9D 01

'Remove IP Signaling Series 3 Card (Mandatory)
00 0f 00 7d 00 00 01 00 00 20 01 00 02 00 02 ff ff

'ADD IP Signaling Series 3 Card
00 0F 00 7d 00 00 01 00 00 20 01 00 01 00 02 00 08

'Configure IP Signaling Series 3 Card
00 1b 00 7d 00 00 01 00 01 73 02 00 08 20 02 \
00 03 00 04 00 01 00 00 \ 'H.323 IP Signaling Series 3
   Card
00 04 00 02 0f 32 'Configuration Tag
00 25 00 7d 00 00 01 00 01 73 02 00 08 20 03 \
00 08 00 02 01 01 \ 'Compatibility Tag
00 09 00 04 00 64 00 0A \'Poll Interval
(time between polls in 1/10 of a second. No of max. missed
   polls)
00 05 00 06 0A 0A BE 3c 00 00'IP Signaling Series 3 Card
   Address
00 31 00 7d 00 00 01 00 01 73 02 00 08 20 02 \
00 0a 00 06 00 00 00 00 31 51 \'Matrix Local Port
```

```
00 0f 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
   00
'IP Signaling Series 3 Card Interface
'Support Disabled
'Bring Device Server Into Service
00 13 00 7d 00 00 01 00 01 73 02 00 08 20 01 00 07 00 02
   00 01
```

Discovering a Gatekeeper

The CSP can auto-discover or manually discover a gatekeeper. In either case, the H.323 pre-configured Gatekeeper Discovery is the only method supported. The CSP must know the IP address of the gatekeeper to discover and it sends the gatekeeper discovery message (GRQ) to that IP address only. The CSP does not support the Broadcast Discovery method.

Before discovery or registration, the host must have already configured the IP Signaling Series 3 card with a Gatekeeper using the *VoIP Protocol Configure (0x00EE)* message.

> **Important!** Auto Discovery and Auto Registration should always be set the same: either both enabled or both disabled

Auto-Discovery

If Auto-Discovery is enabled, the IP Signaling Series 3 card discovers (Unicast) the Gatekeeper on its own, with no prompting from the host.

The GRQ timer is used to configure the CSP to send the Gatekeeper Request (GRQ) message. When the IP Signaling Series 3 card becomes active, the GRQ timer is started. When the GRQ timer expires, the CSP sends a GRQ to the gatekeeper.

Manual Discovery

If Auto-Discovery is disabled, the host must send a PPL Event Request of GRQ. Only then can the IP Signaling Series 3 card send a Gatekeeper Request (GRQ) to the Gatekeeper.

Registering with a Gatekeeper

Automatic Registration

If Auto-Discovery is enabled you must also enable Auto-Registration. The IP Signaling Series 3 card automatically discovers the Gatekeeper and then automatically registers with the Gatekeeper.

The Registration Request (RRQ) timer is used to configure the CSP to send the RRQ message after the CSP receives the Gatekeeper Confirm (GCF) message.

When the RRQ, DRQ, and URQ messages fail because the CSP is not registered, the host is informed of this failure (with the *PPL Event Indication* message) and the CSP does not retry the messages because it will likely keep failing. The CSP stays in the discovered state. The host may send these messages at any time or it can also deconfigure the system which will cause it to go to the undiscovered state.

Manual Registration

If Auto-Discovery is disabled, you must also disable Auto-Registration.

In this scenario, the host must send a PPL Event Request of GRQ to discover the Gatekeeper. The host must send a PPL Event Request of Registration Request (RRQ) to register with the Gatekeeper.

Registration / Un-registration

The IP Signaling Series 3 card, when configured with a gatekeeper's IP address, can register with that gatekeeper with an H.323 ID `EXCEL-CSP`. When two or more IP Signaling Series 3 cards are to register with the same gatekeeper, the host should change the H.323 ID of the IP Signaling Series 3 cards involved, or else the gatekeeper will reject the registration request.

Before the IP Signaling Series 3 card is registered with any gatekeeper, the host can configure the Gateway's alias names using the *VoIP Protocol Configure* (0x00EE) message. The following aliases are configurable: E.164, H.323Id, URL and e-mail.

If the host wants to remove the aliases configured before the IP Signaling Series 3 card is registered with the gatekeeper, it can do so by setting all the aliases except the H.323 IDs to `NULL` or it can change their values. When the IP Signaling Series 3 card is registered with the gatekeeper, the host cannot change any of the aliases. To do so, the host has to un-register from the gatekeeper first, then configure the aliases and then register with the gatekeeper again.

PPL Component for RAS - 0x00A0

Overview This section includes configuration bytes, PPL Events, and Event Indications for the PPL Component RAS.

Table 9-1 Configuration Bytes

Byte (Hex)	Function	Option (*default)
0x01	Toggle PPL indication Enable/Disable	0x00* - enable
0x02	Enable/Disable continuous discovery and registration attempts with the gatekeeper	0x00* - disable 0x01 - enable

Important! By default, the configuration byte is set to zero and PPL event indications are sent to the host. If it is set to anything other than zero, PPL event indications are not sent to the host.

AIB (00 01 0d 03 ff ff ff) is valid for the following PPL Event Requests and Indications which are not related to any active calls: RRQ, URQ, RCF, UCF, RRQ FAIL, URQ FAIL, RRJ, URJ, BRQ, GRQ. Other PPL Event Requests and Indications have a valid Span Channel.

The configuration byte for RAS uses the AIB with FFFFFF as the span channel.

Table 9-2 PPL Event Requests

PPL Event Request ID	Purpose
0x01	Send Gatekeeper Request (GRQ)
0x02	Send Registration Request (RRQ)
0x03	Send Unregistration Request (URQ)
0x04	Send Bandwidth Request (BRQ)

Table 9-3 PPL Event Indications

PPL Event Indication ID	Purpose
0x01	Admission Confirm (ACF) Indication
0x02	Admission Reject (ARJ) Indication
0x03	Admission Request (ARQ) Indication
0x04	Bandwidth Confirm (BCF) Indication
0x05	Bandwidth Reject (BRJ) Indication
0x06	Bandwidth Request (BRQ) Indication
0x07	Disengage Confirm (DCF) Indication
0x08	Disengage Reject (DRJ) Indication
0x09	Disengage Request (DRQ) Indication
0x0A	Gatekeeper Confirm (GCF) Indication
0x0B	Gatekeeper Reject (GRJ) Indication
0x0D	Info Request Ack (IACK) Indication
0x0E	Info Request NACK (INAK) Indication
0x0F	Info Request (IRQ) Indication
0x10	Info Request Response (IRR) Indication
0x14	Resource Available Confirm (RAC) Indication
0x15	Resource Available Indicate (RAI) Indication
0x16	Registration Confirm (RCF) Indication
0x17	Request In Progress (RIP) Indication
0x18	Registration Reject (RRJ) Indication
0x1A	Unregistration Confirm (UCF) Indication

PPL Event Indication ID	Purpose
0x1B	Unregistration Reject (URJ) Indication
0x1C	Unregistration Request (URQ) Indication
0x1D	Gatekeeper Request (GRQ) FAIL Indication
0x1E	Registration Request (RRQ) FAIL Indication
0x1F	Admissions Request (ARQ) FAIL Indication
0x20	Disengage Reject (DRQ) FAIL Indication
0x21	Bandwidth Request (BRQ) FAIL Indication
0x23	Unregister Request (URQ) FAIL Indications
0x24	XRS Indication. Message Not Understood.

See the *API Reference* for information on ICB 0x51 Soft Register All Switch-Initiated Messages.

Refer to *PPL Component Addressing* in the *PPL Component Information* chapter in the *API Reference.*

Table 9-4 PPL Timer

Timer ID (Hex)	Timer Name	Default Value
0x01	RAS PPL Timer 1	3 seconds

Getting Gatekeeper Discovery and Registration

You can query the Status Gatekeeper status using the *PPL Data Query* (0x00DE) message. The response returns information about the Gatekeeper discovery and registration status along with the Gatekeeper IP and Port.

Example:

```
                                    Sp     ch comp.Id entity
00 0F 00 DE 00 00 01 00 01 0D 03 FF FF FF  00 A0   08
```

This query is valid for only the RAS PPL Component (0x00A0). The AIB used is the expanded span channel with the values equal to 0xFFFF 0xFF. Any other PPL Component and span channel values will be rejected with an invalid address response (0xD2).

Example: The following is a sample response from the CSP to the host:

```
00 18 00 DE 00 01 01 00 10    PPL Data Query, Node 1, ACK
00 01 0D 03 FF FF FF          Span Channel 0xFFFF, 0xFF
00 A0                         RAS Component
08                            Entity = Gatekeeper Status
02                            Status = 2
0A 0A 0E 02                   Gatekeeper IP Address
06 B7                         Gatekeeper Port
```

The status values are as follows:

0 - undiscovered

1 - discovered, unregistered

2 - registered

0xFF - Invalid

The invalid status is returned if the query is made when the RAS PPL is in a non-stable state.

For status values of 1 and 2, the response is the same as the example above. For status values of 0 and 0xFF, the status response is the following:

```
00 12 00 de 00 01 01 00 10 00 01 0d 03 ff ff ff 00 a0 08
   00
00 12 00 de 00 01 01 00 10 00 01 0d 03 ff ff ff 00 a0 08
   ff
```

This response differs from the first response above in that the Gatekeeper IP Address and port are not included in the response.

Continuous Attempts at Gatekeeper Discovery and Registration

The CSP can continuously attempt H.323 gatekeeper discovery and registration at the following points in time:

- Initial gatekeeper discover
- Initial gatekeeper registration
- Subsequent registration after the initial registration is successful

Initial Gatekeeper Discovery

When the initial discovery attempt fails, the CSP checks Configuration Byte 2. If the value is non-zero, the value stored in the RAS PPL Timer 1 is loaded into a generic PPL timer. When this timer expires, the CSP attempts to re-discover the gatekeeper. (The default value for the timer is three seconds.)

If this attempt fails, the CSP repeats the process continuously until the CSP receives a successful response from the gatekeeper or the host disables the functionality by clearing Configuration Byte 2.

Each time a discover attempt fails, the CSP sends the host a PPL Event Indication of GRQ Fail (0x1D). Just the event is sent – no ICB. The CSP also sends an ACK to the original PPL Event Request of GRQ. Even though the host sends a single PPL Event Request message, it receives multiple ACKs.

Initial Gatekeeper Registration

When the initial registration attempt fails, the CSP checks Configuration Byte 2. If the value is non-zero, the value stored in the RAS PPL Timer 1 is loaded into a generic PPL timer. When this timer expires, the CSP attempts to re-register with the gatekeeper. (The default value for the timer is three seconds.)

If this attempt fails, the CSP repeats the process until the CSP receives a successful response from the gatekeeper or the host disables the feature by clearing Configuration Byte 2. If the host disables the feature, the PPL returns to the Stable Undiscovered State 1, which is the normal operation of the PPL.

Each time a registration attempt fails, the CSP sends the host a PPL Event Indication of RRQ Fail (0x1E). Just the event is sent – no ICB. The CSP also sends an ACK to the original PPL Event Request of RRQ. Even though the host sends a single PPL Event Request message, it receives multiple ACKs.

If Configuration Byte 2 is disabled (value 0) and the CSP does not get an RCF message in response to the RRQ message, the CSP goes into undiscovered mode.

Subsequent Gatekeeper Registration (Lightweight RRQ)

Once the initial registration is successful, the IP Signaling Series 3 card enters the registered state. In this state, the CSP periodically sends Lightweight RRQ messages to the gatekeeper to maintain registration. The interval of the Lightweight RRQ messages is determined by the gatekeeper in the RCF response.

The gatekeeper responds to the RRQ message with an RCF message. If the CSP receives no response, under normal circumstances (prior to this functionality) the IP Signaling Series 3 card unregisters itself and returns to the undiscovered state.

In this continuous registration mode, the CSP checks the value of Configuration Byte 2. If this value is non-zero, the value stored in the RAS PPL Timer 1 is loaded into a generic PPL timer. (The default value is three seconds.). When this timer expires, the CSP attempts to re-register with the gatekeeper. If this attempt fails, the CSP repeats this process continuously until the CSP receives a successful response from the gatekeeper or the host disables the feature by clearing Configuration Byte 2.

Each time a registration attempt fails, the CSP sends a PPL Event Indication of RRQ Fail (0x1E) to the host.

If Configuration Byte 2 is disabled (value 0) and the CSP does not get an RCF message in response to the RRQ message, the CSP goes into undiscovered mode.

Examples of Host Messages

The following are examples of the PPL Configure and PPL Timer Configure API message used for the new gatekeeper discovery/ registration mode.

Table 9-5 PPL Configure Message with RAS PPL Configuration Byte 2

Header	00 17 00 D7 00 00 FF
AIB (Range of Channels)	01 02 0D 03 FF FF FF 0D 03 FF FF FF
Component ID	00 A0
Entity	01
Number of Config. Bytes	01
Config. Byte Number	02
Data	00 - Disable 01 - Enable

Table 9-6 PPL Timer Configure Message with RAS PPL Timer 1

Header	00 17 00 CF 00 00 FF
AIB (Range of Channels)	01 02 0D 03 FF FF FF 0D 03 FF FF FF

Header	00 17 00 CF 00 00 FF
Component ID	00 A0
Timer Type	01
Timer ID	01
Value	0x0000 - 0xFFFF 01 2C - Default (300 in units of 10ms=3 seconds)

PPL Component for H.225 - 0x00A1

Overview This section includes configuration bytes, PPL Events, and Event Indications for the PPL Component H.225. This section also includes information on how this component affects the Call Progress and Alerting messages

Table 9-7 Configuration Bytes

Byte (Hex)	Function	Option (*default)
0x01	Propagate PPL Event Indications to Host	0x00* - Do not propagate 0x01 - Propagate
0x02	Cut voice path after sending/ receiving Alerting message.	0x00 - Disable 0x01 - Enable
0x03	Cut voice path after sending/ receiving Progress message.	0x00 - Disable 0x01 - Enable
0x04	Enable digit routing to received in Keypad IE in INFO message.	0x00* - Disable 0x01 - Enable Routing

PPL Event Request

Table 9-8 PPL Event Request

PPL Event Request ID	Purpose
0x09	H.225 Alerting
0x0A	H.225 Progress Request

Table 9-9 PPL Event Indications

PPL Event Indication ID	Purpose
0x01	Fax Start
0x02	Fax End
0x05	Modem Start (not supported)
0x06	Modem End (not supported)
0x07	RTP Timer Expired
0x08	Media Inactivity Timer Expired
0x09	H.225 Alerting Indication
0x0A	H.225 Progress Indication

Important! The PPL Indications will always have one ICB which give the IP Signaling Series 3 card ID ICB information.

Call Progress and Alerting Messages

Configuring the PPL Component 0xA1 to Transmit the PPL Event Indications

To enable the CSP to transmit the *PPL Event Indication* message to the host, you must set Configuration Byte 1 for PPL Component L3P H.225 (0xA1) to 0x01. To disable, set the configuration byte to 0x00.

Table 9-10 Sample PPL Configure message

Header	00 01 00 D7 00 00 01
AIB (Range of Channels)	01 02 0D 03 00 A8 00 0D 03 00 AC 1F
Component ID	00 A1
Entity	01
Number of Config. Bytes	01
Config. Byte Number	01
Data	00 - disable 01 - enable

The PPL Event Indications will include an ICB of Progress Indicator IE (Information Element) if this IE is included in the Alerting or Progress message received from the network.

Table 9-11 Example

Header	00 16 00 43 00 00 01
AIB	00 01 0D 03 00 A8 00
Component ID	00 A1
PPL Event	00 09 or 00 0A
Count	0x01 or 0x02
Type	Data 0x02
Subtype	IP Signaling Card ID 0x5B Example: 0x003C
Following data is optional	
Type	Data 0x02
Subtype	ISDN formatted IE 0x10
Length	0x06
Data	01 08 03 00 00 01

H.225 Progress Request (0x0A) signals to the IP Signaling Series 3 card when to transmit the Progress message to the network.

Table 9-12 Sample PPL Event Request

Header	00 11 00 44 00 00 01
AIB (Channel)	00 01 0D 03 00 A8 00
Component ID	00 A1
PPL Event Request	00 0A
Number of ICBs	00 or 01 If the application requires the voice path brought in, select 01 (see ICB below)

Data ICB Progress Indicator IE

The *PLL Event Request* message may also include a data ICB - Progress Indicator Information Element (IE). This IE requests the IP Signaling Series 3 card to transmit a Progress message to the network and include the Progress IE.

This IE may be ISDN Raw (0x11) or Formatted (0x10). See the following examples:

Table 9-13 ISDN Raw Data

Header	00 19 00 44 00 00 FF
AIB (Channel)	00 01 0D 03 00 00 00
Component ID	00 A1
PPL Event Request	00 0A
Number of ICBs	01
ICB Type	02
IE (ISDN Raw)	11
Length	05
Number of IEs	01
IE Type	1E
IE Length	02
IE Data	80 88 or 80 81

Table 9-14 Formatted Data

Header	00 19 00 44 00 00 FF
AIB (Channel)	00 01 0D 03 00 00 00
Component ID	00 A1
PPL Event Request	00 0A
Number of ICBs	01
ICB Type	02
IE (ISDN Raw)	10
Length	06
Number of IEs	01
IE Type	08
IE Length	03
IE Data	00 00 08 or 00 00 01

Configuring the PPL Component 0xA1 to Establish the Voice Path

To enable the CSP to establish the voice path before sending or receiving the Connect message, you must set either Configuration Byte 2 or 3 for the PPL Component H.255 (0x00A1) as described below.

Parking the channel allows the Alerting or Progress message to be sent to the host. Parking is optional when you set Configuration Byte 2 but mandatory when you set Configuration Byte 3.

Alerting Message

If you enable Configuration Byte 2 as indicated below, the voice path cut-through happens after the Alerting message is sent or received, whether or not the Progress Indicator IE is present.

Table 9-15 PPL Configure Message with Configuration Byte 2

Header	01 17 00 D7 00 00 01
AIB (Range of Channels)	01 02 0D 03 00 A8 00 0D 03 00 AC 1F
Component ID	00 A1
Entity	01
Number of Config. Bytes	01
Config. Byte Number	02
Data	00 - Disable 01 - Enable

Progress Message

If you enable the configuration byte below, the voice path cut-through happens after the Progress message is sent. This happens if the Progress message is sent or received when the Progress message contains the Progress Indicator IE and the Progress Description is set to 1 or 8.

Table 9-16 PPL Configure Message with Configuration Byte 3

Header	00 17 00 D7 00 00 01
AIB (Range of Channels)	01 02 0D 03 00 A8 00 0D 03 00 AC 1F
Component ID	00 A1
Entity	01
Number of Config. Bytes	01
Config. Byte Number	03
Data	00 - Disable 01 - Enable

Both Configuration Bytes Set

If you set both configuration bytes above, the voice path cut-through happens after the CSP sends or receives either the Alerting or Progress messages, whichever is processed first.

PPL Component for L3P H.245 - 0x00A2

Overview

This section includes configuration bytes, PPL Events, and Event Indications for the PPL Component L3P H.245.

Configuration Bytes

Table 9-17 Configuration Bytes

Byte (Hex)	Function	Option (*default)
0x01	T.38 fax parameters	0x00*- Send the T.38 fax parameters in the TCS message. 0x01 - Do Not Send T.38 parameters.
0x02	Round trip delay	0x00 - Do not initiate RTDSE. A non-zero character will initiate RTDSE.
0x03	UII Propagation/Reporting UII signals are propagated or reported to the host as a PPL Event Indication	0x00 - Do propagate, do not report 0x01 - Do propagate, do report 0x02 - Do not propagate, do not report 0x03 - Do not propagate, do report
0x04	DTMF Digit Propagation/ Reporting Digit information is propagated or reported to host in the form of a PPL Event Indication DTMF Digit Receive timeout is reported to host in the form of a PPL Event Indication when this configuration byte is set to 0x01 or 0x03.	0x00* - Do propagate, do not report 0x01 - Do propagate, do report 0x02 - Do not propagate, do not report 0x03 - Do not propagate, do report

Byte (Hex)	Function	Option (*default)
0x05	Attach DTMF Receiver	0x00* - Do not attach DTMF Receiver when call answered 0x01 - Attach DTMF Receiver when call answered
0x06	Enable digit routing to L4 received in UII.	0x00* - Disable 0x01 - Enable Routing

PPL Event Requests

Table 9-18 PPL Event Requests

PPL Event Request ID	Purpose
0x01	Attach DTMF Receiver
0x02	Release DTMF Receiver

You can use these PPL Event Requests to attach or release a DTMF receiver from the IP channel in an answered call. The default is that the DTMF receiver will not be attached to the IP channel when the call is answered.

If there is a problem attaching the DTMF receiver, the channels involved will not purge and the CSP sends an alarm to the host:

- 0x06 DSP Resource Block Occurred
- 0x07 DSP Resource Function Type Not Configured
- 0x08 DSP Resource Management Inconsistent

If the host generated the request via a PPL Event Request, an appropriate response is returned to the host.

In cases where the CSP Matrix Series 3 Card does not send an alarm, the IP Signaling Series 3 card generates its own channel alarm:

- 0x09 Invalid Associated Channel
- 0x0A DSP Resource Wait Timeout

If the host generated the request via a PPL Event Request, an appropriate response is returned to the host.

- 0x519C No Associated Timeslot
- 0x519D DSP Resource Wait Timeout
- 0x519E Invalid DSP Resource Request

PPL Event Indication ID	Purpose
0x01	User Input Indication
0x02	Hookflash Indication
0x03	DTMF Digit Indication
0x04	DTMF Digit Receive Timeout
0x05	Remote Endpoints User Input Indication Capabilities

Refer to *PPL Component Addressing* in the *PPL Component Information* in the *API Reference*.

DTMF Timers

The timers below determine the DTMF transmission/reception parameters on the PSTN channel associated with an IP channel in a call.

To configure the DTMF timers for the H.245 component, use the *PPL Timer Configure (0x00CF)* message with PPL Component of 0xA2 and Timer Type of 0x03. The span channel values used are those assigned to the VDAC-ONE (or IP Network Interface Series 2) card.

To query the DTMF timers, use the host message *PPL Data Query 0x00DE* with the PPL component of 0x00A2 (H.245) and the span channel assigned to the VDAC-ONE (or IP Network Interface Series 2) card. Use the PPL Configuration Entity of 0x07 (DTMF timers).

Table 9-19 DTMF Timers

Timer ID (Hex)	Timer Name	Default Value
0x01	DTMF Digit Duration	60 ms
0x02	DTMF Interdigit Duration	60 ms
0x03	DTMF Max. Receive First Digit Detect	20 seconds (0xFFFF disables)
0x05	DTMF Min. Receive Digit Duration	40 ms
0x06	DTMF Min. Receive Interdigit Timeout	40 ms

DTMF Digit Reception for User Input Indication

When an IP channel is connected to a PSTN channel a DTMF receiver needs to be attached to the PSTN channel to receive DTMF digits. By default, the receiver is not attached and the host can control when the receiver is attached and released by sending the relevant PPL Event Requests.

The purpose of the DTMF timers above is similar to that in the *Collect Digit String (0xBC)* host message. The default value for the First Digit Detect timer is 20 seconds. If a digit is not received within this time, the DTMF receiver will time out, will remain attached, and the channel will not purge. The connection will not be affected.

If no action is taken by the host, the timer will restart and time out again after the configured time. This scenario will occur continuously until the call is terminated or the host releases the receiver.

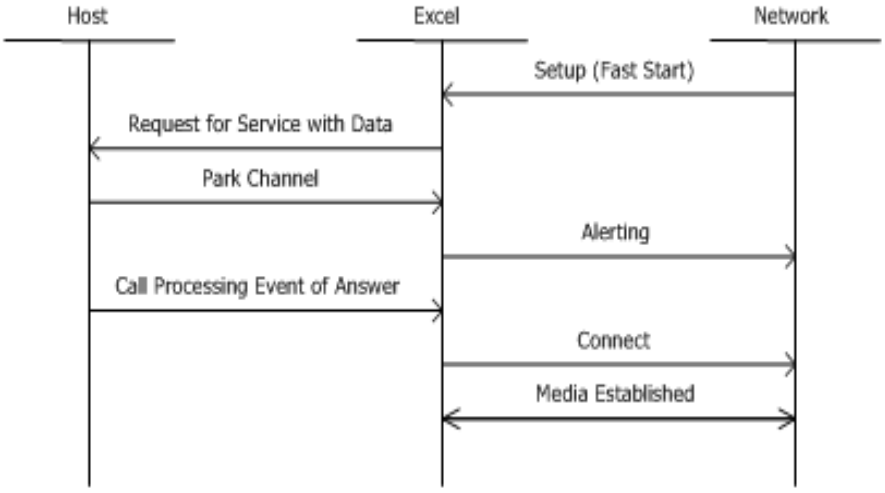
The timer may be disabled by setting its value to 0xffff. In this case the DTMF receiver will not time out. A PPL Event Indication of DTMF Digit Receive Timeout (0x04) will be sent to the host each time the DTMF digit receive timer expires.

The DTMF maximum receive interdigit timeout value means nothing in this case as the DTMF receiver is configured to receive a single digit in a single string and so will report the digit when it has been received.

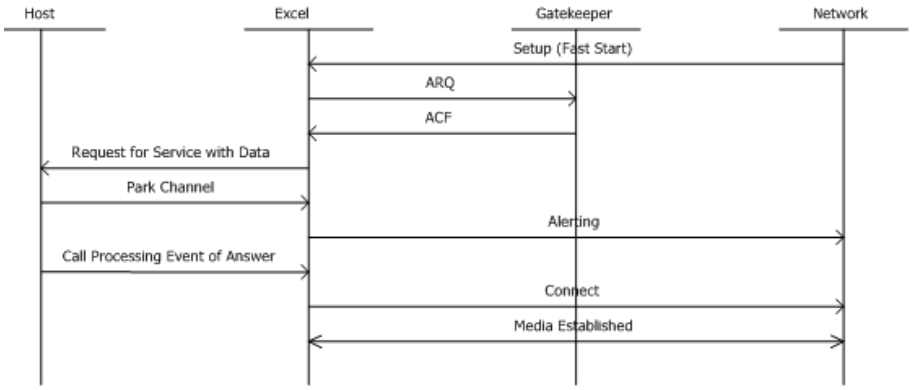
Call Flows

Purpose The following diagrams depict some of the basic H.323 call flows. They are examples and not necessarily functional call flows. Refer to Appendix C, *H.323 Call Flows and Message Traces (C-1)* for call flows that include the message traces.

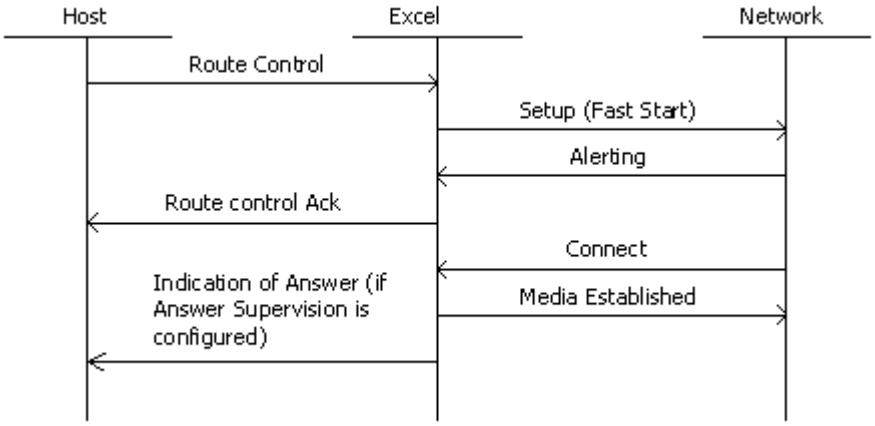
H.323 Incoming Call - Fast Connect, No Gatekeeper



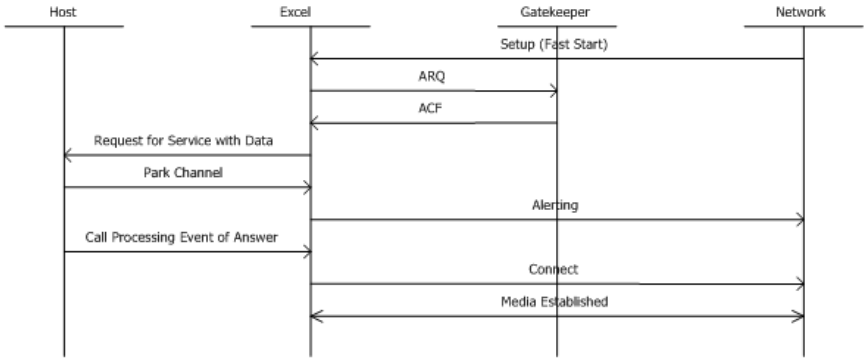
H.323 Incoming Call - Fast Connect, Gatekeeper



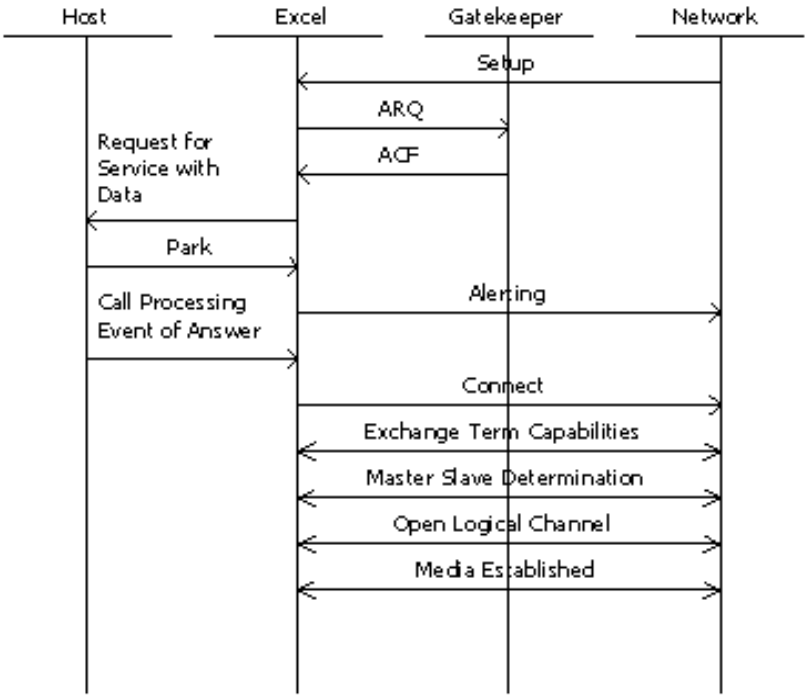
H.323 Outgoing Call - Fast Connect, No Gatekeeper



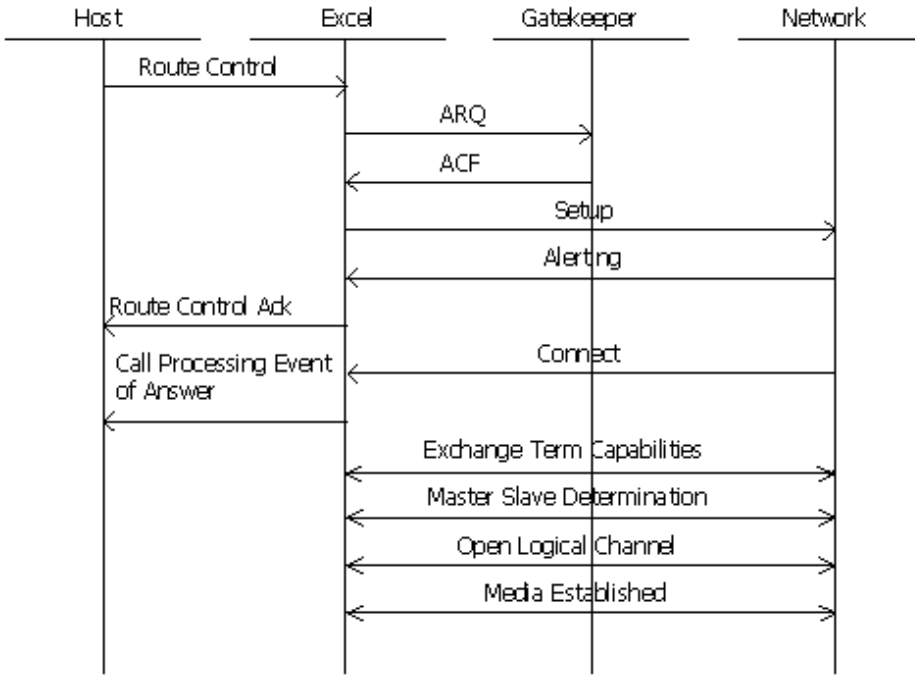
H.323 Incoming - Fast Connect, Gatekeeper



H.323 Incoming Call with H.245 Procedure

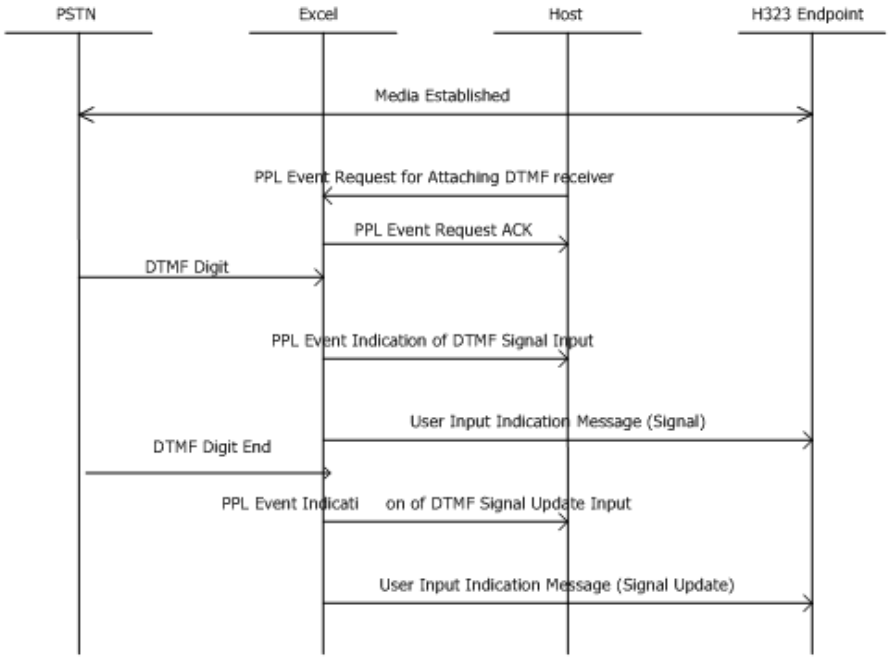


H.323 Outgoing Call with H.245 Procedure

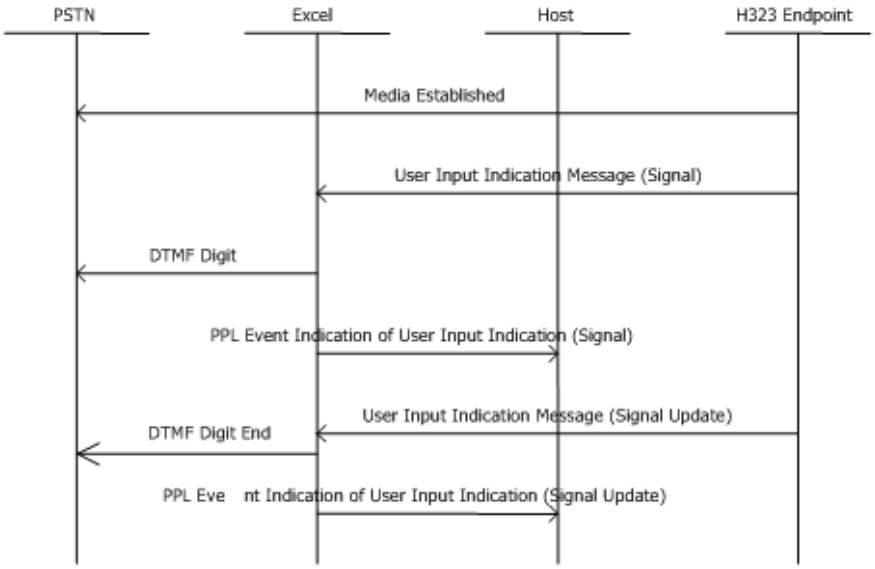


User Input Indication for Pre-Paid Applications

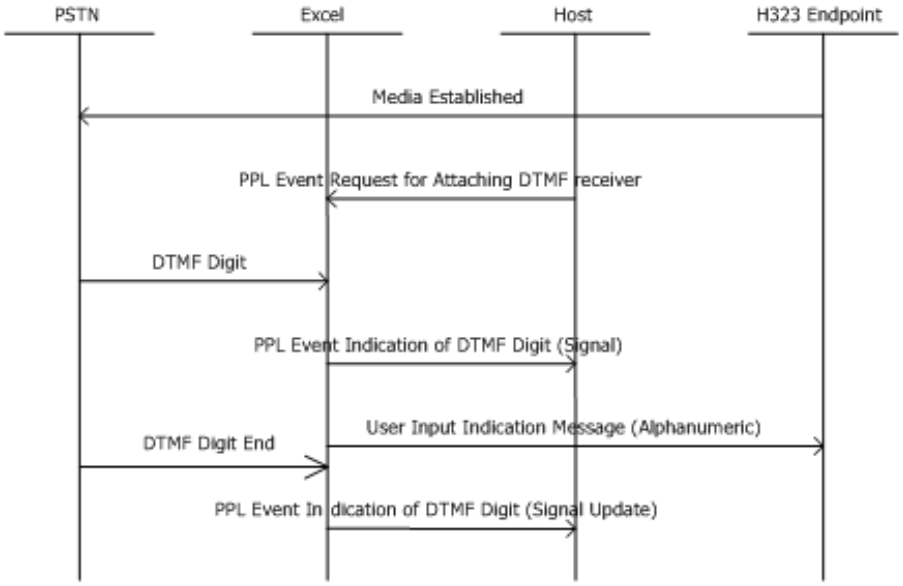
H.323 Endpoint supports DTMF - Digits received from PSTN side



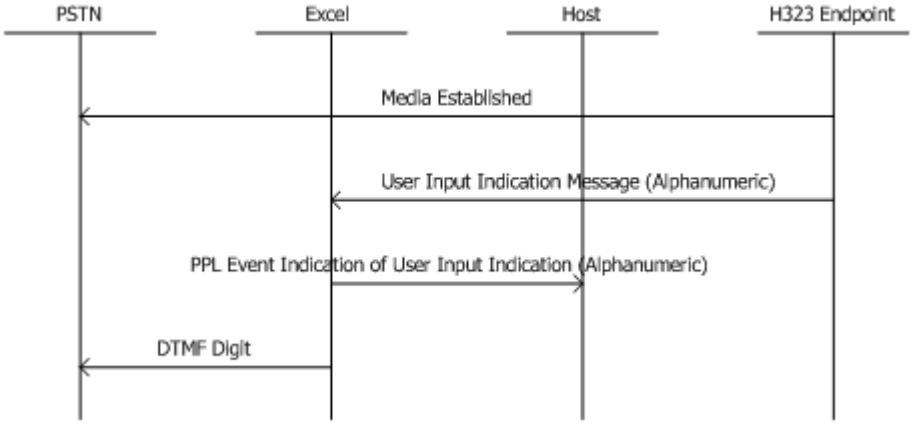
H.323 Endpoint supports DTMF - UII received from IP side



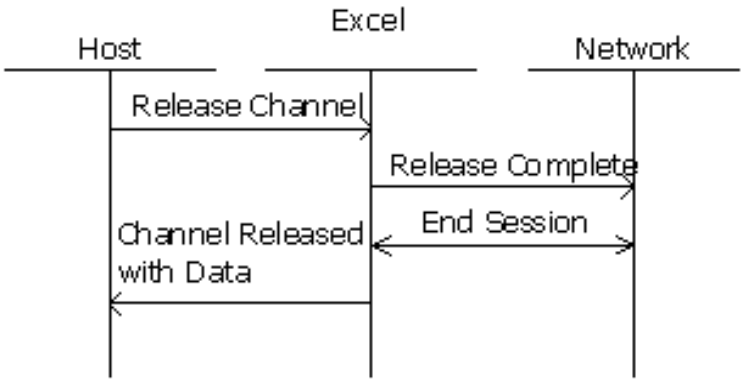
H.323 Endpoint supports only Alphanumeric - Digits received from the PSTN side



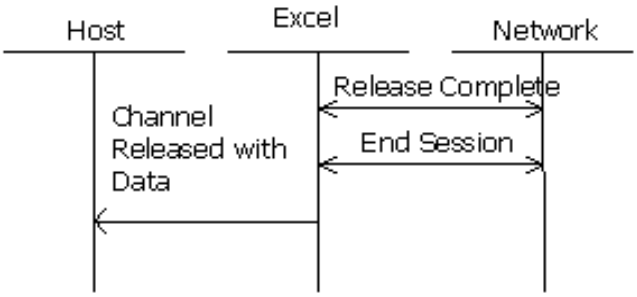
H323 Endpoint supports only Alphanumeric - Digits are received from IP side



Call Release Call Flows Host Initiated Call Release



Network Initiated Call Release



A H.323 Support and Compliance

Purpose

This Appendix briefly describes the H.323 Version 2 support and compliance including:

- Offer Overview
- H.323 Version 2 Compliance
- H.323 offer in relation to IMTC Testing and iNOW! Profile criteria.
- Restrictions and Limitations
- Supported Standards MIBs and RFCs
- Related Documents

H.323 Offer Overview

The CSP H.323 implementation enhances the CSP software to comply with the mandatory requirements of the H.323 Version 2 specification. This feature enhances the existing Voice over IP offering of the VDAC-ONE card for RTP bearer channels.

General Information

- Conforms to the H.323 ITU Protocol Version 2
  - H.225 Call Signaling Protocol
  - H.245 Control Protocol
  - Registration, Admission & Status (RAS)
  - Real-time Transport Protocol/Real-time Transport Control Protocol (RTP/RTCP)
  - Q.931
- Gateway functionality only: Not a Gatekeeper Implementation

- Supports Fast Connect
- Supports H.323-to-SIP Interworking
- Provides well defined EXS API messages for ease of development
- Interoperability
  - Backward compatible with the ITU H.323 Version 1 Protocol
  - Interoperates with leading PC Clients
  - Interoperates leading VoIP Gateways at the RTP level.

Media Processing Features Supported

- G.711 CODEC (A-law and µ-law) - 64 Kbps
- G.729/G.729A/B CODEC - 8 Kbps
- G.723.1 CODEC - 5.3 Kbps and 6.3 Kbps
- Voice Activity Detection (VAD)
- Comfort Noise Generation (CNG)
- Jitter Buffer Management
- Echo Cancellation
- DTMF generation/detection: Supported by the DSP Card. (G.711 CODEC only)

H.323 Version 2 Compliance

The CSP H.323 implementation enables the CSP to send and receive all the REQUIRED or mandatory fields in the H.323 Version 2 messages.

Lightweight Registration

The CSP H.323 implementation enhances the CSP software to comply with the mandatory requirements of the H.323 Version 2 specification. This includes the Lightweight Registration Feature. H.323 Version 2 defines a lightweight registration procedure that still requires the full registration, but uses an abbreviated renewal procedure to update the gatekeeper and subsequently reduces overhead.

Lightweight registration requires each endpoint to specify a Time-to-Live (TTL) value in its *Registration Request (RRQ)* message. When a gatekeeper receives an RRQ message from the CSP with a TTL value, it returns an updated TTL timer value in a *Registration Confirmation (RCF)* message to the endpoint. Prior to the TTL timer expiring, the endpoint sends an *RRQ* message with the KeepAlive field set to “Yes”, which refreshes the existing registration.

An H.323 Version 2 endpoint is not required to indicate a time-to-live in its registration request. If the endpoint does not indicate a time-to-live, the gatekeeper assigns one and sends it to the gateway in the *RCF* message. No configuration changes are permitted during a lightweight registration.

DTMF Support

Dual-Tone Multi-Frequency (DTMF) is the tone generated on a touch-tone phone when you press keypad digits. During a call you might enter the DTMF to access Interactive Voice Response (IVR) systems such as voicemail, calling card services, and so on. DTMF is transported the same way as voice in the PSTN.

There are two methods of providing this DTMF capability in an IP network:

- Out-of-band:
    - H.323 provides the DTMF digit support using the User Input Indication (UII) H.245 message.
- In-band:
    - Media gateways or the VDAC-ONE card can provide DTMF digits in separate RTP packet formats.
    - Media streams can accurately transmit DTMF digits “in-band” using high bit-rate CODECS. Media gateways or the VDAC-ONE card with high bit-rate CODECS can reproduce DTMF digits in the same manner as it creates voice packets.

Converged networks contain media gateways and other devices with high bit-rate and/or low bit-rate CODECS. High bit-rate CODECS, such as G.711, G.726, and G.727, reproduce DTMF digits accurately. However, low bit-rate CODECS, such as G.729, and G.723.1 require DTMF tones to be put in a separate RTP payload format because they cannot reproduce DTMF tone signals accurately enough for automatic recognition. Using separate RTP payload formats for both high and low bit-rate CODECS is one way to address DTMF tone reproduction in a standard way in the IP media stream.

The CSP is capable of supporting all methods of handling DTMF digits for ultimate partner flexibility. The current plan for DTMF digit support in this version of the CSP system is as follows:

- Out-of-band DTMF digit support:
    - H.323 signaling support with the H.245 message set

- In-band DTMF digit support with VDAC-ONE for high bit-rate CODECS such as G.711, G.726, and G.727:
    - VDAC-ONE RTP payload formats (all CODECS)

Hookflash Relay

A 'hookflash" indication is a brief on-hook condition that occurs during a call. For example during a call, a phone user quickly depresses and then releases the hook switch on their telephone. It is not long enough to be interpreted as a signal to disconnect the call, but telephone switches and PBXs are frequently programmed to intercept hookflash indications and use them as a way to allow a user to invoke supplemental services. For example, your local service provider may allow you to enter a hookflash as a means of switching between calls if you subscribe to a call waiting service.

In the traditional telephone network, a hookflash results in a voltage change on the telephone line. Since there is no equivalent of this voltage change in an IP network, the ITU H.245 standard defines a message representing a hookflash. To send a hookflash indication using this message, an H.323 endpoint sends an *H.245 User Input Indication* message containing a "signal" structure with a value of "!". The value represents a hookflash indication.

The CSP H.323 implementation includes support for relaying hookflash indications using the *H.245 User Input Indication* message.

CODEC Negotiation

CODEC negotiation allows the gateway to offer several CODECS during the H.245 capability exchange phase and ultimately settle upon a single common CODEC during the call setup. This increases the probability of establishing a connection because there is a greater chance of overlapping audio capabilities between endpoints. During the call setup phase, the gateway uses the highest priority CODEC selected from the list that it has in common with the remote endpoint. It also adjusts to the CODEC selected by the remote endpoint so that a common CODEC is established for the receive and transmit audio directions.

H.323 Offer in Relation to IMTC Testing and iNOW! Profile Criteria

IMTC stands for the International Multimedia Telecommunications Consortium, Inc., a non-profit corporation comprising more than 120 organizations around the globe. The IMTC's mission is to promote and

facilitate the development and implementation of interoperable, multimedia, conferencing solutions based on open international standards -- particularly the multimedia conferencing standards adopted by the International Telecommunication Union (ITU), as well as other standards organizations.

Table A-1 IMTC VoIP H.323 Scoresheet - CSP as a Gateway Entity

Elements	Supported	Notes
Call Direction: (O)riginate/ (R)ecieve	Yes	
GRQ/GCF RRQ/RCF URQ/UCF	Yes	
BRQ/BCF	Yes	
IRQ/IRR/IACK/INAK	Yes	
GK Routed Call	Yes	
Gateway Resource Availability	No	
PreGranted ARQ	Yes	
Q.931 Connection	Yes	
Fast Start	Yes	
Overlapped Sending	No	
H.245 Tunneling	No	
Q.931 Multiplexing	No	
H.245 Connection Establishment	Yes	
Empty Term Cap Set	Yes	
MC Cascading	No	
Supplementary Services H.450	No	
ReplacementFor	No	
Request Mode	No	Future Release
CommunicationModeCommand	No	
H.235	No	
T.38 FoIP	No	Future Release
IP Addressing	Yes	
E.164 Alias Addressing	Yes	
H.323 ID Alias Addressing	Yes	
UniCast RTP/RTCP	Yes	
MultiCast RTP/RTCP	No	
G.711	Yes	

Elements	Supported	Notes
G.726	No	
G.723.1	Yes	
G.722/G.722.1	No	
G.728	No	
G.729 (A, B)	Yes	G.729C is not supported
H.261	No	
H.263	No	
Data Channel (T.120)	No	

IMTC iNOW!

IMTC iNOW! stands for “interoperability NOW!”. IMTC iNOW! is an industry initiative to achieve interoperability as soon as possible between different vendor's IP telephony equipment platforms. IMTC iNOW! is not a standard and it is not proprietary.

IMTC iNOW! is an initiative that is supported by the major players in IP telephony including Dialogic, with new supporters being added all the time. Supporters of this initiative have committed to building or buying equipment that follows the IMTC iNOW! Profiles, publicly available documents that draw upon existing standards such as H.323. Any vendor that follows these Profiles will have the blueprint to build or operate IMTC iNOW! compliant equipment.

Table A-2 IMTC iNOW! Profile Testing Criteria

Elements	Supported	Notes
SRV Discovery	No	Future Release
TXT Discovery	No	Future Release
A Record Discover	No	Future Release
Pre-Configured ID Discovery	Yes	
GRQ Discover	Yes	
GRQ/GCF RRQ/RCF URQ/UCF	Yes	
GK Routed Call	Yes	
Pre-Granted ARQ	Yes	
Q.931 Connection	Yes	
Fast Start	Yes	
H.245 Tunneling	No	
Lightweight RRQ(Keep Alive)	Yes	
H.323 Annex J Authentication	No	
H.323 Annex J Integrity	No	
E.164 Addressing	Yes	
E-mail Addressing	Yes	
Keypad entries (Q.931 Info Msg.)	No	
Alternate GK	Yes	
Status Inquiry/Status	Yes	
Unicast RTP/RTCP	Yes	
G.711	Yes	
G.723.1	Yes	
G.729 (A, B)	Yes	G.729C is not supported
H.245v4	No	Future Release
H.245v6 syntax for Fax	No	Future Release

Elements	Supported	Notes
Interoperability with Simple Endpoint Type (SET) devices implementing the minimum requirements as defined by H.323 Annex F	No	Future Release

Restrictions

The CSP H.323 Version 2 offering does not operate on Release 5.x. If you are planning to use an CSP H.323 Version 2 Platform, your software requires Release 8.0 or later and the following hardware:

- CSP Matrix Series 3 Card
- IP Signaling Series 3 card
- VDAC card or IP Network Interface card for relaying RTP streams
- DSP-ONE or DSP Series 2 card for detecting/generating DTMF Tones

Feature Limitations:

- Gatekeeper Functionality NOT supported
- Voice Only - Video is NOT supported
- No redundancy in the IP Signaling Series 3 card in this release.

Supported Standard MIBs and RFCs

- No MIBs are supported by this feature release.
- This feature provides support for the ITU H.323 V2 teleconferencing standard.

B SIP Support and Compliance

Purpose This release of the SIP User Agent implementation supports the following methods, responses, and headers per SIP RFC 2543 Bis-05.

Methods SIP call control methods identify the type of request that is being made.

Table B-1 Methods

Method	Supported	Description	Information Reported to Host
INVITE	Yes	Invites a target to participate in a session; establishes a connection. Also used to change call state or capabilities, such as CODEC used. Supported with or without SDP because of the Delayed Media feature.	The *Request for Service* message reports the receipt of the INVITE method using the following TLVs: SIP Extensions (0x294A) To Username (0x2919) To Password (0x291A) To Hostname (0x291B) To Port, Tags (0x291C) From Username (0x2923) From Password (0x2924) From Hostname (0x2925) From Port (0x2926) From Display name (0x2928) From Tags (0x2951) Contact Username (0x292D) Contact Password (0x292E) Contact Hostname (0x292F) Contact Port (0x2930) Contact Expires (0x2931) Contact Transport Type (0x2933) Contact URI Parameters (0x2935) Contact Display Name (0x292B) Contact Header Parameters (0x292C) Request Username (0x2954) Request Hostname (0x2955) Request Port (0x2956) Request URI Parameters (0x2958) Request Header Parameters (0x2947)

Method	Supported	Description	Information Reported to Host
			To & From Username are also reported in Called Party Number TLV (0x2717) and Calling Party Number TLV (0x2718) formats respectively Call ID (0x2950) Remote-Party-ID (0x2959) RPID-Privacy (0x295A) Subject Header (0x295B) Top VIA Header. Privacy, P-Asserted-Identity, P-Preferred-Identity, P-Access-Network-Info. Referred-By URI (0x2929) Referred-By parameters (0x292A) Remote Signaling IP address & Port for TCP Calls 0x294E & 0x294F respectively. Media destination IP address, Port, Payload Type and Payload size in the SDP for non-CAM. For CAM it is reported within the Media Local End Point Information (0x29FF).
Re-INVITE (Offer)	Yes	Used to change media used for the call and to put a call on/off hold	PPL Event Indication to report receipt of ReINVITE in the following TLV: If present, SDP is reported within the Local End Point Info TLV (0x29FF)
180 Ringing	Yes	The called party is alerted	PPL Event Indication to report receipt of 180 received: SIP Extensions TLV (0x294A)

Method	Supported	Description	Information Reported to Host
182 Called Queued	Yes	The called party is unable to take the call immediately so the calling party is put on hold.	PPL Event Indication to report receipt of 182 received: SIP Extensions (0x294A) SIP Response Reason Phrase (0x2949)
183 Session Progress	Yes	Informs about the progress of the call. Mostly used to establish an early media connection.	PPL Event Indication to report receipt of 183 received: SIP Extensions 0x294A Media Local End Point Information (0x29FF)
PRACK	Yes	Makes the non-100 provisional responses reliable.	*PPL Event Indication* to report receipt of PRACK received. No data reported.
ACK	Yes	Acknowledges receipt of a final response to an invite request	Not reported to the host.
ACK with SDP	Yes	Reports receipt of ACK with SDP in case of delayed media scenario.	PPL Event Indication to report receipt of ACK with SDP contains the following TLV: SDP if present is reported within the Local End Point Information (029FF)
BYE	Yes	Indicates that either the originator or the target wishes to end the call after the INVITE transaction is successfully completed.	Channel released 0x49 in case of CAM. Channel released with data 0x69 in case of non-CAM calls.
CANCEL	Yes	Indicates that the originator wishes to cancel the INVITE request it had sent.	The host receives a channel-released messages.

Method	Supported	Description	Information Reported to Host
REGISTER	Yes	Registers with a server	The *PPL Event Indication* message reports the receipt of the REGISTER message using the TLVs below. The CSP does not generate the REGISTER message. SIP To Username (0x2919) SIP To Password (0x291A) SIP To Hostname (0x291B) SIP To Port (0x291C) SIP From Username (0x2923) SIP From Password (0x2924) SIP From Hostname (0x2925) SIP From Port (0x2926) SIP Contact Username (0x292D) SIP Contact Password (0x292E) SIP Contact Hostname (0x292F) SIP Contact Port (0x2930) SIP Contact Expires (0x2931) SIP Contact Transport Type (0x2933)
REFER	Yes	Indicates that the recipient should contact a third party using provided contact information; initiates a transfer.	The PPL Event Indication reports the receipt of the REFER method in the following TLVs: SIP Request URI User Name (0x2954) SIP Request URI Password (0x2946) SIP Request URI Host Name (0x2955) SIP Request URI Port (0x2956) SIP Request URI Parameters (0x2958) SIP Request URI Header (0x2947) SIP Refer To Username (0x291E) SIP Refer To Password (0x291F) SIP Refer To Hostname (0x2920) SIP Refer To Port (0x2921) SIP Refer To Header Parameter (0x2922) SIP Referred By URI (0x2929) SIP Referred By Parameter (0x292A) The PPL Event Indication reports the received for REFER (Blind and Consultative) SIP Response Code (0x2915)

Method	Supported	Description	Information Reported to Host
NOTIFY	Yes	Provides information about a state change; not related to specific session.	The PPL Event Indication to report the receipt of the NOTIFY and the REFER request (Blind and Consulative) sent out. SIP Response Code 0x2915 The CSP support generation of the NOTIFY message.
SUBSCRIBE	Yes	Used by a SIP entity to declare its interest in a particular event. SIP entity subscribes to a certain event of a class of events.	Reporting is not supported. The CSP supports generation of the SUBSCRIBE method.
INFO	Yes	Transports any mid call information.	PPL Event Indication to report receipt of INFO received: SIP Message Body (0x295C) SIP Content Type (0x295D) PPL Event Indication to report the response for INFO sent out: SIP Response Code (0x2915)
OPTIONS	Yes	Solicits information about features supported by SIP servers such as supported methods and media capabilities.	PPL Event Indication reports receipt of OPTIONS in the following TLVs: SIP Request URI User Name (0x2954) SIP Request URI Password (0x2946) SIP Request URI Host Name (0x2955) SIP Request URI Port (0x2956) SIP Request URI Parameters (0x2958) SIP Request URI Header (0x2947) CSP does not support generation of OPTIONS method.

Responses

SIP response codes indicate the status of a session. Responses with 1xx response codes are also call provisional responses, while the remaining response codes (2xx, 3xx, 4xx, 5xx, and 6xx) indicate final responses.

This release of the SIP Implementation treats the 4xx, 5xx, and 6xx responses as rejected calls.

Table B-2 Responses

Code Type	Function
1xx	Informational: Trying, ringing, forwarding, queuing, in progress
2xx	Successful: OK
3xx	Redirection: Indicate additional information for call forwarding
4xx	Request Failure: Indicate request errors such as missing information
5xx	Server Failures: Time outs, unavailable services, and other server errors
6xx	Global Failures: Busy declined, not found, not acceptable

Responses	Supported	Notes
100 Trying	Yes	
180 Ringing	Yes	The called party is alerted. *PPL Event Indication* message reports receipt of 180 Ringing in the following TLV: SIP Extensions (0x294A)
181 Forwarded	Yes	Treated as 180 Ringing Response.

182 Called Queued	Yes	The called party is unable to take the call immediately so the calling party is put on hold. *PPL Event Indication* message reports receipt of 182 Queue using the following TLVs: SIP Extensions (0x294A) SIP Response Reason Phrase (0x2949)
183 Session Progress	Yes	Informs the progress of the call. Mostly used to establish an early media connection. *PPL Event Indication* message reports receipt of 183 Session Progress using the following TLVs: SIP Extensions 0x294A Media Local End Point Information (0x29FF)
200 OK	Yes	PPL Event Indication to report receipt of Answer using the following TLVs: SIP Contact Display Name (0x292B) SIP Contact Parameters (0x292C) SIP Contact Username (0x292D) Media Local End Point Information (0x29FF)
300 Multiple Choices	Yes	Supports Redirections.
301 Moved Permanently	Yes	
302 Moved Temporarily	Yes	
305 Use Proxy	Yes	
380 Alternative Service	Yes	
400 Bad Request	Yes	For outbound call (UAC), these responses are treated as Call Rejected.

401 Unauthorized	Yes	Supports basic and digest.
402 Payment Required	No	Treated as Call Rejected.
403 Forbidden	No	Treated as Call Rejected.
404 Not Found	Yes	Generate and accept responses. For outbound calls, these responses are treated as Call Rejected.
405 Method Not Allowed	Yes	Generate and accept responses. For outbound calls, these responses are treated as Call Rejected.
406 Not Acceptable	No	Treated as Call Rejected.
407 Proxy Authentication Required	Yes	Treated as Call Rejected.
408 Request Timeout	No	Treated as Call Rejected.
409 Conflict	No	Treated as Call Rejected.
410 Gone	No	Treated as Call Rejected.
411 Length Required	No	Treated as Call Rejected.
413 Request Entity Too Large	No	Treated as Call Rejected.
414 Request – URI Too Long	No	Treated as Call Rejected.
415 Unsupported Media	Yes	Generate and accept responses. For outbound calls, these responses are treated as Call Rejected.
420 Bad Extension	Yes	Generate and accept responses. For outbound calls, these responses are treated as Call Rejected.
422 Session Timer Duration Too Small	Yes	Retry the INVITE method with new session interval.
480 Temporarily Not Available	Yes	Generate and accept responses. For outbound calls, these responses are treated as Call Rejected.
481 Call Leg Transaction Does Not Exist	Yes	Generate and accept responses. For outbound calls, these responses are treated as Call Rejected.
482 Loop Detected	No	Treated as Call Rejected.
483 Too Many Hops	No	Treated as Call Rejected.

484 Address Incomplete	No	Treated as Call Rejected.
485 Ambiguous	No	Treated as Call Rejected.
486 Busy Here	No	Treated as Call Rejected. For outbound calls, these responses are treated as Call Rejected.
500 Server Internal Error	Yes	Treated as Call Rejected. For outbound calls, these responses are treated as Call Rejected.
501 Not Implemented	No	Treated as Call Rejected.
502 Bad Gateway	No	Treated as Call Rejected.
503 Service Unavailable	Yes	Generate and Accept Responses.
504 Gateway Timeout	No	Treated as Call Rejected.
505 SIP Version Not Supported	Yes	Generate and Accept Responses.
513 Message Too Large	Yes	Treated as Call Rejected.
600 Busy Everywhere	No	Treated as Call Rejected.
603 Decline	No	Treated as Call Rejected.
604 Does Not Exist Anywhere	No	Treated as Call Rejected.
606 Not Acceptable	No	Treated as Call Rejected.

Message Headers

Each SIP message is accompanied by a header. The following are the required fields in all messages (see Table B-1.)

Table B-3 Required Fields

Required Field	Description
From	The address of the session originator; expressed as a SIP URL.
To	The address of the session target; expressed as a SIP URL.
Call-ID	A unique identifier assigned to all of the SIP messages related to a call.
Cseq	The SIP call control method and an identifying sequence number

Table B-4 Supported Headers

Header	Supported	Compressed Header
Accept	No	
Accept-Encoding	Yes	Not applicable
Accept-Language	No	
Allow	Yes	
Authorization	Yes	
Call-ID	Yes	i
Contact	Yes	m
Content-Disposition	Yes	Not applicable
Content - Encoding	No	
Content-Length	Yes	l
Content-Type	Yes	c
Cseq	Yes	Not applicable
Date	No	
Encryption	No	
Expires	Yes	Not applicable
From	Yes	f
Hide	No	
In-Reply-To	No	
Max - Forwards	No	
MIME - Version	No	
Min-SE	Yes	
Organization	No	
Priority	No	
Proxy - Authenticate	Yes	
Proxy - Authorization	Yes	
Proxy - Require	No	
Record-Route	Yes	Not applicable
Require	Yes	

Header	Supported	Compressed Header
Retry-After	Yes	Not applicable
Response-Key	No	
Route	Yes	Not applicable
Server	No	
Session-Expires	Yes	
Subject	No	
Supported	Yes	
Timestamp	No	
To	Yes	t
Unsupported	No	
User-Agent	Yes	Not applicable
Via	Yes	v
Warning	No	
WWW-Authenticate	Yes	

Important! You must set the System Timer accurately on the CSP Matrix Series 3 Card.

SIP Addresses

Instead of being assigned to specific devices the way that telephone numbers traditionally are assigned, SIP URLs are assigned to the individual users who participates in SIP sessions. As a result, when a phone call (or other interactive session) is made to a SIP address, it can be routed to the appropriate individual regardless of a change in physical location or IP telephone device.

The From and To fields in every message header contain the SIP URLs of the session's originator and target.

SIP URLs use the same format as IP addresses or e-mail addresses:

<sip:user-id@server-address:port;parameters;?header-names=values>.

The formats of some sample SIP URLs might appear as:

sip:123@OPENet.com

sip:3333@10.1.1.123

sip:johndoe@sip.OPENet.net:5070

John Doe<sip:johndoe@sip.OPENet.net:5070>

SIP Interoperability Compliance with the CSP

The following is required for SIP Interoperability compliance with the CSP:

- SIP User Agent supports RFC 2543 Bis-05 and RFC 2327.
- SDP provided in Invite for Media negotiation.
- RTP with G.711 must be supported.

Limitation of Codec Selection

Normally, a SIP INVITE message specifies the CODECS to be used in the media description part of the SDP. The CODECS that appear first in the list are those with a higher priority for usage. However, the CSP requires the use of a route table that specifies a CODEC (any CODEC supported by CSP).

If the CODEC listed in the route table appears anywhere in the media description of the SDP (regardless of priority) then that CODEC is used first.

Example: The Pulse Code Modulation U Law (PCMU) is specified in the route table. The SDP specifies Pulse Code Modulation A Law (PCMA) first in the media description and PCMU second. Normally PCMA is used if the other end supports it, but because the way the CSP handles the call, PCMU is used because it appears in the media description list. If PCMU did not appear in the media description, then PCMA would be used.

C H.323 Call Flows and Message Traces

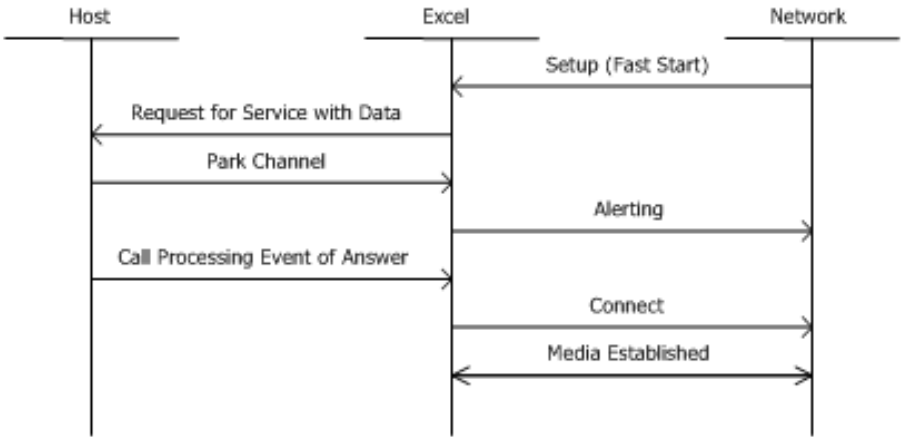
Purpose

The following diagrams depict some of the basic H.323 call flows. They are examples and not necessarily functional call flows.

The corresponding message traces follow the diagrams. These message traces include output from a protocol analyzer and examples of API messages.

Call Flows

H.323 Incoming Call - Fast Connect, No Gatekeeper



Message Trace

```
 Receive Setup
----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 2D71
      H225:  0... ....  .... .... = Message from
   originator
     H225:  .010 1101  0111 0001 = Call reference value =
   11633
      H225: Message type = 5 (Setup)
      H225:
      H225: Bearer Capacity
      H225: Information element identifier = 4
      H225: Length of bearer capacity = 3
      H225: Coding and capability flags = 88
      H225:                    1... .... = Expected
   extension bit
      H225:                   .00. .... = Coding standard
   = 0 (CCITT standardized coding)
      H225:                    ...0 1000 = Information
   transfer capability = 8 (Unrestricted digital
             information)
```

```
      H225: Mode and rate flags = C0
      H225:           1... .... = Expected extension bit
      H225:          .10. .... = Transfer mode = 2 (Packet
   mode)
      H225:          ...0 0000 = Information transfer rate
   = 0 (Used for packet mode calls)
      H225: Layer 1 protocol flag = A5
      H225:            1... .... = Expected extension bit
      H225:            .01. .... = Layer 1 identification
   = 1
      H225:             ...0 0101 = Layer 1 protocol = 5
   (H.221 and H.242)
      H225:
      H225: Display
      H225: Information element identifier = 40
      H225: Display length = 7
      H225: Display flag = 4C
      H225:    0... .... = Expected extension bit
      H225: Display = "Excel"
      H225:
      H225: Calling Party Number
      H225: Information element identifier  = 108
      H225: Calling party number contents length = 11
      H225: Type and numbering flags = FF
      H225:               1... .... = Expected extension
   bit
      H225:               .111 .... = Type of number = 7
   (Reserved for extension)
      H225: .... 1111 = Numbering plan ID = 15 (Reserved
   for extension)
      H225: Number digits flag = 35
      H225:          0... .... = Expected extension bit
      H225: Number digits = "5088623456"
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 578
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 20
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..1. .... = Extension values present in h323-
   uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
```

```
   H225: h323-message = 0 (Setup)
   H225:
   H225: Flags     = A8
   H225: 1... .... = Extension value(s) present in
Setup
   H225: .0.. .... = h245Address is not present
   H225: ..1. .... = sourceAddress is present
   H225: ...0 .... = destinationAddress is not present
   H225: .... 1... = destCallSignalAddress is present
   H225: .... .0.. = destExtraCallInfo is not present
   H225: .... ..0. = destExtraCRV is not present
   H225: .... ...0 = callServices is not present
   H225:
   H225: Protocol id = {0.0.8.2250.0.2}
   H225:
   H225: Number of sourceAddress = 2
   H225:
   H225: Flags     = 40
   H225: 0... .... = No extension value(s) present in
sourceAddress
   H225: sourceAddress = 1 (H323-ID)
   H225: H323-ID = "Excel"
   H225:
   H225: Flags     = 04
   H225: 0... .... = No extension value(s) present in
sourceAddress
   H225: sourceAddress = 0 (E164)
   H225: Length of e164 = 10
   H225: E.164 = 5088623456
   H225:
   H225: Flags     = 22
   H225: 0... .... = No extension value(s) present in
sourceInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..1. .... = Vendor is present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
vendor
   H225: Flags     = C0
   H225: 1... .... = Product ID is present
   H225: .1.. .... = Version ID is present
   H225:
   H225: ..0. .... = No extension value(s) present in
vendor
   H225: T.35 country code = 0x 9 (Australia)
   H225: T.35 extension    = 0
   H225: Manufacture code = 0x  3D (?)
```

```
   H225: Product Id = Equivalence OpenPhone<0000>
   H225: Version Id = 1.2.0<0000>
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
terminal
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = MC = 0
   H225: ...0 .... = Undefined node = 0
   H225:
   H225: destCallSignalAddress
   H225:
   H225: .... 0... = No extension value(s) present in
destCallSignalAddress
   H225: destCallSignalAddress = 0 (IP address)
   H225: IP   = 135.119.55.181
   H225: Port = 1720
   H225:
   H225: Flags     = 00
   H225: 0... .... = Active MC = 0
   H225: Conference id =
3FB87FC806EA18109CD80050DAD08480
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
conferenceGoal
   H225: Conference goal = 0 (Create)
   H225:
   H225: callType
   H225: ...0 .... = No extension value(s) present in
callType
   H225: callType = 0 (Point to point)
   H225:
   H225: .... ..0. = Extension length determinant
   H225: Number of extension = 12
  H225: .... .1.. = sourceCallSignalAddress is present
   H225: .... ..0. = remoteExtensionAddress is not
present
   H225: .... ...1 = callIdentifier is present
   H225: Flags     = 1D
   H225: 0... .... = h245SecurityCapability is not
present
   H225: .0.. .... = tokens is not present
   H225: ..0. .... = cryptoTokens is not present
   H225: ...1 .... = fastStart is present
   H225: .... 1... = mediaWaitForConnect is present
   H225: .... .1.. = canOverlapSend is present
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
sourceCallSignalAddress
   H225: sourceCallSignalAddress = 0 (IP address)
```

```
      H225: IP   = 135.119.55.170
      H225: Port = 2302
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   callIdentifier
      H225: Guid = 3FB87FC806EA18109CD90050DAD08480
      H225: Number of fastStart = 440

FastStartElement Received
40 00 00 06 04 01 00 4c 10 09 00 00 3d 06 4c 50 43 2d 31
   30 80 11 1c 00 01 00 87
 77 37 aa
13 88 00 87 77 37 aa 13 89

FastStartElement Received
00 00 64 0c 10 09 00 00 3d 06 4c 50 43 2d 31 30 80 0b 0d
   00 01 00 87 77 37 aa 13
 89 80

FastStartElement Received
40 00 00 06 04 01 00 4e 0c 03 00 83 00 80 11 1c 00 01 00
   87 77 37 aa 13 88 00 87
 77 37 aa
13 89

FastStartElement Received
00 00 65 0e 0c 03 00 83 00 80 0b 0d 00 01 00 87 77 37 aa
   13 89 80

FastStartElement Received
40 00 00 06 04 01 00 4c 10 b5 00 53 4c 2a 02 00 00 00 00
   00 40 01 00 00 40 01 02
 00 08 00
00 00 00 00 31 00 01 00 40 1f 00 00 59 06 00 00 41 00 00
   00 02 00 40 01 00 00 80
 12 1c 40
01 00 87 77 37 aa 13 88 00 87 77 37 aa 13 89 00

FastStartElement Received
00 00 66 0c 10 b5 00 53 4c 2a 02 00 00 00 00 00 40 01 00
   00 40 01 02 00 08 00 00
 00 00 00
31 00 01 00 40 1f 00 00 59 06 00 00 41 00 00 00 02 00 40
   01 00 00 80 0b 0d 40 01
 00 87 77
37 aa 13 89 80
FastStartElement Received
40 00 00 06 04 01 00 4c 20 1d 80 11 1c 00 01 00 87 77 37
   aa 13 88 00 87 77 37 aa
 13 89
```

```
FastStartElement Received
00 00 68 0c 20 1d 80 0b 0d 00 01 00 87 77 37 aa 13 89 80
FastStartElement Received
40 00 00 06 04 01 00 48 71 03 51 00 80 01 00 80 11 1c 00
   02 00 87 77 37 aa 13 8a
 00 87 77
37 aa 13 8b
FastStartElement Received
40 00 00 06 04 01 00 48 6b 03 51 00 80 01 00 80 11 1c 00
   02 00 87 77 37 aa 13 8a
 00 87 77
37 aa 13 8b


Request for Service with Data
X->H
    [00 52 00 2d 00 19 01 00 01 0d 03 00 a8 1a 00 33 01 03
   00 33
     00 3e 00 07 27 7e 00 03 09 00 00 27 4e 00 02 00 05 27
     92 00 04 87 77 37 b7 27 93 00 04 00 00 38 00 27 18 00
     0a 04 00 00 00 0a 50 88 62 34 56 27 c1 00 02 00 3c 27
     d8 00 07 4c 75 63 65 6e 74 00]

Park Channel
H->X
    [00 11 00 bf 00 01 01 00 02 0d 03 00 a8 1a 0d 03 00 a8
     1a]

X->H
    [00 07 00 bf 00 01 01 00 10]


Send Alerting
----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = AD71
      H225:  1... ....  .... .... = Message to originator
     H225:  .010 1101  0111 0001 = Call reference value =
   11633
      H225: Message type = 1 (Alerting)
      H225:
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 92
```

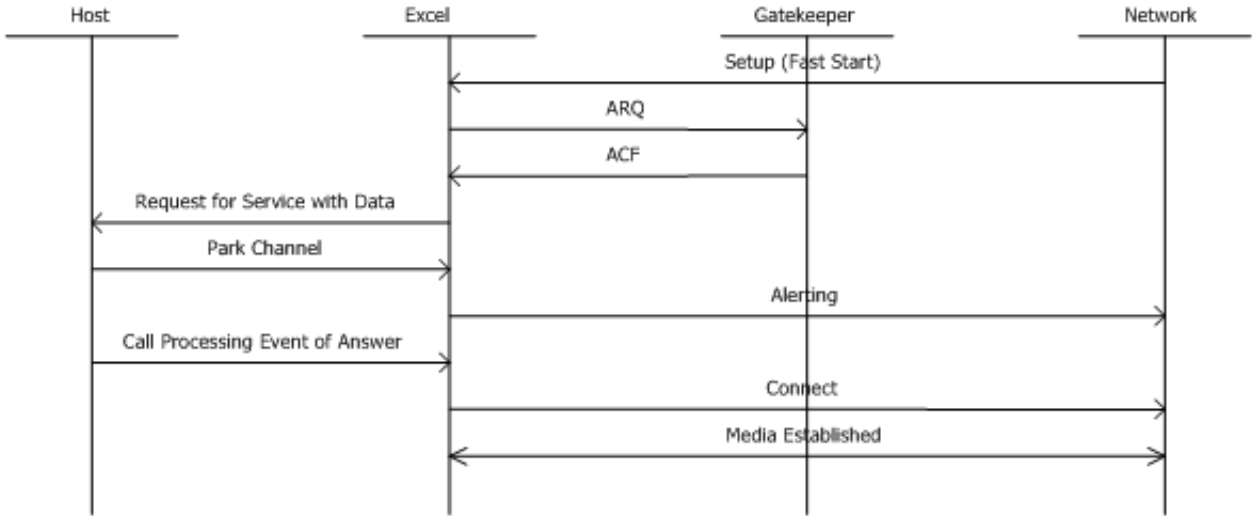
```
   H225: Discriminator = 5 (X.208/X.209 (ASN.1))
   H225:
   H225: Flags     = 03
   H225: 0... .... = No extension values present in
H323-UserInformation
   H225: .0.. .... = user-data is not present
   H225: ..0. .... = No extension values present in
h323-uu-pdu
   H225: ...0 .... = nonStandardData is not present
   H225: .... 0... = No extension values present in
h323-message-body
   H225: h323-message = 3 (Alerting)
   H225: Flags     = C0
   H225: 1... .... = Extension values present in
Alerting-UUIE
   H225: .1.. .... = h245Address is present
   H225: Protocol Id = {0.0.8.2250.0.2}
   H225:
   H225: Destination information
   H225: Flags     = 02
   H225: 0... .... = No extension value(s) present in
destinationInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = Vendor is not present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
terminal
   H225: Flags     = 00
   H225: 0... .... = nonStandardData is not present
   H225: .0.. .... = MC = 0
   H225: ..0. .... = Undefined node = 0
   H225:
   H225: ...0 .... = No extension value(s) present in
h245Address
   H225: h245Address = 0 (IP address)
   H225: IP   = 135.119.55.181
   H225: Port = 53503
   H225:
   H225: Flags     = 09
   H225: 0... .... = Extension length determinant
   H225: Number of extension = 5
   H225: .... ...1 = callIdentifier is present
   H225: Flags     = 10
   H225: 0... .... = h245SecurityMode is not present
   H225: .0.. .... = tokens is not present
   H225: ..0. .... = cryptoTokens is not present
   H225: ...1 .... = fastStart is present
```

```
      H225: Length of callIdentifier = 17
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   callIdentifier
      H225: Guid = 3FB87FC806EA18109CD90050DAD08480
      H225: Number of fastStart = 2
 FastStartElement sent
40 06 93 06 04 01 00 4c 60 1d 80 0b 05 00 01 00 87 77 37
   b7 38 01 00
FastStartElement sent
00 00 67 0c 60 1d 80 12 15 00 01 00 87 77 37 b7 38 00 00
   87 77 37 b7 38 01 80

Call processing Event of Answer
H->X
    [00 0d 00 ba 00 02 01 00 01 0d 03 00 a8 1a 01]

X->H
    [00 07 00 ba 00 02 01 00 10]

Send Connect
----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = AD71
      H225:  1... ....  .... .... = Message to originator
     H225:  .010 1101  0111 0001 = Call reference value =
   11633
      H225: Message type = 7 (Connect)
      H225:
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 109
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 02
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..0. .... = No extension values present in
   h323-uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 2 (Connect)
```

```
   H225: Flags     = C0
   H225: 1... .... = Extension values present in
Connect-UUIE
   H225: .1.. .... = h245Address is present
   H225: Protocol Id = {0.0.8.2250.0.2}
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
h245Address
   H225: h245Address = 0 (IP address)
   H225: IP   = 135.119.55.181
   H225: Port = 53503
   H225:
   H225: Destination information
   H225: Flags     = 02
   H225: 0... .... = No extension value(s) present in
destinationInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = Vendor is not present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
terminal
   H225: Flags     = 00
   H225: 0... .... = nonStandardData is not present
   H225: .0.. .... = MC = 0
   H225: ..0. .... = Undefined node = 0
   H225: Conference id =
3FB87FC806EA18109CD80050DAD08480
   H225:
   H225: Flags     = 09
   H225: 0... .... = Extension length determinant
   H225: Number of extension = 5
   H225: .... ...1 = callIdentifier is present
   H225: Flags     = 10
   H225: 0... .... = h245SecurityMode is not present
   H225: .0.. .... = tokens is not present
   H225: ..0. .... = cryptoTokens is not present
   H225: ...1 .... = fastStart is present
   H225: Length of callIdentifier = 17
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
callIdentifier
   H225: Guid = 3FB87FC806EA18109CD90050DAD08480
   H225: Number of fastStart = 2
   FastStartElement sent
```

```
40 06 93 06 04 01 00 4c 60 1d 80 0b 05 00 01 00 87 77 37
   b7 38 01 00
FastStartElement sent
00 00 67 0c 60 1d 80 12 15 00 01 00 87 77 37 b7 38 00 00
   87 77 37 b7 38 01 80
```

H.323 Incoming Call - Fast Connect, Gatekeeper



Message Trace

```
Receive Setup
----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 2D74
      H225:  0... ....  .... .... = Message from
   originator
     H225:  .010 1101  0111 0100 = Call reference value =
   11636
      H225: Message type = 5 (Setup)
      H225:
      H225: Bearer Capacity
      H225: Information element identifier = 4
      H225: Length of bearer capacity = 3
      H225: Coding and capability flags = 88
      H225:                    1... .... = Expected
   extension bit
      H225:                   .00. .... = Coding standard
   = 0 (CCITT standardized coding)
```

```
   H225:            0 1000 = Information transfer
capability = 8 (Unrestricted digital
                           information)
   H225: Mode and rate flags = C0
   H225:           1... .... = Expected extension bit
   H225:          .10. .... = Transfer mode = 2 (Packet
mode)
   H225:          ...0 0000 = Information transfer rate
= 0 (Used for packet mode calls)
   H225: Layer 1 protocol flag = A5
   H225:             1... .... = Expected extension bit
   H225:            .01. .... = Layer 1 identification
= 1
   H225:             ...0 0101 = Layer 1 protocol = 5
(H.221 and H.242)
   H225:
   H225: Display
   H225: Information element identifier = 40
   H225: Display length = 7
   H225: Display flag = 4C
   H225:    0... .... = Expected extension bit
   H225: Display = "Excel"
   H225:
   H225: Calling Party Number
   H225: Information element identifier  = 108
   H225: Calling party number contents length = 11
   H225: Type and numbering flags = FF
   H225:               1... .... = Expected extension
bit
   H225:               .111 .... = Type of number = 7
(Reserved for extension)
   H225:               .... 1111 = Numbering plan ID =
15 (Reserved for extension)
   H225: Number digits flag = 35
   H225:          0... .... = Expected extension bit
   H225: Number digits = "5088623456"
   H225:
   H225: Called Party Number
   H225: Information element identifier  = 112
   H225: Called party number contents length = 11
   H225: Type and numbering flags = 81
   H225:               1... .... = Expected extension
bit
   H225:               .000 .... = Type of number = 0
(Unknown)
   H225:               .... 0001 = Numbering plan ID =
1 (ISDN/telephony (E.164))
   H225: Call number flag = 35
   H225:        0... .... = Expected extension bit
   H225: Call number  = "5088622222"
   H225:
```

```
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 587
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 20
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..1. .... = Extension values present in h323-
   uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 0 (Setup)
      H225:
      H225: Flags     = B8
      H225: 1... .... = Extension value(s) present in
   Setup
      H225: .0.. .... = h245Address is not present
      H225: ..1. .... = sourceAddress is present
      H225: ...1 .... = destinationAddress is present
      H225: .... 1... = destCallSignalAddress is present
      H225: .... .0.. = destExtraCallInfo is not present
      H225: .... ..0. = destExtraCRV is not present
      H225: .... ...0 = callServices is not present
      H225:
      H225: Protocol id = {0.0.8.2250.0.2}
      H225:
      H225: Number of sourceAddress = 2
      H225:
      H225: Flags     = 40
      H225: 0... .... = No extension value(s) present in
   sourceAddress
      H225: sourceAddress = 1 (H323-ID)
      H225: H323-ID = "Excel"
      H225:
      H225: Flags     = 04
      H225: 0... .... = No extension value(s) present in
   sourceAddress
      H225: sourceAddress = 0 (E164)
      H225: Length of e164 = 10
      H225: E.164 = 5088623456
      H225:
      H225: Flags     = 22
      H225: 0... .... = No extension value(s) present in
   sourceInfo
      H225: .0.. .... = nonStandardData is not present
```

```
   H225: ..1. .... = Vendor is present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
vendor
   H225: Flags     = C0
   H225: 1... .... = Product ID is present
   H225: .1.. .... = Version ID is present
   H225:
   H225: ..0. .... = No extension value(s) present in
vendor
   H225: T.35 country code = 0x 9 (Australia)
   H225: T.35 extension    = 0
   H225: Manufacture code = 0x  3D (?)
   H225: Product Id = Equivalence OpenPhone<0000>
   H225: Version Id = 1.2.0<0000>
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
terminal
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = MC = 0
   H225: ...0 .... = Undefined node = 0
   H225:
   H225: Number of destinationAddress = 1
   H225:
   H225: Flags     = 04
   H225: 0... .... = No extension value(s) present in
destinationAddress
   H225: destinationAddress = 0 (E164)
   H225: Length of e164 = 10
   H225: E.164 = 5088622222
   H225:
   H225: destCallSignalAddress
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
destCallSignalAddress
   H225: destCallSignalAddress = 0 (IP address)
   H225: IP   = 135.119.55.181
   H225: Port = 1720
   H225:
   H225: Flags     = 00
   H225: 0... .... = Active MC = 0
   H225: Conference id =
E978D4CF06EA18109CDB0050DAD08480
   H225: Flags     = 00
```

```
      H225: 0... .... = No extension value(s) present in
   conferenceGoal
      H225: Conference goal = 0 (Create)
      H225:
      H225: callType
      H225: ...0 .... = No extension value(s) present in
   callType
      H225: callType = 0 (Point to point)
      H225:
      H225: .... ..0. = Extension length determinant
      H225: Number of extension = 12
     H225: .... .1.. = sourceCallSignalAddress is present
NUMBER OF OLC'S IN THE FS PROPOSAL = 12
FastStartElement Received
40 00 00 06 04 01 00 4c 10 09 00 00 3d 06 4c 50 43 2d 31
   30 80 11 1c 00 01 00 87
 77 37 aa
13 88 00 87 77 37 aa 13 89

FastStartElement Received
00 00 64 0c 10 09 00 00 3d 06 4c 50 43 2d 31 30 80 0b 0d
   00 01 00 87 77 37 aa 13
 89 80
FastStartElement Received
40 00 00 06 04 01 00 4e 0c 03 00 83 00 80 11 1c 00 01 00
   87 77 37 aa 13 88 00 87
 77 37 aa
13 89

FastStartElement Received
00 00 65 0e 0c 03 00 83 00 80 0b 0d 00 01 00 87 77 37 aa
   13 89 80

FastStartElement Received
40 00 00 06 04 01 00 4c 10 b5 00 53 4c 2a 02 00 00 00 00
   00 40 01 00 00 40 01 02
 00 08 00
00 00 00 00 31 00 01 00 40 1f 00 00 59 06 00 00 41 00 00
   00 02 00 40 01 00 00 80
 12 1c 40
01 00 87 77 37 aa 13 88 00 87 77 37 aa 13 89 00

FastStartElement Received
00 00 66 0c 10 b5 00 53 4c 2a 02 00 00 00 00 00 40 01 00
   00 40 01 02 00 08 00 00
 00 00 00
31 00 01 00 40 1f 00 00 59 06 00 00 41 00 00 00 02 00 40
   01 00 00 80 0b 0d 40 01
 00 87 77
37 aa 13 89 80
```

```
FastStartElement Received
00 00 67 0c 60 1d 80 0b 0d 00 01 00 87 77 37 aa 13 89 80
FastStartElement Received
40 00 00 06 04 01 00 4c 20 1d 80 11 1c 00 01 00 87 77 37
   aa 13 88 00 87 77 37 aa
 13 89
FastStartElement Received
00 00 68 0c 20 1d 80 0b 0d 00 01 00 87 77 37 aa 13 89 80
FastStartElement Received
40 00 00 06 04 01 00 48 71 03 51 00 80 01 00 80 11 1c 00
   02 00 87 77 37 aa 13 8a
 00 87 77
37 aa 13 8b
FastStartElement Received
40 00 00 06 04 01 00 48 6b 03 51 00 80 01 00 80 11 1c 00
   02 00 87 77 37 aa 13 8a
 00 87 77
37 aa 13 8b

Send ARQ
----- H.225 Gatekeeper Registration, Admission and Status
   -----
      RAS:
      RAS: Flags     = 26
      RAS: 0... .... = No extension value(s) present in
   RasMessage
      RAS: RAS message = 9 (Admission Request)
      RAS: .... ..1. = Extension value(s) present in
   admissionRequest
      RAS: .... ...0 = callModel is not present
      RAS: Flags     = 80
      RAS: 1... .... = destinationInfo is present
      RAS: .0.. .... = destCallSignalAddress is not
   present
      RAS: ..0. .... = destExtraCallInfo is not present
      RAS: ...0 .... = srcCallSignalAddress is not present
      RAS: .... 0... = nonStandardData is not present
      RAS: .... .0.. = callServices is not present
      RAS: Request sequence number = 8
      RAS: Flags     = 02
      RAS: 0... .... = No extension value(s) present in
   callType
      RAS: callType = 0 (Point to point)
      RAS: Endpoint ID = "Excel-CSP"
      RAS: Number of destinationInfo = 1
      RAS:
      RAS: Flags     = 04
      RAS: 0... .... = No extension value(s) present in
   destinationInfo
      RAS: destinationInfo = 0 (E164)
      RAS: Length of e164 = 10
```

```
      RAS: E.164 = 5088622222
      RAS: Number of srcInfo = 2
      RAS:
      RAS: Flags     = 04
      RAS: 0... .... = No extension value(s) present in
   srcInfo
      RAS: srcInfo = 0 (E164)
      RAS: Length of e164 = 10
      RAS: E.164 = 5088623456
      RAS:
      RAS: Flags     = 40
      RAS: 0... .... = No extension value(s) present in
   srcInfo
      RAS: srcInfo = 1 (H323-ID)
      RAS: H323-ID = "Excel"
      RAS: Band width  = 1280
      RAS: Call reference value = 44404
      RAS: Conference id =
   E978D4CF06EA18109CDB0050DAD08480
      RAS: Flags     = 44
      RAS: 0... .... = Active MC = 0
      RAS: .1.. .... = Answer call = 1
      RAS:
      RAS: ..0. .... = Extension index format
      RAS: Number of extension = 10
      RAS: .1.. .... = canMapAlias is present
      RAS: ..1. .... = callIdentifier is present
      RAS: ...0 .... = srcAlternatives is not present
      RAS: .... 0... = destAlternatives is not present
      RAS: .... .0.. = gatekeeperIdentifier is not present
      RAS: .... ..0. = tokens is not present
      RAS: .... ...0 = cryptoTokens is not present
      RAS: Flags     = 20
      RAS: 0... .... = integrityCheckValue is not present
      RAS: .0.. .... = transportQOS is not present
      RAS: ..1. .... = willSupplyUUIEs is present
      RAS: Flags     = 80
      RAS: 1... .... = canMapAlias = 1
      RAS:
      RAS: Flags     = 00
      RAS: 0... .... = No extension value(s) present in
   callIdentifier
      RAS: Guid = E978D4CF06EA18109CDC0050DAD08480
      RAS: Flags     = 00
      RAS: 0... .... = Will supply UUIEs = 0

Receive ACF
----- H.225 Gatekeeper Registration, Admission and Status
   -----
      RAS:
```

```
   RAS: Flags     = 2A
   RAS: 0... .... = No extension value(s) present in
RasMessage
   RAS: RAS message = 10 (Admission Confirm)
   RAS: .... ..1. = Extension value(s) present in
admissionConfirm
   RAS: .... ...0 = irrFrequency is not present
   RAS: Flags     = 80
   RAS: 1... .... = nonStandardData is present
   RAS: Request sequence number = 8
   RAS: Band width  = 1280
   RAS:
   RAS: Flags     = 00
   RAS: 0... .... = No extension value(s) present in
callModel
   RAS: Call model = 0 (Direct)
   RAS:
   RAS: ..0. .... = No extension value(s) present in
destCallSignalAddress
   RAS: destCallSignalAddress = 0 (IP address)
   RAS: IP   = 0.0.0.0
   RAS: Port = 1720
   RAS:
   RAS: Flags     = 40
   RAS: 0... .... = No extension value(s) present in
NonStandardIdentifier
   RAS: nonStandardIdentifier = 1 (H221 non standard)
   RAS:
   RAS: ..0. .... = No extension value(s) present in
h221NonStandard
   RAS: T.35 country code = 0xB5 (USA)
   RAS: T.35 extension    = 0
   RAS: Manufacture code = 0x  14 (Ascend)
   RAS: Length of data = 10
   RAS: Data: 10 byte(s) of data
   RAS:
   RAS: Flags     = 15
   RAS: 0... .... = Extension index format
   RAS: Number of extension = 11
   RAS: .... ...1 = destinationInfo is present
   RAS: Flags     = 00
   RAS: 0... .... = destExtraCallInfo is not present
   RAS: .0.. .... = destinationType is not present
   RAS: ..0. .... = remoteExtensionAddress is not
present
   RAS: ...0 .... = alternateEndpoints is not present
   RAS: .... 0... = tokens is not present
   RAS: .... .0.. = cryptoTokens is not present
   RAS: .... ..0. = integrityCheckValue is not present
   RAS: .... ...0 = transportQOS is not present
   RAS: Flags     = C0
```

```
      RAS: 1... .... = willRespondToIRR is present
      RAS: .1.. .... = uuiesRequested is present
      RAS: Number of destinationInfo = 1
      RAS:
      RAS: Flags     = 04
      RAS: 0... .... = No extension value(s) present in
   destinationInfo
      RAS: destinationInfo = 0 (E164)
      RAS: Length of e164 = 10
      RAS: E.164 = 5088622222
      RAS: Flags     = 00
      RAS: 0... .... = willRespondToIRR = 0
      RAS:
      RAS: Flags     = 00
      RAS: 0... .... = No extension value(s) present in
   uuiesRequested
      RAS: .0.. .... = Setup = 0
      RAS: ..0. .... = Call proceeding = 0
      RAS: ...0 .... = Connect = 0
      RAS: .... 0... = Alerting = 0
      RAS: .... .0.. = User Information = 0
      RAS: .... ..0. = Release complete = 0
      RAS: .... ...0 = Facility = 0
      RAS: Flags     = 00
      RAS: 0... .... = Progress = 0
      RAS: .0.. .... = Empty = 0

Request for Service with data
X->H
    [00 66 00 2d 00 1d 01 00 01 0d 03 00 a8 1d 00 33 01 03
   00 33
     00 52 00 09 27 7e 00 03 09 00 00 27 4e 00 02 00 05 27
     92 00 04 87 77 37 b7 27 93 00 04 00 00 38 e0 27 18 00
     0a 04 00 00 00 0a 50 88 62 34 56 27 17 00 08 04 00 0a
     50 88 62 22 22 27 c1 00 02 00 3c 27 d8 00 07 4c 75 63
     65 6e 74 00 27 c3 00 04 00 00 06 b8]


Park Channel
H->X
    [00 11 00 bf 00 01 01 00 02 0d 03 00 a8 1d 0d 03 00 a8
     1d]

X->H
    [00 07 00 bf 00 01 01 00 10]

Send Alerting
----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
```

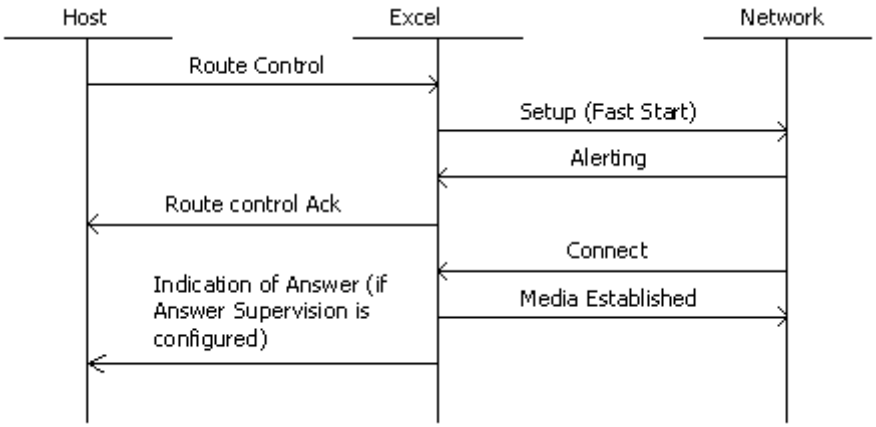
```
      H225: Length of call reference = 2
      H225: Call reference field = AD74
      H225:  1... ....  .... .... = Message to originator
     H225:  .010 1101  0111 0100 = Call reference value =
   11636
      H225: Message type = 1 (Alerting)
      H225:
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 92
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 03
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..0. .... = No extension values present in
   h323-uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 3 (Alerting)
      H225: Flags     = C0
      H225: 1... .... = Extension values present in
   Alerting-UUIE
      H225: .1.. .... = h245Address is present
      H225: Protocol Id = {0.0.8.2250.0.2}
      H225:
      H225: Destination information
      H225: Flags     = 02
      H225: 0... .... = No extension value(s) present in
   destinationInfo
      H225: .0.. .... = nonStandardData is not present
      H225: ..0. .... = Vendor is not present
      H225: ...0 .... = Gatekeeper is not present
      H225: .... 0... = Gateway is not present
      H225: .... .0.. = MCU is not present
      H225: .... ..1. = Terminal is present
      H225:
      H225: .... ...0 = No extension value(s) present in
   terminal
      H225: Flags     = 00
      H225: 0... .... = nonStandardData is not present
      H225: .0.. .... = MC = 0
      H225: ..0. .... = Undefined node = 0
      H225:
```

```
      H225: ...0 .... = No extension value(s) present in
   h245Address
      H225: h245Address = 0 (IP address)
      H225: IP   = 135.119.55.181
      H225: Port = 53506
      H225:
      H225: Flags     = 09
      H225: 0... .... = Extension length determinant
      H225: Number of extension = 5
      H225: .... ...1 = callIdentifier is present
      H225: Flags     = 10
      H225: 0... .... = h245SecurityMode is not present
      H225: .0.. .... = tokens is not present
      H225: ..0. .... = cryptoTokens is not present
      H225: ...1 .... = fastStart is present
      H225: Length of callIdentifier = 17
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   callIdentifier
      H225: Guid = E978D4CF06EA18109CDC0050DAD08480
      H225: Number of fastStart = 2
FastStartElement sent
40 06 cb 06 04 01 00 4c 60 1d 80 0b 05 00 01 00 87 77 37
   b7 38 e1 00
FastStartElement sent
00 00 67 0c 60 1d 80 12 15 00 01 00 87 77 37 b7 38 e0 00
   87 77 37 b7 38 e1 80

Call Processing Event of Answer
H->X
    [00 0d 00 ba 00 02 01 00 01 0d 03 00 a8 1d 01]

X->H
    [00 07 00 ba 00 02 01 00 10]

Send Connect
----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = AD74
      H225:  1... ....  .... .... = Message to originator
     H225:  .010 1101  0111 0100 = Call reference value =
   11636
      H225: Message type = 7 (Connect)
      H225:
      H225:
H225:
      H225: ----User-User Information----
```

```
   H225:
   H225: Information element identifier = 126 (User-
User)
   H225: Length        = 109
   H225: Discriminator = 5 (X.208/X.209 (ASN.1))
   H225:
   H225: Flags     = 02
   H225: 0... .... = No extension values present in
H323-UserInformation
   H225: .0.. .... = user-data is not present
   H225: ..0. .... = No extension values present in
h323-uu-pdu
   H225: ...0 .... = nonStandardData is not present
   H225: .... 0... = No extension values present in
h323-message-body
   H225: h323-message = 2 (Connect)
   H225: Flags     = C0
   H225: 1... .... = Extension values present in
Connect-UUIE
   H225: .1.. .... = h245Address is present
   H225: Protocol Id = {0.0.8.2250.0.2}
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
h245Address
   H225: h245Address = 0 (IP address)
   H225: IP   = 135.119.55.181
   H225: Port = 53506
   H225:
   H225: Destination information
   H225: Flags     = 02
   H225: 0... .... = No extension value(s) present in
destinationInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = Vendor is not present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
terminal
   H225: Flags     = 00
   H225: 0... .... = nonStandardData is not present
   H225: .0.. .... = MC = 0
   H225: ..0. .... = Undefined node = 0
   H225: Conference id =
E978D4CF06EA18109CDB0050DAD08480
   H225:
   H225: Flags     = 09
   H225: 0... .... = Extension length determinant
```

```
      H225: Number of extension = 5
      H225: .... ...1 = callIdentifier is present
      H225: Flags     = 10
      H225: 0... .... = h245SecurityMode is not present
      H225: .0.. .... = tokens is not present
      H225: ..0. .... = cryptoTokens is not present
      H225: ...1 .... = fastStart is present
      H225: Length of callIdentifier = 17
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   callIdentifier
      H225: Guid = E978D4CF06EA18109CDC0050DAD08480
      H225: Number of fastStart = 2
FastStartElement sent
40 06 cb 06 04 01 00 4c 60 1d 80 0b 05 00 01 00 87 77 37
   b7 38 e1 00
FastStartElement sent
00 00 67 0c 60 1d 80 12 15 00 01 00 87 77 37 b7 38 e0 00
   87 77 37 b7 38 e1 80
```

H.323 Outgoing Call - Fast Connect, No Gatekeeper



Message Trace

```
Route Control
H->X
    [00 5e 00 e8 00 01 01 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 2f 00 06 27 7e 00
     01 09 27 c1 00 02 00 3c 27 c2 00 04 87 77 37 aa 27 17
     00 05 10 00 04 30 95 27 18 00 07 10 00 04 0a 04 34 50
     27 b0 00 02 00 01]

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 06B0
      H225:  0... ....  .... .... = Message from
   originator
     H225:  .000 0110  1011 0000 = Call reference value =
   1712
      H225: Message type = 5 (Setup)
      H225:
      H225: Bearer Capacity
      H225: Information element identifier = 4
      H225: Length of bearer capacity = 3
      H225: Coding and capability flags = 88
```

```
   H225:                    1... .... = Expected
extension bit
   H225: .00. .... = Coding standard = 0 (CCITT
standardized coding)
   H225:                    ...0 1000 = Information
transfer capability = 8 (Unrestricted digital
information)
   H225: Mode and rate flags = D0
   H225:           1... .... = Expected extension bit
   H225:          .10. .... = Transfer mode = 2 (Packet
mode)
   H225:          ...1 0000 = Information transfer rate
= 16 (64 kbit/s)
   H225: Layer 1 protocol flag = A5
   H225:             1... .... = Expected extension bit
   H225:            .01. .... = Layer 1 identification
= 1
   H225:             ...0 0101 = Layer 1 protocol = 5
(H.221 and H.242)
   H225:
   H225: Display
   H225: Information element identifier = 40
   H225: Display length = 6
   H225: Display flag = 4C
   H225:    0... .... = Expected extension bit
   H225: Display = "Excel"
   H225:
   H225: Calling Party Number
   H225: Information element identifier  = 108
   H225: Calling party number contents length = 5
   H225: Type and numbering flags = 89
   H225:                1... .... = Expected extension
bit
   H225:                .000 .... = Type of number = 0
(Unknown)
   H225:               .... 1001 = Numbering plan ID =
9 (Private)
   H225: Number digits flag = 33
   H225:           0... .... = Expected extension bit
   H225: Number digits = "3450"
   H225:
   H225: Called Party Number
   H225: Information element identifier  = 112
   H225: Called party number contents length = 5
   H225: Type and numbering flags = 89
   H225:                1... .... = Expected extension
bit
   H225:                .000 .... = Type of number = 0
(Unknown)
   H225:               .... 1001 = Numbering plan ID =
9 (Private)
   H225: Call number flag = 33
```

```
      H225:        0... .... = Expected extension bit
      H225: Call number  = "3095"
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 142
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..0. .... = No extension values present in
   h323-uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 0 (Setup)
      H225:
      H225: Flags     = F0
      H225: 1... .... = Extension value(s) present in
   Setup
      H225: .1.. .... = h245Address is present
      H225: ..1. .... = sourceAddress is present
      H225: ...1 .... = destinationAddress is present
      H225: .... 0... = destCallSignalAddress is not
   present
      H225: .... .0.. = destExtraCallInfo is not present
      H225: .... ..0. = destExtraCRV is not present
      H225: .... ...0 = callServices is not present
      H225:
      H225: Protocol id = {0.0.8.2250.0.2}
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   h245Address
      H225: h245Address = 0 (IP address)
      H225: IP   = 135.119.55.181
      H225: Port = 53510
      H225:
      H225: Number of sourceAddress = 1
      H225:
      H225: Flags     = 01
      H225: 0... .... = No extension value(s) present in
   sourceAddress
      H225: sourceAddress = 0 (E164)
      H225: Length of e164 = 4
      H225: E.164 = 3450
```

```
      H225:
      H225: Flags     = 02
      H225: 0... .... = No extension value(s) present in
   sourceInfo
      H225: .0.. .... = nonStandardData is not present
      H225: ..0. .... = Vendor is not present
      H225: ...0 .... = Gatekeeper is not present
      H225: .... 0... = Gateway is not present
      H225: .... .0.. = MCU is not present
      H225: .... ..1. = Terminal is present
      H225:
      H225: .... ...0 = No extension value(s) present in
   terminal
      H225: Flags     = 00
      H225: 0... .... = nonStandardData is not present
      H225: .0.. .... = MC = 0
      H225: ..0. .... = Undefined node = 0
      H225:
      H225: Number of destinationAddress = 2
      H225:
      H225: Flags     = 01
      H225: 0... .... = No extension value(s) present in
   destinationAddress
      H225: destinationAddress = 0 (E164)
      H225: Length of e164 = 4
      H225: E.164 = 3095
      H225:
      H225: Flags     = 81
      H225: 1... .... = Extension value(s) present in
   destinationAddress
      H225: destinationAddress = 0 (URL-ID)
      H225: URL-ID =
FastStartElement sent
00 07 13 0c 20 13 80 0b 05 00 01 00 87 77 37 b7 3a 01 00
FastStartElement sent
40 ff fe 06 04 01 00 4c 20 13 80 12 15 00 01 00 87 77 37
   b7 3a 00 00 87 77 37 b7
 3a 01 00

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 86B0
      H225:  1... ....  .... .... = Message to originator
     H225:  .000 0110  1011 0000 = Call reference value =
   1712
      H225: Message type = 1 (Alerting)
      H225:
      H225: Display
      H225: Information element identifier = 40
```

```
      H225: Display length = 7
      H225: Display flag = 4C
      H225:    0... .... = Expected extension bit
      H225: Display = "Excel"
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 77
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 23
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..1. .... = Extension values present in h323-
   uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 3 (Alerting)
      H225: Flags     = 80
      H225: 1... .... = Extension values present in
   Alerting-UUIE
      H225: .0.. .... = h245Address is not present
      H225: Protocol Id = {0.0.8.2250.0.2}
      H225:
      H225: Destination information
      H225: Flags     = 22
      H225: 0... .... = No extension value(s) present in
   destinationInfo
      H225: .0.. .... = nonStandardData is not present
      H225: ..1. .... = Vendor is present
      H225: ...0 .... = Gatekeeper is not present
      H225: .... 0... = Gateway is not present
      H225: .... .0.. = MCU is not present
      H225: .... ..1. = Terminal is present
      H225:
      H225: .... ...0 = No extension value(s) present in
   vendor
      H225: Flags     = C0
      H225: 1... .... = Product ID is present
      H225: .1.. .... = Version ID is present
      H225:
      H225: ..0. .... = No extension value(s) present in
   vendor
      H225: T.35 country code = 0x 9 (Australia)
      H225: T.35 extension    = 0
      H225: Manufacture code = 0x  3D (?)
```

```
      H225: Product Id = Equivalence OpenPhone<0000>
      H225: Version Id = 1.2.0<0000>
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   terminal
      H225: .0.. .... = nonStandardData is not present
      H225: ..0. .... = MC = 0
      H225: ...0 .... = Undefined node = 0
      H225:
      H225: .... 0... = Extension length determinant
      H225: Number of extension = 7
      H225: ...1 .... = callIdentifier is present
      H225: .... 0... = h245SecurityMode is not present
      H225: .... .0.. = tokens is not present
      H225: .... ..0. = cryptoTokens is not present
      H225: .... ...0 = fastStart is not present
      H225:
      H225: Flags     = C0
      H225: 1... .... = Extension value(s) present in
   callIdentifier
      H225: Guid = 1100000006B0000006B0616263643132
      H225:
      H225: Flags     = 33
      H225: 0... .... = Extension length determinant
      H225: Number of extension = 26
      H225: .... ...1 = h4501SupplementaryService is
   present
      H225: Flags     = 34
      H225: 0... .... = h245Tunneling is not present
      H225: .0.. .... = h245Control is not present
      H225: ..1. .... = nonStandardControl is present
      H225: Number of h4501SupplementaryService = 0
      H225: Flags     = 00
      H225: 0... .... = H245 tunneling = 0
      H225: Number of nonStandardControl = 1
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   NonStandardIdentifier
      H225: nonStandardIdentifier = 0 (Object)
      H225: Data: 0 byte(s) of data

Route Control Ack
X->H
    [00 14 00 e8 00 01 01 00 10 01 02 1e 09 00 01 00 39 00
   03 00
     a9 02]

Receive Connect
```

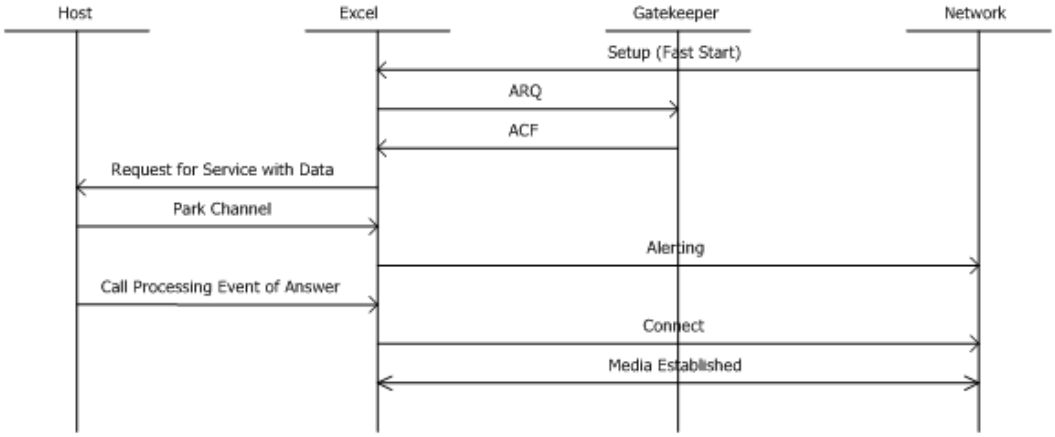
```
----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 86B0
      H225:  1... ....  .... .... = Message to originator
     H225:  .000 0110  1011 0000 = Call reference value =
   1712
      H225: Message type = 7 (Connect)
      H225:
      H225: Bearer Capacity
      H225: Information element identifier = 4
      H225: Length of bearer capacity = 3
      H225: Coding and capability flags = 88
      H225:                    1... .... = Expected
   extension bit
      H225:                   .00. .... = Coding standard
   = 0 (CCITT standardized coding)
      H225:                    ...0 1000 = Information
   transfer capability = 8 (Unrestricted digital
   information)
      H225: Mode and rate flags = C0
      H225:           1... .... = Expected extension bit
     H225:          .10. .... = Transfer mode = 2 (Packet
   mode)
     H225:          ...0 0000 = Information transfer rate
   = 0 (Used for packet mode calls)
      H225: Layer 1 protocol flag = A5
      H225:             1... .... = Expected extension bit
     H225:            .01. .... = Layer 1 identification
   = 1
      H225:              ...0 0101 = Layer 1 protocol = 5
   (H.221 and H.242)
      H225:
      H225: Display
      H225: Information element identifier = 40
      H225: Display length = 7
      H225: Display flag = 4C
      H225:    0... .... = Expected extension bit
      H225: Display = "Excel"
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 152
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 22
```

```
   H225: 0... .... = No extension values present in
H323-UserInformation
   H225: .0.. .... = user-data is not present
   H225: ..1. .... = Extension values present in h323-
uu-pdu
   H225: ...0 .... = nonStandardData is not present
   H225: .... 0... = No extension values present in
h323-message-body
   H225: h323-message = 2 (Connect)
   H225: Flags     = C0
   H225: 1... .... = Extension values present in
Connect-UUIE
   H225: .1.. .... = h245Address is present
   H225: Protocol Id = {0.0.8.2250.0.2}
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
h245Address
   H225: h245Address = 0 (IP address)
   H225: IP   = 135.119.55.170
   H225: Port = 2358
   H225:
   H225: Destination information
   H225: Flags     = 22
   H225: 0... .... = No extension value(s) present in
destinationInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..1. .... = Vendor is present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
vendor
   H225: Flags     = C0
   H225: 1... .... = Product ID is present
   H225: .1.. .... = Version ID is present
   H225:
   H225: ..0. .... = No extension value(s) present in
vendor
   H225: T.35 country code = 0x 9 (Australia)
   H225: T.35 extension    = 0
   H225: Manufacture code = 0x  3D (?)
   H225: Product Id = Equivalence OpenPhone<0000>
   H225: Version Id = 1.2.0<0000>
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
terminal
   H225: .0.. .... = nonStandardData is not present
```

```
      H225: ..0. .... = MC = 0
      H225: ...0 .... = Undefined node = 0
      H225: Conference id =
   000006B0000006B03133353761636466
      H225:
      H225: Flags     = 0D
      H225: 0... .... = Extension length determinant
      H225: Number of extension = 7
      H225: .... ...1 = callIdentifier is present
      H225: Flags     = 1C
      H225: 0... .... = h245SecurityMode is not present
      H225: .0.. .... = tokens is not present
      H225: ..0. .... = cryptoTokens is not present
      H225: ...1 .... = fastStart is present
      H225: Length of callIdentifier = 17
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   callIdentifier
      H225: Guid = 000006B0000006B06162636431323334
      H225: Number of fastStart = 2
      H225: Length of fastStart = 25
      H225: Fast start = <0007130C> <13>
      H225: Length of fastStart = 23
      H225: Fast start = @<00>d<06040100>L <13>
      H225:
      H225: Flags     = 01
      H225: 0... .... = Extension length determinant
      H225: Number of extension = 1
      H225: .... ...1 = h4501SupplementaryService is
   present
      H225: Flags     = 00
      H225: 0... .... = h245Tunneling is not present
      H225: .0.. .... = h245Control is not present
      H225: ..0. .... = nonStandardControl is not present
      H225: Number of h4501SupplementaryService = 0
      H225: Flags     = 80
      H225: 1... .... = H245 tunneling = 1
      H225:
FastStartElement Received
00 07 13 0c 20 13 80 11 1c 00 01 00 87 77 37 aa 13 88 00
   87 77 37 aa 13 89
FastStartElement Received
40 00 64 06 04 01 00 4c 20 13 80 0b 0d 00 01 00 87 77 37
   aa 13 89 80


Indication of Answer (if Answer Supervision is
   Configured)

X->H
```

```
[00 0d 00 2e 00 07 01 00 01 0d 03 00 a8 07 20]
```

H.323 Outgoing Call - Fast Connect, Gatekeeper



Message Trace

```
Route Control
H->X
    [00 5e 00 e8 00 01 01 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 2f 00 06 27 7e 00
     01 09 27 c1 00 02 00 3c 27 c2 00 04 87 77 37 aa 27 17
     00 05 10 00 04 30 95 27 18 00 07 10 00 04 0a 04 34 50
     27 b0 00 02 00 01]

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 06B0
      H225:  0... ....  .... .... = Message from
   originator
     H225:  .000 0110  1011 0000 = Call reference value =
   1712
      H225: Message type = 5 (Setup)
```

```
   H225:
   H225: Bearer Capacity
   H225: Information element identifier = 4
   H225: Length of bearer capacity = 3
   H225: Coding and capability flags = 88
   H225:                   1... .... = Expected
extension bit
   H225:                   .00. .... = Coding standard
= 0 (CCITT standardized coding)
   H225:                   ...0 1000 = Information
transfer capability = 8 (Unrestricted digital
information)
   H225: Mode and rate flags = D0
   H225:           1... .... = Expected extension bit
   H225:          .10. .... = Transfer mode = 2 (Packet
mode)
   H225:          ...1 0000 = Information transfer rate
= 16 (64 kbit/s)
   H225: Layer 1 protocol flag = A5
   H225:            1... .... = Expected extension bit
   H225:            .01. .... = Layer 1 identification
= 1
   H225:             ...0 0101 = Layer 1 protocol = 5
(H.221 and H.242)
   H225:
   H225: Display
   H225: Information element identifier = 40
   H225: Display length = 6
   H225: Display flag = 4C
   H225:    0... .... = Expected extension bit
   H225: Display = "Excel"
   H225:
   H225: Calling Party Number
   H225: Information element identifier  = 108
   H225: Calling party number contents length = 5
   H225: Type and numbering flags = 89
   H225:               1... .... = Expected extension
bit
   H225:               .000 .... = Type of number = 0
(Unknown)
   H225:               .... 1001 = Numbering plan ID =
9 (Private)
   H225: Number digits flag = 33
   H225:          0... .... = Expected extension bit
   H225: Number digits = "3450"
   H225:
   H225: Called Party Number
   H225: Information element identifier  = 112
   H225: Called party number contents length = 5
   H225: Type and numbering flags = 89
```

```
      H225:                1... .... = Expected extension
   bit
      H225:                .000 .... = Type of number = 0
   (Unknown)
      H225:                .... 1001 = Numbering plan ID =
   9 (Private)
      H225: Call number flag = 33
      H225:        0... .... = Expected extension bit
      H225: Call number  = "3095"
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 142
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..0. .... = No extension values present in
   h323-uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 0 (Setup)
      H225:
      H225: Flags     = F0
      H225: 1... .... = Extension value(s) present in
   Setup
      H225: .1.. .... = h245Address is present
      H225: ..1. .... = sourceAddress is present
      H225: ...1 .... = destinationAddress is present
      H225: .... 0... = destCallSignalAddress is not
   present
      H225: .... .0.. = destExtraCallInfo is not present
      H225: .... ..0. = destExtraCRV is not present
      H225: .... ...0 = callServices is not present
      H225:
      H225: Protocol id = {0.0.8.2250.0.2}
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   h245Address
      H225: h245Address = 0 (IP address)
      H225: IP   = 135.119.55.181
      H225: Port = 53510
      H225:
      H225: Number of sourceAddress = 1
```

```
      H225:
      H225: Flags     = 01
      H225: 0... .... = No extension value(s) present in
   sourceAddress
      H225: sourceAddress = 0 (E164)
      H225: Length of e164 = 4
      H225: E.164 = 3450
      H225:
      H225: Flags     = 02
      H225: 0... .... = No extension value(s) present in
   sourceInfo
      H225: .0.. .... = nonStandardData is not present
      H225: ..0. .... = Vendor is not present
      H225: ...0 .... = Gatekeeper is not present
      H225: .... 0... = Gateway is not present
      H225: .... .0.. = MCU is not present
      H225: .... ..1. = Terminal is present
      H225:
      H225: .... ...0 = No extension value(s) present in
   terminal
      H225: Flags     = 00
      H225: 0... .... = nonStandardData is not present
      H225: .0.. .... = MC = 0
      H225: ..0. .... = Undefined node = 0
      H225:
      H225: Number of destinationAddress = 2
      H225:
      H225: Flags     = 01
      H225: 0... .... = No extension value(s) present in
   destinationAddress
      H225: destinationAddress = 0 (E164)
      H225: Length of e164 = 4
      H225: E.164 = 3095
      H225:
      H225: Flags     = 81
      H225: 1... .... = Extension value(s) present in
   destinationAddress
      H225: destinationAddress = 0 (URL-ID)
      H225: URL-ID =
FastStartElement sent
00 07 13 0c 20 13 80 0b 05 00 01 00 87 77 37 b7 3a 01 00
FastStartElement sent
40 ff fe 06 04 01 00 4c 20 13 80 12 15 00 01 00 87 77 37
   b7 3a 00 00 87 77 37 b7
 3a 01 00

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 86B0
```

```
      H225:  1... ....  .... .... = Message to originator
     H225:  .000 0110  1011 0000 = Call reference value =
   1712
      H225: Message type = 1 (Alerting)
      H225:
      H225: Display
      H225: Information element identifier = 40
      H225: Display length = 7
      H225: Display flag = 4C
      H225:    0... .... = Expected extension bit
      H225: Display = "Excel"
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 77
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 23
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..1. .... = Extension values present in h323-
   uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 3 (Alerting)
      H225: Flags     = 80
      H225: 1... .... = Extension values present in
   Alerting-UUIE
      H225: .0.. .... = h245Address is not present
      H225: Protocol Id = {0.0.8.2250.0.2}
      H225:
      H225: Destination information
      H225: Flags     = 22
      H225: 0... .... = No extension value(s) present in
   destinationInfo
      H225: .0.. .... = nonStandardData is not present
      H225: ..1. .... = Vendor is present
      H225: ...0 .... = Gatekeeper is not present
      H225: .... 0... = Gateway is not present
      H225: .... .0.. = MCU is not present
      H225: .... ..1. = Terminal is present
      H225:
      H225: .... ...0 = No extension value(s) present in
   vendor
      H225: Flags     = C0
      H225: 1... .... = Product ID is present
```

```
   H225: .1.. .... = Version ID is present
   H225:
   H225: ..0. .... = No extension value(s) present in
vendor
   H225: T.35 country code = 0x 9 (Australia)
   H225: T.35 extension    = 0
   H225: Manufacture code = 0x  3D (?)
   H225: Product Id = Equivalence OpenPhone<0000>
   H225: Version Id = 1.2.0<0000>
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
terminal
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = MC = 0
   H225: ...0 .... = Undefined node = 0
   H225:
   H225: .... 0... = Extension length determinant
   H225: Number of extension = 7
   H225: ...1 .... = callIdentifier is present
   H225: .... 0... = h245SecurityMode is not present
   H225: .... .0.. = tokens is not present
   H225: .... ..0. = cryptoTokens is not present
   H225: .... ...0 = fastStart is not present
   H225:
   H225: Flags     = C0
   H225: 1... .... = Extension value(s) present in
callIdentifier
   H225: Guid = 1100000006B0000006B0616263643132
   H225:
   H225: Flags     = 33
   H225: 0... .... = Extension length determinant
   H225: Number of extension = 26
   H225: .... ...1 = h4501SupplementaryService is
present
   H225: Flags     = 34
   H225: 0... .... = h245Tunneling is not present
   H225: .0.. .... = h245Control is not present
   H225: ..1. .... = nonStandardControl is present
   H225: Number of h4501SupplementaryService = 0
   H225: Flags     = 00
   H225: 0... .... = H245 tunneling = 0
   H225: Number of nonStandardControl = 1
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
NonStandardIdentifier
   H225: nonStandardIdentifier = 0 (Object)
   H225: Data: 0 byte(s) of data
```

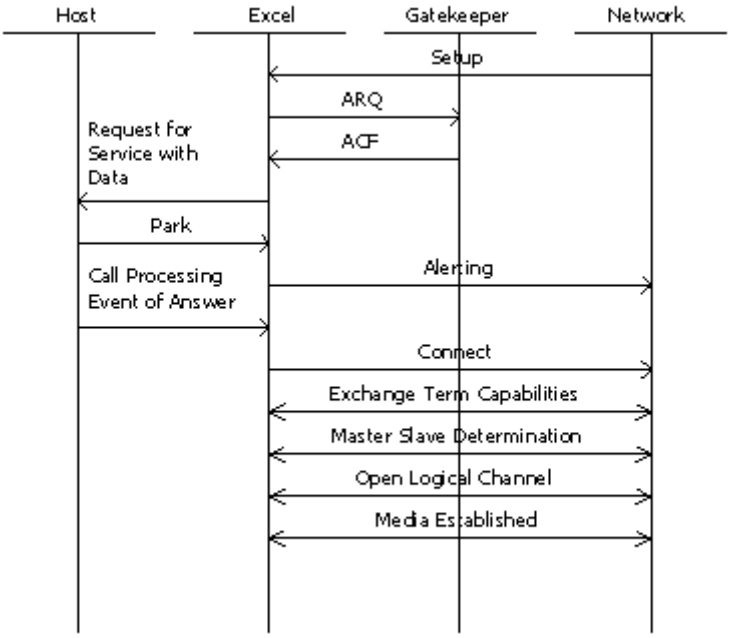
```
Route Control Ack
X->H
    [00 14 00 e8 00 01 01 00 10 01 02 1e 09 00 01 00 39 00
   03 00
     a9 02]

Receive Connect

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 86B0
      H225:  1... ....  .... .... = Message to originator
     H225:  .000 0110  1011 0000 = Call reference value =
   1712
      H225: Message type = 7 (Connect)
      H225:
      H225: Bearer Capacity
      H225: Information element identifier = 4
      H225: Length of bearer capacity = 3
      H225: Coding and capability flags = 88
      H225:                    1... .... = Expected
   extension bit
      H225:                   .00. .... = Coding standard
   = 0 (CCITT standardized coding)
      H225:                    ...0 1000 = Information
   transfer capability = 8 (Unrestricted digital
   information)
      H225: Mode and rate flags = C0
      H225:           1... .... = Expected extension bit
     H225:          .10. .... = Transfer mode = 2 (Packet
   mode)
     H225:          ...0 0000 = Information transfer rate
   = 0 (Used for packet mode calls)
      H225: Layer 1 protocol flag = A5
      H225:             1... .... = Expected extension bit
     H225:            .01. .... = Layer 1 identification
   = 1
      H225:              ...0 0101 = Layer 1 protocol = 5
   (H.221 and H.242)
      H225:
      H225: Display
      H225: Information element identifier = 40
      H225: Display length = 7
      H225: Display flag = 4C
      H225:    0... .... = Expected extension bit
      H225: Display = "Excel"
      H225:
H225:
      H225: ----User-User Information----
```

```
   H225:
   H225: Information element identifier = 126 (User-
User)
   H225: Length        = 152
   H225: Discriminator = 5 (X.208/X.209 (ASN.1))
   H225:
   H225: Flags     = 22
   H225: 0... .... = No extension values present in
H323-UserInformation
   H225: .0.. .... = user-data is not present
   H225: ..1. .... = Extension values present in h323-
uu-pdu
   H225: ...0 .... = nonStandardData is not present
   H225: .... 0... = No extension values present in
h323-message-body
   H225: h323-message = 2 (Connect)
   H225: Flags     = C0
   H225: 1... .... = Extension values present in
Connect-UUIE
   H225: .1.. .... = h245Address is present
   H225: Protocol Id = {0.0.8.2250.0.2}
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
h245Address
   H225: h245Address = 0 (IP address)
   H225: IP   = 135.119.55.170
   H225: Port = 2358
   H225:
   H225: Destination information
   H225: Flags     = 22
   H225: 0... .... = No extension value(s) present in
destinationInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..1. .... = Vendor is present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
vendor
   H225: Flags     = C0
   H225: 1... .... = Product ID is present
   H225: .1.. .... = Version ID is present
   H225:
   H225: ..0. .... = No extension value(s) present in
vendor
   H225: T.35 country code = 0x 9 (Australia)
   H225: T.35 extension    = 0
   H225: Manufacture code = 0x  3D (?)
```

```
   H225: Product Id = Equivalence OpenPhone<0000>
   H225: Version Id = 1.2.0<0000>
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
terminal
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = MC = 0
   H225: ...0 .... = Undefined node = 0
   H225: Conference id =
000006B0000006B03133353761636466
   H225:
   H225: Flags     = 0D
   H225: 0... .... = Extension length determinant
   H225: Number of extension = 7
   H225: .... ...1 = callIdentifier is present
   H225: Flags     = 1C
   H225: 0... .... = h245SecurityMode is not present
   H225: .0.. .... = tokens is not present
   H225: ..0. .... = cryptoTokens is not present
   H225: ...1 .... = fastStart is present
   H225: Length of callIdentifier = 17
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
callIdentifier
   H225: Guid = 000006B0000006B06162636431323334
   H225: Number of fastStart = 2
   H225: Length of fastStart = 25
   H225: Fast start = <0007130C> <13>
   H225: Length of fastStart = 23
   H225: Fast start = @<00>d<06040100>L <13>
   H225:
   H225: Flags     = 01
   H225: 0... .... = Extension length determinant
   H225: Number of extension = 1
   H225: .... ...1 = h4501SupplementaryService is
present
   H225: Flags     = 00
   H225: 0... .... = h245Tunneling is not present
   H225: .0.. .... = h245Control is not present
   H225: ..0. .... = nonStandardControl is not present
   H225: Number of h4501SupplementaryService = 0
   H225: Flags     = 80
   H225: 1... .... = H245 tunneling = 1
   H225:
FastStartElement Received
00 07 13 0c 20 13 80 11 1c 00 01 00 87 77 37 aa 13 88 00
   87 77 37 aa 13 89
FastStartElement Received
```

```
40 00 64 06 04 01 00 4c 20 13 80 0b 0d 00 01 00 87 77 37
   aa 13 89 80


Indication of Answer (if Answer Supervision is
   Configured)

X->H
    [00 0d 00 2e 00 07 01 00 01 0d 03 00 a8 07 20]
```

H.323 Incoming Call with H.245 Procedure



Message Trace

```
Receive Setup
H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 38DA
      H225:  0... ....  .... .... = Message from
   originator
     H225:  .011 1000  1101 1010 = Call reference value =
   14554
```

```
   H225: Message type = 5 (Setup)
   H225:
   H225: Bearer Capacity
   H225: Information element identifier = 4
   H225: Length of bearer capacity = 3
   H225: Coding and capability flags = 88
   H225:                    1... .... = Expected
extension bit
   H225:                   .00. .... = Coding standard
= 0 (CCITT standardized coding)
   H225:                    ...0 1000 = Information
transfer capability = 8 (Unrestricted digital
information)
   H225: Mode and rate flags = C0
   H225:           1... .... = Expected extension bit
   H225:          .10. .... = Transfer mode = 2 (Packet
mode)
   H225:          ...0 0000 = Information transfer rate
= 0 (Used for packet mode calls)
   H225: Layer 1 protocol flag = A5
   H225:             1... .... = Expected extension bit
   H225:            .01. .... = Layer 1 identification
= 1
   H225:             ...0 0101 = Layer 1 protocol = 5
(H.221 and H.242)
   H225:
   H225: Display
   H225: Information element identifier = 40
   H225: Display length = 7
   H225: Display flag = 4C
   H225:    0... .... = Expected extension bit
   H225: Display = "Excel"
   H225:
   H225: Calling Party Number
   H225: Information element identifier  = 108
   H225: Calling party number contents length = 11
   H225: Type and numbering flags = FF
   H225:               1... .... = Expected extension
bit
   H225:               .111 .... = Type of number = 7
(Reserved for extension)
   H225:              .... 1111 = Numbering plan ID =
15 (Reserved for extension)
   H225: Number digits flag = 35
   H225:          0... .... = Expected extension bit
   H225: Number digits = "5088623456"
   H225:
   H225: Called Party Number
   H225: Information element identifier  = 112
   H225: Called party number contents length = 11
   H225: Type and numbering flags = 81
```

```
      H225:                1... .... = Expected extension
   bit
      H225:                .000 .... = Type of number = 0
   (Unknown)
      H225:                .... 0001 = Numbering plan ID =
   1 (ISDN/telephony (E.164))
      H225: Call number flag = 35
      H225:        0... .... = Expected extension bit
      H225: Call number  = "5088622222"
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 348
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 20
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..1. .... = Extension values present in h323-
   uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 0 (Setup)
      H225:
      H225: Flags     = B8
      H225: 1... .... = Extension value(s) present in
   Setup
      H225: .0.. .... = h245Address is not present
      H225: ..1. .... = sourceAddress is present
      H225: ...1 .... = destinationAddress is present
      H225: .... 1... = destCallSignalAddress is present
      H225: .... .0.. = destExtraCallInfo is not present
      H225: .... ..0. = destExtraCRV is not present
      H225: .... ...0 = callServices is not present
      H225:
      H225: Protocol id = {0.0.8.2250.0.2}
      H225:
      H225: Number of sourceAddress = 2
      H225:
      H225: Flags     = 40
      H225: 0... .... = No extension value(s) present in
   sourceAddress
      H225: sourceAddress = 1 (H323-ID)
      H225: H323-ID = "Excel"
      H225:
      H225: Flags     = 04
```

```
   H225: 0... .... = No extension value(s) present in
sourceAddress
   H225: sourceAddress = 0 (E164)
   H225: Length of e164 = 10
   H225: E.164 = 5088623456
   H225:
   H225: Flags     = 22
   H225: 0... .... = No extension value(s) present in
sourceInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..1. .... = Vendor is present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
vendor
   H225: Flags     = C0
   H225: 1... .... = Product ID is present
   H225: .1.. .... = Version ID is present
   H225:
   H225: ..0. .... = No extension value(s) present in
vendor
   H225: T.35 country code = 0x 9 (Australia)
   H225: T.35 extension    = 0
   H225: Manufacture code = 0x  3D (?)
   H225: Product Id = Equivalence OpenPhone<0000>
   H225: Version Id = 1.2.0<0000>
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
terminal
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = MC = 0
   H225: ...0 .... = Undefined node = 0
   H225:
   H225: Number of destinationAddress = 1
   H225:
   H225: Flags     = 04
   H225: 0... .... = No extension value(s) present in
destinationAddress
   H225: destinationAddress = 0 (E164)
   H225: Length of e164 = 10
   H225: E.164 = 5088622222
   H225:
   H225: destCallSignalAddress
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
destCallSignalAddress
```

```
      H225: destCallSignalAddress = 0 (IP address)
      H225: IP   = 135.119.55.181
      H225: Port = 1720
      H225:
      H225: Flags     = 00
      H225: 0... .... = Active MC = 0
      H225: Conference id =
   D0B440311DEA181092E40050DAD08480
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   conferenceGoal
      H225: Conference goal = 0 (Create)
      H225:
      H225: callType
      H225: ...0 .... = No extension value(s) present in
   callType
      H225: callType = 0 (Point to point)
      H225:
      H225: .... ..0. = Extension length determinant
      H225: Number of extension = 12
     H225: .... .1.. = sourceCallSignalAddress is present



Send ARQ
----- H.225 Gatekeeper Registration, Admission and Status
   -----
      RAS:
      RAS: Flags     = 26
      RAS: 0... .... = No extension value(s) present in
   RasMessage
      RAS: RAS message = 9 (Admission Request)
      RAS: .... ..1. = Extension value(s) present in
   admissionRequest
      RAS: .... ...0 = callModel is not present
      RAS: Flags     = 80
      RAS: 1... .... = destinationInfo is present
      RAS: .0.. .... = destCallSignalAddress is not
   present
      RAS: ..0. .... = destExtraCallInfo is not present
     RAS: ...0 .... = srcCallSignalAddress is not present
      RAS: .... 0... = nonStandardData is not present
      RAS: .... .0.. = callServices is not present
      RAS: Request sequence number = 4
      RAS: Flags     = 03
      RAS: 0... .... = No extension value(s) present in
   callType
      RAS: callType = 0 (Point to point)
      RAS: Endpoint ID = "618B086000000003"
      RAS: Number of destinationInfo = 1
      RAS:
```

```
   RAS: Flags     = 04
   RAS: 0... .... = No extension value(s) present in
destinationInfo
   RAS: destinationInfo = 0 (E164)
   RAS: Length of e164 = 10
   RAS: E.164 = 5088622222
   RAS: Number of srcInfo = 2
   RAS:
   RAS: Flags     = 04
   RAS: 0... .... = No extension value(s) present in
srcInfo
   RAS: srcInfo = 0 (E164)
   RAS: Length of e164 = 10
   RAS: E.164 = 5088623456
   RAS:
   RAS: Flags     = 40
   RAS: 0... .... = No extension value(s) present in
srcInfo
   RAS: srcInfo = 1 (H323-ID)
   RAS: H323-ID = "Excel"
   RAS: Band width  = 1280
   RAS: Call reference value = 47322
   RAS: Conference id =
D0B440311DEA181092E40050DAD08480
   RAS: Flags     = 44
   RAS: 0... .... = Active MC = 0
   RAS: .1.. .... = Answer call = 1
   RAS:
   RAS: ..0. .... = Extension index format
   RAS: Number of extension = 10
   RAS: .1.. .... = canMapAlias is present
   RAS: ..1. .... = callIdentifier is present
   RAS: ...0 .... = srcAlternatives is not present
   RAS: .... 0... = destAlternatives is not present
  RAS: .... .0.. = gatekeeperIdentifier is not present
   RAS: .... ..0. = tokens is not present
   RAS: .... ...0 = cryptoTokens is not present
   RAS: Flags     = 20
   RAS: 0... .... = integrityCheckValue is not present
   RAS: .0.. .... = transportQOS is not present
   RAS: ..1. .... = willSupplyUUIEs is present
   RAS: Flags     = 80
   RAS: 1... .... = canMapAlias = 1
   RAS:
   RAS: Flags     = 00
   RAS: 0... .... = No extension value(s) present in
callIdentifier
   RAS: Guid = B9B840311DEA181092E40050DAD08480
   RAS: Flags     = 00
   RAS: 0... .... = Will supply UUIEs = 0
```

```
Receive ACF

----- H.225 Gatekeeper Registration, Admission and Status
   -----
      RAS:
      RAS: Flags     = 2B
      RAS: 0... .... = No extension value(s) present in
   RasMessage
      RAS: RAS message = 10 (Admission Confirm)
      RAS: .... ..1. = Extension value(s) present in
   admissionConfirm
      RAS: .... ...1 = irrFrequency is present
      RAS: Flags     = 00
      RAS: 0... .... = nonStandardData is not present
      RAS: Request sequence number = 4
      RAS: Band width  = 1280
      RAS:
      RAS: Flags     = 00
      RAS: 0... .... = No extension value(s) present in
   callModel
      RAS: Call model = 0 (Direct)
      RAS:
      RAS: ..0. .... = No extension value(s) present in
   destCallSignalAddress
      RAS: destCallSignalAddress = 0 (IP address)
      RAS: IP   = 135.119.55.181
      RAS: Port = 1720
      RAS: IrrFrequency = 239
      RAS:
      RAS: Flags     = 28
      RAS: 0... .... = Extension index format
      RAS: Number of extension = 21
      RAS: .... ...0 = destinationInfo is not present
      RAS: Flags     = 00
      RAS: 0... .... = destExtraCallInfo is not present
      RAS: .0.. .... = destinationType is not present
      RAS: ..0. .... = remoteExtensionAddress is not
   present
      RAS: ...0 .... = alternateEndpoints is not present
      RAS: .... 0... = tokens is not present
      RAS: .... .0.. = cryptoTokens is not present
      RAS: .... ..0. = integrityCheckValue is not present
      RAS: .... ...0 = transportQOS is not present
      RAS: Flags     = C0
      RAS: 1... .... = willRespondToIRR is present
      RAS: .1.. .... = uuiesRequested is present
      RAS: Flags     = 01
      RAS: 0... .... = willRespondToIRR = 0
      RAS:
```

```
      RAS: Flags     = 02
      RAS: 0... .... = No extension value(s) present in
   uuiesRequested
      RAS: .0.. .... = Setup = 0
      RAS: ..0. .... = Call proceeding = 0
      RAS: ...0 .... = Connect = 0
      RAS: .... 0... = Alerting = 0
      RAS: .... .0.. = User Information = 0
      RAS: .... ..1. = Release complete = 1
      RAS: .... ...0 = Facility = 0
      RAS: Flags     = 00
      RAS: 0... .... = Progress = 0
      RAS: .0.. .... = Empty = 0
Request for Service with Data
X->H
    [00 6e 00 2d 00 06 01 00 01 0d 03 00 a8 06 00 33 01 03
   00 33
     00 5a 00 0a 27 7e 00 03 09 00 00 27 4e 00 02 00 05 27
     92 00 04 87 77 37 b7 27 93 00 04 00 00 38 80 27 18 00
     0a 04 00 00 00 0a 50 88 62 34 56 27 17 00 08 04 00 0a
     50 88 62 22 22 27 c1 00 02 00 3c 27 d8 00 07 4c 75 63
     65 6e 74 00 27 c2 00 04 87 77 37 b5 27 c3 00 04 00 00
     06 b8]

Park the Channel

H->X
    [00 11 00 bf 00 01 01 00 02 0d 03 00 a8 06 0d 03 00 a8
     06]

X->H
    [00 07 00 bf 00 01 01 00 10]

Send Alerting

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = B8DA
      H225:  1... ....  .... .... = Message to originator
     H225:  .011 1000  1101 1010 = Call reference value =
   14554
      H225: Message type = 2 (Call Proceeding)
      H225:
      H225:
H225:
      H225: ----User-User Information----
      H225:
```

```
   H225: Information element identifier = 126 (User-
User)
   H225: Length        = 32
   H225: Discriminator = 5 (X.208/X.209 (ASN.1))
   H225:
   H225: Flags     = 01
   H225: 0... .... = No extension values present in
H323-UserInformation
   H225: .0.. .... = user-data is not present
   H225: ..0. .... = No extension values present in
h323-uu-pdu
   H225: ...0 .... = nonStandardData is not present
   H225: .... 0... = No extension values present in
h323-message-body
   H225: h323-message = 1 (Call proceeding)
   H225: Flags     = 80
   H225: 1... .... = Extension values present in
CallProceeding-UUIE
   H225: .0.. .... = h245Address is not present
   H225: Protocol Id = {0.0.8.2250.0.2}
   H225:
   H225: Destination information
   H225: Flags     = 02
   H225: 0... .... = No extension value(s) present in
destinationInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = Vendor is not present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
terminal
   H225: Flags     = 01
   H225: 0... .... = nonStandardData is not present
   H225: .0.. .... = MC = 0
   H225: ..0. .... = Undefined node = 0
   H225:
   H225: ...0 .... = Extension length determinant
   H225: Number of extension = 5
   H225: ..1. .... = callIdentifier is present
   H225: ...0 .... = h245SecurityMode is not present
   H225: .... 0... = tokens is not present
   H225: .... .0.. = cryptoTokens is not present
   H225: .... ..0. = fastStart is not present
   H225: Length of callIdentifier = 17
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
callIdentifier
```

```
      H225: Guid = B9B840311DEA181092E40050DAD08480
      H225:

Call Processing Event of Answer

H->X
    [00 0d 00 ba 00 02 01 00 01 0d 03 00 a8 06 01]

X->H
    [00 07 00 ba 00 02 01 00 10]


Send Connect
----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = B8DA
      H225:  1... ....  .... .... = Message to originator
     H225:  .011 1000  1101 1010 = Call reference value =
   14554
      H225: Message type = 7 (Connect)
      H225:
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 56
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 02
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..0. .... = No extension values present in
   h323-uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 2 (Connect)
      H225: Flags     = C0
      H225: 1... .... = Extension values present in
   Connect-UUIE
      H225: .1.. .... = h245Address is present
      H225: Protocol Id = {0.0.8.2250.0.2}
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   h245Address
```

```
      H225: h245Address = 0 (IP address)
      H225: IP   = 135.119.55.181
      H225: Port = 53500
      H225:
      H225: Destination information
      H225: Flags     = 02
      H225: 0... .... = No extension value(s) present in
    destinationInfo
      H225: .0.. .... = nonStandardData is not present
      H225: ..0. .... = Vendor is not present
      H225: ...0 .... = Gatekeeper is not present
      H225: .... 0... = Gateway is not present
      H225: .... .0.. = MCU is not present
      H225: .... ..1. = Terminal is present
      H225:
      H225: .... ...0 = No extension value(s) present in
    terminal
      H225: Flags     = 00
      H225: 0... .... = nonStandardData is not present
      H225: .0.. .... = MC = 0
      H225: ..0. .... = Undefined node = 0
      H225: Conference id =
    D0B440311DEA181092E40050DAD08480
      H225:
      H225: Flags     = 09
      H225: 0... .... = Extension length determinant
      H225: Number of extension = 5
      H225: .... ...1 = callIdentifier is present
      H225: Flags     = 00
      H225: 0... .... = h245SecurityMode is not present
      H225: .0.. .... = tokens is not present
      H225: ..0. .... = cryptoTokens is not present
      H225: ...0 .... = fastStart is not present
      H225: Length of callIdentifier = 17
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
    callIdentifier
      H225: Guid = B9B840311DEA181092E40050DAD08480
      H225:


   Send Terminal capabilities

   ----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 330
      H245: Flags     = 02
      H245: 0... .... = No extension value(s) present in
    H.245 control message
```

```
   H245: H.245 call control message type = 0 (Request)
   H245:
   H245: ...0 .... = No extension value(s) present in
request
   H245: Request type = 2 (Terminal Capability Set)
   H245:
   H245: Flags     = 70
   H245: 0... .... = No extension value(s) present in
terminalCapabilitySet
   H245: .1.. .... = multiplexCapability is present
   H245: ..1. .... = capabilityTable is present
   H245: ...1 .... = capabilityDescriptors is present
   H245: Sequence number = 1
   H245: Protocol id = {0.0.0.0.0.8}
   H245:
   H245: Flags     = 80
   H245: 1... .... = Extension value(s) present in
MultiplexCapability
   H245: .0.. .... = Choice value is 6 bits long
   H245: Multiplex capability = 0 (H2250 Capability)
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
h2250Capability
   H245: Maximum audio delay jitter       = 60 (ms)
   H245:
   H245: *** Receive Multipoint Capability ***
   H245:
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
receive multipoint capability
   H245: .0.. .... = Multicast capability is OFF
   H245: ..0. .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
   H245: *** Transmit Multipoint Capability ***
   H245:
   H245: .0.. .... = No extension value(s) present in
transmit multipoint capability
```

```
   H245: ..0. .... = Multicast capability is OFF
   H245: ...0 .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
  H245: *** Receive and Transmit Multipoint Capability
***
   H245:
   H245: .0.. .... = No extension value(s) present in
receive and transmit multipoint capability
   H245: ..0. .... = Multicast capability is OFF
   H245: ...0 .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
   H245: *** MC Capability ***
   H245:
   H245: .0.. .... = No extension value(s) present in
mcCapability
   H245: ..0. .... = Centralized Conference MC is OFF
  H245: ...0 .... = Decentralized Conference MC is OFF
   H245:
   H245: .... 0... = RTCP video control capability is
OFF
   H245:
   H245: *** Media Packetization Capability ***
   H245:
```

```
   H245: .... .0.. = No extension value(s) present in
MediaPacketizationCapability
   H245: .... ..0. = H261A video packetization is OFF
   H245:
   H245: *** Capability Table ***
   H245:
   H245: Number of capability table entry = 29
   H245:
   H245: Capability Table Entry Set #1
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 1
   H245: Flags     = 28
   H245: 0... .... = No extension value(s) present in
Capability
   H245: Capability = 5 (Transmit audio capability)
   H245: .... .0.. = No extension value(s) present in
AudioCapability
   H245: Audio capability = 3 (G.711 ulaw 64k)
   H245: Value = 20
   H245:
   H245: Capability Table Entry Set #2
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 2
   H245: Flags     = 20
   H245: 0... .... = No extension value(s) present in
Capability
   H245: Capability = 4 (Receive audio capability)
   H245: .... .0.. = No extension value(s) present in
AudioCapability
   H245: Audio capability = 3 (G.711 ulaw 64k)
   H245: Value = 20
   H245:
   H245: Capability Table Entry Set #3
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 3
   H245: Flags     = 28
   H245: 0... .... = No extension value(s) present in
Capability
   H245: Capability = 5 (Transmit audio capability)
   H245: .... .0.. = No extension value(s) present in
AudioCapability
   H245: Audio capability = 1 (G.711 alaw 64k)
   H245: Value = 20
   H245:
   H245: Capability Table Entry Set #4
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 4
```

```
      H245: Flags     = 20
      H245: 0... .... = No extension value(s) present in
   Capability
      H245: Capability = 4 (Receive audio capability)
      H245: .... .0.. = No extension value(s) present in
   AudioCapability
      H245: Audio capability = 1 (G.711 alaw 64k)
      H245: Value = 20
      H245:
      H245: Capability Table Entry Set #5
      H245: Flags     = 80
      H245: 1... .... = capability is present
      H245: Capability table entry number = 5
      H245: Flags     = 2A
      H245: 0... .... = No extension value(s) present in
   Capability


Receive Terminal capabilities

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 194
      H245: Flags     = 02
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 2 (Terminal Capability Set)
      H245:
      H245: Flags     = 70
      H245: 0... .... = No extension value(s) present in
   terminalCapabilitySet
      H245: .1.. .... = multiplexCapability is present
      H245: ..1. .... = capabilityTable is present
      H245: ...1 .... = capabilityDescriptors is present
      H245: Sequence number = 2
      H245: Protocol id = {0.0.8.245.0.3}
      H245:
      H245: Flags     = 80
      H245: 1... .... = Extension value(s) present in
   MultiplexCapability
      H245: .0.. .... = Choice value is 6 bits long
      H245: Multiplex capability = 0 (H2250 Capability)
      H245: Flags     = 80
      H245: 1... .... = Extension value(s) present in
   h2250Capability
      H245: Maximum audio delay jitter       = 50 (ms)
      H245:
```

```
   H245: *** Receive Multipoint Capability ***
   H245:
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
receive multipoint capability
   H245: .0.. .... = Multicast capability is OFF
   H245: ..0. .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
   H245: *** Transmit Multipoint Capability ***
   H245:
   H245: .0.. .... = No extension value(s) present in
transmit multipoint capability
   H245: ..0. .... = Multicast capability is OFF
   H245: ...0 .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
  H245: *** Receive and Transmit Multipoint Capability
***
   H245:
   H245: .0.. .... = No extension value(s) present in
receive and transmit multipoint capability
   H245: ..0. .... = Multicast capability is OFF
   H245: ...0 .... = Multi unicast conference is OFF
   H245:
```

```
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
   H245: *** MC Capability ***
   H245:
   H245: .0.. .... = No extension value(s) present in
mcCapability
   H245: ..0. .... = Centralized Conference MC is OFF
   H245: ...0 .... = Decentralized Conference MC is OFF
   H245:
   H245: .... 0... = RTCP video control capability is
OFF
   H245:
   H245: *** Media Packetization Capability ***
   H245:
   H245: .... .0.. = No extension value(s) present in
MediaPacketizationCapability
   H245: .... ..0. = H261A video packetization is OFF
   H245: .... ...0 = Extension index format
   H245: Number of extension = 4
   H245: .... ..0. = transportCapability is not present
   H245: .... ...0 = redundancyEncodingCapability is
not present
   H245: Flags     = C0
   H245: 1... .... = logicalChannelSwitchingCapability
is present
   H245: .1.. .... = t120DynamicPortCapability is
present
   H245: Flags     = 00
   H245: 0... .... = Logical channel switching
capability is OFF
   H245: Flags     = 00
   H245: 0... .... = T120 dynamic port capability is
OFF
   H245:
   H245: *** Capability Table ***
   H245:
   H245: Number of capability table entry = 10
   H245:
   H245: Capability Table Entry Set #1
```

```
      H245: Flags     = 80
      H245: 1... .... = capability is present
      H245: Capability table entry number = 1
      H245: Flags     = 20
      H245: 0... .... = No extension value(s) present in
   Capability
      H245: Capability = 4 (Receive audio capability)
      H245: .... .0.. = No extension value(s) present in
   AudioCapability
      H245: Audio capability = 0 (Non standard)
      H245: nonStandardIdentifier = 1 (H221 Non Standard)
      H245: T.35 country code = 0x 9 (Australia)
      H245: T.35 extension    = 0
      H245: Manufacture code = 0x  3D (?)
      H245: Data: 6 byte(s) of data
      H245:
      H245: Capability Table Entry Set #2
      H245: Flags     = 80
      H245: 1... .... = capability is present
      H245: Capability table entry number = 2
      H245: Flags     = 24
      H245: 0... .... = No extension value(s) present in
   Capability
      H245: Capability = 4 (Receive audio capability)
      H245: .... .1.. = Extension value(s) present in
   AudioCapability
      H245: .... ..0. = Extension index format
      H245: Audio capability = 3 (GSM full rate)
      H245: .... .0.. = No extension value(s) present in
   GSMAudioCapability
      H245: Audio unit size = 4
      H245: Flags     = 00
      H245: 0... .... = Confort noise
      H245: .0.. .... = Scrambled
      H245:
      H245: Capability Table Entry Set #3
      H245: ..0. .... = capability is not present
      H245: Capability table entry number = 58881
      H245:
      H245: Capability Table Entry Set #4
      H245: Flags     = 80
      H245: 1... .... = capability is present
      H245: Capability table entry number = 3

Send Request Master Slave determination

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 11
      H245: Flags     = 01
```

```
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 1 (Master Slave Determination)
      H245:
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   MasterSlaveDetermination
      H245: Terminal type = 50
      H245: Status determination number = 16750783

Receive Request Master Slave Determination

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 11
      H245: Flags     = 01
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 1 (Master Slave Determination)
      H245:
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   MasterSlaveDetermination
      H245: Terminal type = 50
      H245: Status determination number = 3293797


Send Response Terminal capabilities

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 7
      H245: Flags     = 21
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 3 (Terminal Capability Set Ack)
      H245: .0.. .... = No extension value(s) present in
   TerminalCapabilitySetAck
      H245: Sequence number = 2
```

```
Receive Response Terminal Capabilities

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 7
      H245: Flags     = 21
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 3 (Terminal Capability Set Ack)
      H245: .0.. .... = No extension value(s) present in
   TerminalCapabilitySetAck
      H245: Sequence number = 1

Send Response Master Slave Determination

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 6
      H245: Flags     = 20
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 1 (Master Slave Determination Ack)
      H245: .0.. .... = No extension value(s) present in
   MasterSlaveDeterminationAck
      H245: Decision  = 1 (Slave)

Receive Response Master Slave Determination

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 6
      H245: Flags     = 20
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 1 (Master Slave Determination Ack)
      H245: .0.. .... = No extension value(s) present in
   MasterSlaveDeterminationAck
      H245: Decision  = 0 (Master)
```

```
Receive Request Open Logical Channel
----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 24
      H245: Flags     = 03
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 3 (Open Logical Channel)
      H245:
      H245:
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   OpenLogicalChannel
        H245: .0.. .... = reverseLogicalChannelParameters is
   not present
      H245: Forward logical channel number = 100
      H245:
      H245: Flags     = 0C
      H245: 0... .... = No extension value(s) present in
   forwardLogicalChannelParameters
      H245: .0.. .... = portNumber is not present
      H245:
      H245: ..0. .... = No extension value(s) present in
   dataType
      H245: Data type = 3 (Audio data)
      H245: .... ..0. = No extension value(s) present in
   audioData
      H245: Audio capability = 3 (G.711 ulaw 64k)
      H245: Value = 20
      H245:
      H245: Flags     = 80
      H245: 1... .... = Extension value(s) present in
   MultiplexParameters
      H245: .0.. .... = Extension index format
     H245: Multiplex parameters = 0 (H225 logical channel
   parameters)
      H245:
      H245: Flags     = 0D
      H245: 0... .... = No extension value(s) present in
   h2250LogicalChannelParameters
      H245: .0.. .... = nonStandard is not present
     H245: ..0. .... = associatedSessionID is not present
      H245: ...0 .... = mediaChannel is not present
     H245: .... 1... = mediaGuaranteedDelivery is present
      H245: .... .1.. = mediaControlChannel is present
```

```
      H245: .... ..0. = mediaControlGuaranteedDelivery is
    not present
       H245: .... ...1 = silenceSuppression is present
       H245: Flags     = 00
       H245: 0... .... = destination is not present
       H245: .0.. .... = dynamicRTPPayloadType is not
    present
       H245: ..0. .... = mediaPacketization is not present
       H245: Session ID = 1
       H245: Flags     = 00
       H245: 0... .... = Media guaranted delivery is OFF
       H245:
       H245: *** Media Control Channel ***
       H245: .0.. .... = No extension value(s) present in
    mediaControlChannel(RTCP)
       H245: Transport Address = 0 (Unicast address)
       H245: ...0 .... = No extension value(s) present in
    UnicastAddress
       H245: Unicast address = 0 (IP address)
       H245: .... ...0 = No extension value(s) present in
    iPAddress
       H245: Network = 135.119.55.170
       H245: TSAP indentifier = 5001
       H245: Flags     = 80
       H245: 1... .... = Silence suppression is ON

Send Response Open Logical Channel
----- Control Protocol for Multimedia Communication -----
       H245:
       H245: Message length = 27
       H245: Flags     = 22
       H245: 0... .... = No extension value(s) present in
    H.245 control message
      H245: H.245 call control message type = 1 (Response)
       H245:
       H245: ...0 .... = No extension value(s) present in
    ResponseMessage
       H245: Response  = 5 (Open Logical Channel Ack)
       H245:
       H245: .1.. .... = Extension value(s) present in
    openLogicalChannelAck
      H245: ..0. .... = reverseLogicalChannelParameters is
    not present
       H245: Forward logical channel number = 100
       H245:
       H245: Flags     = 04
       H245: 0... .... = Extension index format
       H245: Number of extension = 3
       H245: .... ...0 = separateStack is not present
       H245: Flags     = 80
```

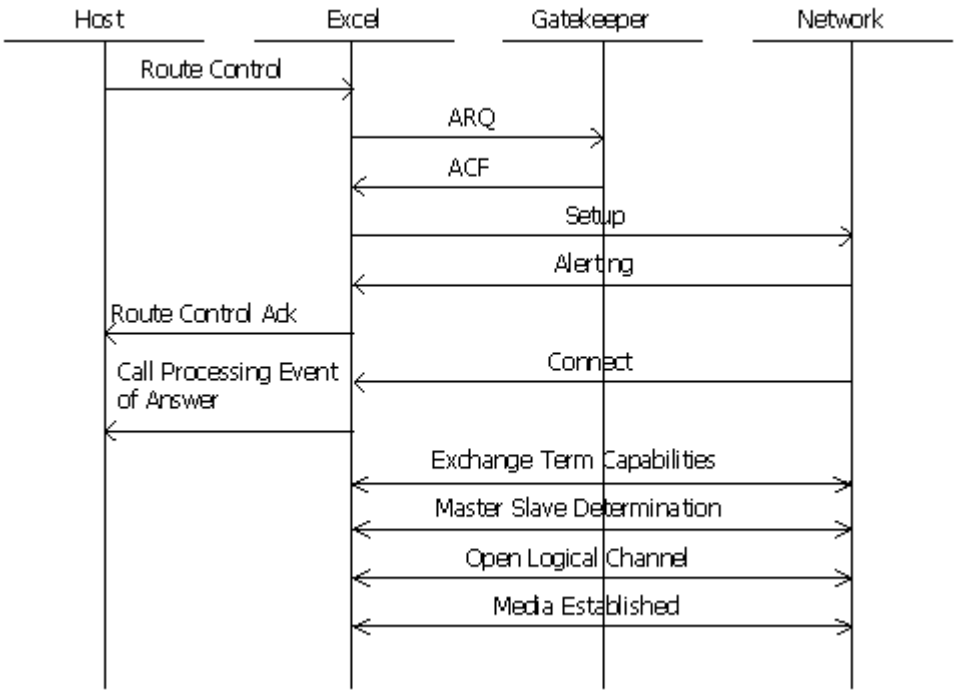
```
      H245: 1... .... = forwardMultiplexAckParameters is
   present
      H245: .0.. .... = encryptionSync is not present
      H245: Flags     = 1C
      H245: 0... .... = No extension value(s) present in
   forwardMultiplexAckParameters
      H245: .0.. .... = No extension value(s) present in
   H2250LogicalChannelAckParameters
      H245: ..0. .... = nonStandard is not present
      H245: ...1 .... = sessionID is present
      H245: .... 1... = mediaChannel is present
      H245: .... .1.. = mediaControlChannel is present
      H245: .... ..0. = dynamicRTPPayloadType is not
   present
      H245: Session ID = 1
      H245: .... ...0 = No extension value(s) present in
   mediaChannel
      H245: Transport Address = 0 (Unicast address)
      H245: .0.. .... = No extension value(s) present in
   UnicastAddress
      H245: Unicast address = 0 (IP address)
      H245: .... .0.. = No extension value(s) present in
   iPAddress
      H245: Network = 135.119.55.183
      H245: TSAP indentifier = 14464
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   mediaControlChannel (forward RTCP channel)
      H245: Transport Address = 0 (Unicast address)
      H245: ..0. .... = No extension value(s) present in
   UnicastAddress
      H245: Unicast address = 0 (IP address)
      H245: .... ..0. = No extension value(s) present in
   iPAddress
      H245: Network = 135.119.55.183
      H245: TSAP indentifier = 14465

Send Request Open Logical Channel

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 24
      H245: Flags     = 03
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 3 (Open Logical Channel)
      H245:
```

```
   H245:
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
OpenLogicalChannel
  H245: .0.. .... = reverseLogicalChannelParameters is
not present
   H245: Forward logical channel number = 1715
   H245:
   H245: Flags     = 0C
   H245: 0... .... = No extension value(s) present in
forwardLogicalChannelParameters
   H245: .0.. .... = portNumber is not present
   H245:
   H245: ..0. .... = No extension value(s) present in
dataType
   H245: Data type = 3 (Audio data)
   H245: .... ..0. = No extension value(s) present in
audioData
   H245: Audio capability = 3 (G.711 ulaw 64k)
   H245: Value = 20
   H245:
   H245: Flags     = 80
   H245: 1... .... = Extension value(s) present in
MultiplexParameters
   H245: .0.. .... = Extension index format
  H245: Multiplex parameters = 0 (H225 logical channel
parameters)
   H245:
   H245: Flags     = 05
   H245: 0... .... = No extension value(s) present in
h2250LogicalChannelParameters
   H245: .0.. .... = nonStandard is not present
  H245: ..0. .... = associatedSessionID is not present
   H245: ...0 .... = mediaChannel is not present
   H245: .... 0... = mediaGuaranteedDelivery is not
present
   H245: .... .1.. = mediaControlChannel is present
  H245: .... ..0. = mediaControlGuaranteedDelivery is
not present
   H245: .... ...1 = silenceSuppression is present
   H245: Flags     = 00
   H245: 0... .... = destination is not present
   H245: .0.. .... = dynamicRTPPayloadType is not
present
   H245: ..0. .... = mediaPacketization is not present
   H245: Session ID = 1
   H245:
   H245: *** Media Control Channel ***
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaControlChannel(RTCP)
```

```
      H245: Transport Address = 0 (Unicast address)
      H245: ..0. .... = No extension value(s) present in
   UnicastAddress
      H245: Unicast address = 0 (IP address)
      H245: .... ..0. = No extension value(s) present in
   iPAddress
      H245: Network = 135.119.55.183
      H245: TSAP indentifier = 14465
      H245: Flags     = 00
      H245: 0... .... = Silence suppression is OFF


Receive Response Open Logical Channel

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 30
      H245: Flags     = 22
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 5 (Open Logical Channel Ack)
      H245:
      H245: .1.. .... = Extension value(s) present in
   openLogicalChannelAck
      H245: ..0. .... = reverseLogicalChannelParameters is
   not present
      H245: Forward logical channel number = 1715
      H245:
      H245: Flags     = 02
      H245: 0... .... = Extension index format
      H245: Number of extension = 2
      H245: .... ...0 = separateStack is not present
      H245: Flags     = 80
      H245: 1... .... = forwardMultiplexAckParameters is
   present
      H245: .0.. .... = encryptionSync is not present
      H245: Flags     = 5C
      H245: 0... = No extension value(s) present in
   forward MultiplexAck Parameters
      H245: .1.. .... = Extension value(s) present in
   H2250LogicalChannelAckParameters
      H245: ..0. .... = nonStandard is not present
      H245: ...1 .... = sessionID is present
      H245: .... 1... = mediaChannel is present
      H245: .... .1.. = mediaControlChannel is present
      H245: .... ..0. = dynamicRTPPayloadType is not
   present
```

```
   H245: Session ID = 1
   H245: .... ...0 = No extension value(s) present in
mediaChannel
   H245: Transport Address = 0 (Unicast address)
   H245: .0.. .... = No extension value(s) present in
UnicastAddress
   H245: Unicast address = 0 (IP address)
   H245: .... .0.. = No extension value(s) present in
iPAddress
   H245: Network = 135.119.55.170
   H245: TSAP indentifier = 5000
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaControlChannel (forward RTCP channel)
   H245: Transport Address = 0 (Unicast address)
   H245: ..0. .... = No extension value(s) present in
UnicastAddress
   H245: Unicast address = 0 (IP address)
   H245: .... ..0. = No extension value(s) present in
iPAddress
   H245: Network = 135.119.55.170
   H245: TSAP indentifier = 5001
   H245:
   H245: Flags     = 01
   H245: 0... .... = Extension index format
   H245: Number of extension = 1
   H245: .... ...1 = flowControlToZero is present
   H245: Flags     = 00
   H245: 0... .... = Flow control to zero is OFF
```

H.323 Outgoing Call with H.245 Procedure



Message Trace

```
Receive Route Control

H->X
    [00 53 00 e8 00 02 01 00 01 29 02 ff fe 02 03 00 1e 00
     19 00 04 00 13 00 02 00 08 00 08 00 02 00 65 00 0f 00
     01 0b 00 65 00 02 00 00 03 00 33 00 24 00 04 27 7e 00
     01 09 27 c1 00 02 00 3c 27 17 00 08 10 00 0a 50 88 62
     34 56 27 18 00 07 10 00 04 0a 04 30 00]



Send ARQ

----- H.225 Gatekeeper Registration, Admission and Status
   -----
      RAS:
      RAS: Flags     = 26
      RAS: 0... .... = No extension value(s) present in
   RasMessage
      RAS: RAS message = 9 (Admission Request)
```

```
   RAS: .... ..1. = Extension value(s) present in
admissionRequest
   RAS: .... ...0 = callModel is not present
   RAS: Flags     = 80
   RAS: 1... .... = destinationInfo is present
   RAS: .0.. .... = destCallSignalAddress is not
present
   RAS: ..0. .... = destExtraCallInfo is not present
   RAS: ...0 .... = srcCallSignalAddress is not present
   RAS: .... 0... = nonStandardData is not present
   RAS: .... .0.. = callServices is not present
   RAS: Request sequence number = 11
   RAS: Flags     = 03
   RAS: 0... .... = No extension value(s) present in
callType
   RAS: callType = 0 (Point to point)
   RAS: Endpoint ID = "618B086000000003"
   RAS: Number of destinationInfo = 1
   RAS:
   RAS: Flags     = 04
   RAS: 0... .... = No extension value(s) present in
destinationInfo
   RAS: destinationInfo = 0 (E164)
   RAS: Length of e164 = 10
   RAS: E.164 = 5088623456
   RAS: Number of srcInfo = 1
   RAS:
   RAS: Flags     = 01
   RAS: 0... .... = No extension value(s) present in
srcInfo
   RAS: srcInfo = 0 (E164)
   RAS: Length of e164 = 4
   RAS: E.164 = 3000
   RAS: Band width  = 1280
   RAS: Call reference value = 1664
   RAS: Conference id =
00000680000006803133353761636466
   RAS: Flags     = 04
   RAS: 0... .... = Active MC = 0
   RAS: .0.. .... = Answer call = 0
   RAS:
   RAS: ..0. .... = Extension index format
   RAS: Number of extension = 10
   RAS: .1.. .... = canMapAlias is present
   RAS: ..1. .... = callIdentifier is present
   RAS: ...0 .... = srcAlternatives is not present
   RAS: .... 0... = destAlternatives is not present
   RAS: .... .0.. = gatekeeperIdentifier is not present
   RAS: .... ..0. = tokens is not present
   RAS: .... ...0 = cryptoTokens is not present
   RAS: Flags     = 20
```

```
      RAS: 0... .... = integrityCheckValue is not present
      RAS: .0.. .... = transportQOS is not present
      RAS: ..1. .... = willSupplyUUIEs is present
      RAS: Flags     = 80
      RAS: 1... .... = canMapAlias = 1
      RAS:
      RAS: Flags     = 00
      RAS: 0... .... = No extension value(s) present in
   callIdentifier
      RAS: Guid = 00000680000006806162636431323334
      RAS: Flags     = 00
      RAS: 0... .... = Will supply UUIEs = 0

Receive ACF

----- H.225 Gatekeeper Registration, Admission and Status
   -----
      RAS:
      RAS: Flags     = 2B
      RAS: 0... .... = No extension value(s) present in
   RasMessage
      RAS: RAS message = 10 (Admission Confirm)
      RAS: .... ..1. = Extension value(s) present in
   admissionConfirm
      RAS: .... ...1 = irrFrequency is present
      RAS: Flags     = 00
      RAS: 0... .... = nonStandardData is not present
      RAS: Request sequence number = 11
      RAS: Band width  = 1280
      RAS:
      RAS: Flags     = 00
      RAS: 0... .... = No extension value(s) present in
   callModel
      RAS: Call model = 0 (Direct)
      RAS:
      RAS: ..0. .... = No extension value(s) present in
   destCallSignalAddress
      RAS: destCallSignalAddress = 0 (IP address)
      RAS: IP   = 135.119.55.170
      RAS: Port = 1720
      RAS: IrrFrequency = 239
      RAS:
      RAS: Flags     = 28
      RAS: 0... .... = Extension index format
      RAS: Number of extension = 21
      RAS: .... ...0 = destinationInfo is not present
      RAS: Flags     = 00
      RAS: 0... .... = destExtraCallInfo is not present
      RAS: .0.. .... = destinationType is not present
      RAS: ..0. .... = remoteExtensionAddress is not
   present
```

```
      RAS: ...0 .... = alternateEndpoints is not present
      RAS: .... 0... = tokens is not present
      RAS: .... .0.. = cryptoTokens is not present
      RAS: .... ..0. = integrityCheckValue is not present
      RAS: .... ...0 = transportQOS is not present
      RAS: Flags     = C0
      RAS: 1... .... = willRespondToIRR is present
      RAS: .1.. .... = uuiesRequested is present
      RAS: Flags     = 01
      RAS: 0... .... = willRespondToIRR = 0
      RAS:
      RAS: Flags     = 02
      RAS: 0... .... = No extension value(s) present in
   uuiesRequested
      RAS: .0.. .... = Setup = 0
      RAS: ..0. .... = Call proceeding = 0
      RAS: ...0 .... = Connect = 0
      RAS: .... 0... = Alerting = 0
      RAS: .... .0.. = User Information = 0
      RAS: .... ..1. = Release complete = 1
      RAS: .... ...0 = Facility = 0
      RAS: Flags     = 00
      RAS: 0... .... = Progress = 0
      RAS: .0.. .... = Empty = 0


Send Setup

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 0680
      H225:  0... ....  .... .... = Message from
   originator
     H225:  .000 0110  1000 0000 = Call reference value =
   1664
      H225: Message type = 5 (Setup)
      H225:
      H225: Bearer Capacity
      H225: Information element identifier = 4
      H225: Length of bearer capacity = 3
      H225: Coding and capability flags = 88
      H225:                    1... .... = Expected
   extension bit
      H225:                   .00. .... = Coding standard
   = 0 (CCITT standardized coding)
      H225:                    ...0 1000 = Information
   transfer capability = 8 (Unrestricted digital
   information)
      H225: Mode and rate flags = D0
```

```
    H225:           1... .... = Expected extension bit
    H225:          .10. .... = Transfer mode = 2 (Packet
mode)
    H225:          ...1 0000 = Information transfer rate
= 16 (64 kbit/s)
    H225: Layer 1 protocol flag = A5
    H225:             1... .... = Expected extension bit
    H225:            .01. .... = Layer 1 identification
= 1
    H225:             ...0 0101 = Layer 1 protocol = 5
(H.221 and H.242)
    H225:
    H225: Display
    H225: Information element identifier = 40
    H225: Display length = 6
    H225: Display flag = 4C
    H225:    0... .... = Expected extension bit
    H225: Display = "Excel"
    H225:
    H225: Calling Party Number
    H225: Information element identifier  = 108
    H225: Calling party number contents length = 5
    H225: Type and numbering flags = 89
    H225:               1... .... = Expected extension
bit
    H225:               .000 .... = Type of number = 0
(Unknown)
    H225:               .... 1001 = Numbering plan ID =
9 (Private)
    H225: Number digits flag = 33
    H225:          0... .... = Expected extension bit
    H225: Number digits = "3000"
    H225:
    H225: Called Party Number
    H225: Information element identifier  = 112
    H225: Called party number contents length = 11
    H225: Type and numbering flags = 89
    H225:               1... .... = Expected extension
bit
    H225:               .000 .... = Type of number = 0
(Unknown)
    H225:               .... 1001 = Numbering plan ID =
9 (Private)
    H225: Call number flag = 35
    H225:        0... .... = Expected extension bit
    H225: Call number  = "5088623456"
    H225:
H225:
    H225: ----User-User Information----
    H225:
```

```
   H225: Information element identifier = 126 (User-
User)
   H225: Length        = 145
   H225: Discriminator = 5 (X.208/X.209 (ASN.1))
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension values present in
H323-UserInformation
   H225: .0.. .... = user-data is not present
   H225: ..0. .... = No extension values present in
h323-uu-pdu
   H225: ...0 .... = nonStandardData is not present
   H225: .... 0... = No extension values present in
h323-message-body
   H225: h323-message = 0 (Setup)
   H225:
   H225: Flags     = F0
   H225: 1... .... = Extension value(s) present in
Setup
   H225: .1.. .... = h245Address is present
   H225: ..1. .... = sourceAddress is present
   H225: ...1 .... = destinationAddress is present
   H225: .... 0... = destCallSignalAddress is not
present
   H225: .... .0.. = destExtraCallInfo is not present
   H225: .... ..0. = destExtraCRV is not present
   H225: .... ...0 = callServices is not present
   H225:
   H225: Protocol id = {0.0.8.2250.0.2}
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
h245Address
   H225: h245Address = 0 (IP address)
   H225: IP   = 135.119.55.181
   H225: Port = 53502
   H225:
   H225: Number of sourceAddress = 1
   H225:
   H225: Flags     = 01
   H225: 0... .... = No extension value(s) present in
sourceAddress
   H225: sourceAddress = 0 (E164)
   H225: Length of e164 = 4
   H225: E.164 = 3000
   H225:
   H225: Flags     = 02
   H225: 0... .... = No extension value(s) present in
sourceInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = Vendor is not present
```

```
      H225: ...0 .... = Gatekeeper is not present
      H225: .... 0... = Gateway is not present
      H225: .... .0.. = MCU is not present
      H225: .... ..1. = Terminal is present
      H225:
      H225: .... ...0 = No extension value(s) present in
   terminal
      H225: Flags     = 00
      H225: 0... .... = nonStandardData is not present
      H225: .0.. .... = MC = 0
      H225: ..0. .... = Undefined node = 0
      H225:
      H225: Number of destinationAddress = 2
      H225:
      H225: Flags     = 04
      H225: 0... .... = No extension value(s) present in
   destinationAddress
      H225: destinationAddress = 0 (E164)
      H225: Length of e164 = 10
      H225: E.164 = 5088623456
      H225:
Receive Alerting

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 8680
      H225:  1... ....  .... .... = Message to originator
     H225:  .000 0110  1000 0000 = Call reference value =
   1664
      H225: Message type = 1 (Alerting)
      H225:
      H225: Display
      H225: Information element identifier = 40
      H225: Display length = 7
      H225: Display flag = 4C
      H225:    0... .... = Expected extension bit
      H225: Display = "Excel"
      H225:
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 77
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 23
```

```
   H225: 0... .... = No extension values present in
H323-UserInformation
   H225: .0.. .... = user-data is not present
   H225: ..1. .... = Extension values present in h323-
uu-pdu
   H225: ...0 .... = nonStandardData is not present
   H225: .... 0... = No extension values present in
h323-message-body
   H225: h323-message = 3 (Alerting)
   H225: Flags     = 80
   H225: 1... .... = Extension values present in
Alerting-UUIE
   H225: .0.. .... = h245Address is not present
   H225: Protocol Id = {0.0.8.2250.0.2}
   H225:
   H225: Destination information
   H225: Flags     = 22
   H225: 0... .... = No extension value(s) present in
destinationInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..1. .... = Vendor is present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
vendor
   H225: Flags     = C0
   H225: 1... .... = Product ID is present
   H225: .1.. .... = Version ID is present
   H225:
   H225: ..0. .... = No extension value(s) present in
vendor
   H225: T.35 country code = 0x 9 (Australia)
   H225: T.35 extension    = 0
   H225: Manufacture code = 0x  3D (?)
   H225: Product Id = Equivalence OpenPhone<0000>
   H225: Version Id = 1.2.0<0000>
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
terminal
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = MC = 0
   H225: ...0 .... = Undefined node = 0
   H225:
   H225: .... 0... = Extension length determinant
   H225: Number of extension = 7
   H225: ...1 .... = callIdentifier is present
   H225: .... 0... = h245SecurityMode is not present
```

```
      H225: .... .0.. = tokens is not present
      H225: .... ..0. = cryptoTokens is not present
      H225: .... ...0 = fastStart is not present
      H225:
      H225: Flags     = C0
      H225: 1... .... = Extension value(s) present in
   callIdentifier
      H225: Guid = 1100000006800000068061626364313 2
      H225:
      H225: Flags     = 33
      H225: 0... .... = Extension length determinant
      H225: Number of extension = 26
      H225: .... ...1 = h4501SupplementaryService is
   present
      H225: Flags     = 34
      H225: 0... .... = h245Tunneling is not present
      H225: .0.. .... = h245Control is not present
      H225: ..1. .... = nonStandardControl is present
      H225: Number of h4501SupplementaryService = 0
      H225: Flags     = 00
      H225: 0... .... = H245 tunneling = 0
      H225: Number of nonStandardControl = 1
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   NonStandardIdentifier
      H225: nonStandardIdentifier = 0 (Object)
      H225: Data: 0 byte(s) of data

Send Route Control Ack

X->H
    [00 14 00 e8 00 02 01 00 10 01 02 1e 09 00 01 00 39 00
   03 00
     a8 08]

Receive Connect

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = 8680
      H225:  1... ....  .... .... = Message to originator
     H225:  .000 0110  1000 0000 = Call reference value =
   1664
      H225: Message type = 7 (Connect)
      H225:
      H225: Bearer Capacity
      H225: Information element identifier = 4
```

```
   H225: Length of bearer capacity = 3
   H225: Coding and capability flags = 88
   H225:                    1... .... = Expected
extension bit
   H225:                   .00. .... = Coding standard
= 0 (CCITT standardized coding)
   H225:                    ...0 1000 = Information
transfer capability = 8 (Unrestricted digital
information)
   H225: Mode and rate flags = C0
   H225:           1... .... = Expected extension bit
   H225:          .10. .... = Transfer mode = 2 (Packet
mode)
   H225:          ...0 0000 = Information transfer rate
= 0 (Used for packet mode calls)
   H225: Layer 1 protocol flag = A5
   H225:             1... .... = Expected extension bit
   H225:            .01. .... = Layer 1 identification
= 1
   H225:             ...0 0101 = Layer 1 protocol = 5
(H.221 and H.242)
   H225:
   H225: Display
   H225: Information element identifier = 40
   H225: Display length = 7
   H225: Display flag = 4C
   H225:    0... .... = Expected extension bit
   H225: Display = "Excel"
   H225:
H225:
   H225: ----User-User Information----
   H225:
   H225: Information element identifier = 126 (User-
User)
   H225: Length        = 100
   H225: Discriminator = 5 (X.208/X.209 (ASN.1))
   H225:
   H225: Flags     = 22
   H225: 0... .... = No extension values present in
H323-UserInformation
   H225: .0.. .... = user-data is not present
   H225: ..1. .... = Extension values present in h323-
uu-pdu
   H225: ...0 .... = nonStandardData is not present
   H225: .... 0... = No extension values present in
h323-message-body
   H225: h323-message = 2 (Connect)
   H225: Flags     = C0
   H225: 1... .... = Extension values present in
Connect-UUIE
   H225: .1.. .... = h245Address is present
   H225: Protocol Id = {0.0.8.2250.0.2}
```

```
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
h245Address
   H225: h245Address = 0 (IP address)
   H225: IP   = 135.119.55.170
   H225: Port = 2086
   H225:
   H225: Destination information
   H225: Flags     = 22
   H225: 0... .... = No extension value(s) present in
destinationInfo
   H225: .0.. .... = nonStandardData is not present
   H225: ..1. .... = Vendor is present
   H225: ...0 .... = Gatekeeper is not present
   H225: .... 0... = Gateway is not present
   H225: .... .0.. = MCU is not present
   H225: .... ..1. = Terminal is present
   H225:
   H225: .... ...0 = No extension value(s) present in
vendor
   H225: Flags     = C0
   H225: 1... .... = Product ID is present
   H225: .1.. .... = Version ID is present
   H225:
   H225: ..0. .... = No extension value(s) present in
vendor
   H225: T.35 country code = 0x 9 (Australia)
   H225: T.35 extension    = 0
   H225: Manufacture code = 0x  3D (?)
   H225: Product Id = Equivalence OpenPhone<0000>
   H225: Version Id = 1.2.0<0000>
   H225:
   H225: Flags     = 00
   H225: 0... .... = No extension value(s) present in
terminal
   H225: .0.. .... = nonStandardData is not present
   H225: ..0. .... = MC = 0
   H225: ...0 .... = Undefined node = 0
   H225: Conference id =
00000680000006803133353761636466
   H225:
   H225: Flags     = 0D
   H225: 0... .... = Extension length determinant
   H225: Number of extension = 7
   H225: .... ...1 = callIdentifier is present
   H225: Flags     = 0C
   H225: 0... .... = h245SecurityMode is not present
   H225: .0.. .... = tokens is not present
   H225: ..0. .... = cryptoTokens is not present
   H225: ...0 .... = fastStart is not present
```

```
      H225: Length of callIdentifier = 17
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   callIdentifier
      H225: Guid = 00000680000006806162636431323334
      H225:
      H225: Flags     = 01
      H225: 0... .... = Extension length determinant
      H225: Number of extension = 1
      H225: .... ...1 = h4501SupplementaryService is
   present
      H225: Flags     = 00
      H225: 0... .... = h245Tunneling is not present
      H225: .0.. .... = h245Control is not present
      H225: ..0. .... = nonStandardControl is not present
      H225: Number of h4501SupplementaryService = 0
      H225: Flags     = 80
      H225: 1... .... = H245 tunneling = 1
      H225:


Send Call Processing Event of Answer
X->H
    [00 0d 00 2e 00 08 01 00 01 0d 03 00 a8 08 20]


Send Request Terminal Capabilities Set
----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 330
      H245: Flags     = 02
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 2 (Terminal Capability Set)
      H245:
      H245: Flags     = 70
      H245: 0... .... = No extension value(s) present in
   terminalCapabilitySet
      H245: .1.. .... = multiplexCapability is present
      H245: ..1. .... = capabilityTable is present
      H245: ...1 .... = capabilityDescriptors is present
      H245: Sequence number = 1
      H245: Protocol id = {0.0.0.0.0.8}
      H245:
      H245: Flags     = 80
```

```
   H245: 1... .... = Extension value(s) present in
MultiplexCapability
   H245: .0.. .... = Choice value is 6 bits long
   H245: Multiplex capability = 0 (H2250 Capability)
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
h2250Capability
   H245: Maximum audio delay jitter       = 60 (ms)
   H245:
   H245: *** Receive Multipoint Capability ***
   H245:
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
receive multipoint capability
   H245: .0.. .... = Multicast capability is OFF
   H245: ..0. .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
   H245: *** Transmit Multipoint Capability ***
   H245:
   H245: .0.. .... = No extension value(s) present in
transmit multipoint capability
   H245: ..0. .... = Multicast capability is OFF
   H245: ...0 .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
```

```
   H245:
  H245: *** Receive and Transmit Multipoint Capability
***
   H245:
   H245: .0.. .... = No extension value(s) present in
receive and transmit multipoint capability
   H245: ..0. .... = Multicast capability is OFF
   H245: ...0 .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
   H245: *** MC Capability ***
   H245:
   H245: .0.. .... = No extension value(s) present in
mcCapability
   H245: ..0. .... = Centralized Conference MC is OFF
  H245: ...0 .... = Decentralized Conference MC is OFF
   H245:
   H245: .... 0... = RTCP video control capability is
OFF
   H245:
   H245: *** Media Packetization Capability ***
   H245:
   H245: .... .0.. = No extension value(s) present in
MediaPacketizationCapability
   H245: .... ..0. = H261A video packetization is OFF
   H245:
   H245: *** Capability Table ***
   H245:
   H245: Number of capability table entry = 29
   H245:
   H245: Capability Table Entry Set #1
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 1
   H245: Flags     = 28
   H245: 0... .... = No extension value(s) present in
Capability
   H245: Capability = 5 (Transmit audio capability)
```

```
   H245: .... .0.. = No extension value(s) present in
AudioCapability
   H245: Audio capability = 3 (G.711 ulaw 64k)
   H245: Value = 20
   H245:
   H245: Capability Table Entry Set #2
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 2
   H245: Flags     = 20
   H245: 0... .... = No extension value(s) present in
Capability
   H245: Capability = 4 (Receive audio capability)
   H245: .... .0.. = No extension value(s) present in
AudioCapability
   H245: Audio capability = 3 (G.711 ulaw 64k)
   H245: Value = 20
   H245:
   H245: Capability Table Entry Set #3
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 3
   H245: Flags     = 28
   H245: 0... .... = No extension value(s) present in
Capability
   H245: Capability = 5 (Transmit audio capability)
   H245: .... .0.. = No extension value(s) present in
AudioCapability
   H245: Audio capability = 1 (G.711 alaw 64k)
   H245: Value = 20
   H245:
   H245: Capability Table Entry Set #4
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 4
   H245: Flags     = 20
   H245: 0... .... = No extension value(s) present in
Capability
   H245: Capability = 4 (Receive audio capability)
   H245: .... .0.. = No extension value(s) present in
AudioCapability
   H245: Audio capability = 1 (G.711 alaw 64k)
   H245: Value = 20
   H245:
   H245: Capability Table Entry Set #5
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 5
   H245: Flags     = 2A
   H245: 0... .... = No extension value(s) present in
Capability
```

```
Receive Request Terminal Capabilities Set

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 194
      H245: Flags     = 02
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 2 (Terminal Capability Set)
      H245:
      H245: Flags     = 70
      H245: 0... .... = No extension value(s) present in
   terminalCapabilitySet
      H245: .1.. .... = multiplexCapability is present
      H245: ..1. .... = capabilityTable is present
      H245: ...1 .... = capabilityDescriptors is present
      H245: Sequence number = 1
      H245: Protocol id = {0.0.8.245.0.3}
      H245:
      H245: Flags     = 80
      H245: 1... .... = Extension value(s) present in
   MultiplexCapability
      H245: .0.. .... = Choice value is 6 bits long
      H245: Multiplex capability = 0 (H2250 Capability)
      H245: Flags     = 80
      H245: 1... .... = Extension value(s) present in
   h2250Capability
      H245: Maximum audio delay jitter       = 50 (ms)
      H245:
      H245: *** Receive Multipoint Capability ***
      H245:
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   receive multipoint capability
      H245: .0.. .... = Multicast capability is OFF
      H245: ..0. .... = Multi unicast conference is OFF
      H245:
      H245: Media Distribution Capability #1
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   mediaDistributionCapability
      H245: .0.. .... = centralizedData is not present
      H245: ..0. .... = distributedData is not present
      H245: ...0 .... = Centralized control is OFF
      H245: .... 0... = Distributed control is OFF
      H245: .... .0.. = Centralized audio is OFF
```

```
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
   H245: *** Transmit Multipoint Capability ***
   H245:
   H245: .0.. .... = No extension value(s) present in
transmit multipoint capability
   H245: ..0. .... = Multicast capability is OFF
   H245: ...0 .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
  H245: *** Receive and Transmit Multipoint Capability
***
   H245:
   H245: .0.. .... = No extension value(s) present in
receive and transmit multipoint capability
   H245: ..0. .... = Multicast capability is OFF
   H245: ...0 .... = Multi unicast conference is OFF
   H245:
   H245: Media Distribution Capability #1
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaDistributionCapability
   H245: .0.. .... = centralizedData is not present
   H245: ..0. .... = distributedData is not present
   H245: ...0 .... = Centralized control is OFF
   H245: .... 0... = Distributed control is OFF
   H245: .... .0.. = Centralized audio is OFF
   H245: .... ..0. = Distributed audio is OFF
   H245: .... ...0 = Centralized video is OFF
   H245: Flags     = 00
   H245: 0... .... = Distributed video is OFF
   H245:
   H245: *** MC Capability ***
   H245:
```

```
   H245: .0.. .... = No extension value(s) present in
mcCapability
   H245: ..0. .... = Centralized Conference MC is OFF
   H245: ...0 .... = Decentralized Conference MC is OFF
   H245:
   H245: .... 0... = RTCP video control capability is
OFF
   H245:
   H245: *** Media Packetization Capability ***
   H245:
   H245: .... .0.. = No extension value(s) present in
MediaPacketizationCapability
   H245: .... ..0. = H261A video packetization is OFF
   H245: .... ...0 = Extension index format
   H245: Number of extension = 4
   H245: .... ..0. = transportCapability is not present
   H245: .... ...0 = redundancyEncodingCapability is
not present
   H245: Flags     = C0
   H245: 1... .... = logicalChannelSwitchingCapability
is present
   H245: .1.. .... = t120DynamicPortCapability is
present
   H245: Flags     = 00
   H245: 0... .... = Logical channel switching
capability is OFF
   H245: Flags     = 00
   H245: 0... .... = T120 dynamic port capability is
OFF
   H245:
   H245: *** Capability Table ***
   H245:
   H245: Number of capability table entry = 10
   H245:
   H245: Capability Table Entry Set #1
   H245: Flags     = 80
   H245: 1... .... = capability is present
   H245: Capability table entry number = 1
   H245: Flags     = 20
   H245: 0... .... = No extension value(s) present in
Capability
   H245: Capability = 4 (Receive audio capability)
   H245: .... .0.. = No extension value(s) present in
AudioCapability
   H245: Audio capability = 0 (Non standard)
   H245: nonStandardIdentifier = 1 (H221 Non Standard)
   H245: T.35 country code = 0x 9 (Australia)
   H245: T.35 extension    = 0
   H245: Manufacture code = 0x  3D (?)
   H245: Data: 6 byte(s) of data
   H245:
```

```
      H245: Capability Table Entry Set #2
      H245: Flags     = 80
      H245: 1... .... = capability is present
      H245: Capability table entry number = 2
      H245: Flags     = 24
      H245: 0... .... = No extension value(s) present in
   Capability
      H245: Capability = 4 (Receive audio capability)
      H245: .... .1.. = Extension value(s) present in
   AudioCapability
      H245: .... ..0. = Extension index format
      H245: Audio capability = 3 (GSM full rate)
      H245: .... .0.. = No extension value(s) present in
   GSMAudioCapability
      H245: Audio unit size = 4
      H245: Flags     = 00
      H245: 0... .... = Confort noise
      H245: .0.. .... = Scrambled
      H245:
      H245: Capability Table Entry Set #3
      H245: ..0. .... = capability is not present
      H245: Capability table entry number = 58881
      H245:
      H245: Capability Table Entry Set #4
      H245: Flags     = 80
      H245: 1... .... = capability is present
      H245: Capability table entry number = 3

Send Request Master Slave Determination

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 11
      H245: Flags     = 01
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 1 (Master Slave Determination)
      H245:
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   MasterSlaveDetermination
      H245: Terminal type = 50
      H245: Status determination number = 16761311

Receive Request Master Slave Determination

----- Control Protocol for Multimedia Communication -----
```

```
      H245:
      H245: Message length = 11
      H245: Flags     = 01
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 1 (Master Slave Determination)
      H245:
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   MasterSlaveDetermination
      H245: Terminal type = 50
      H245: Status determination number = 13961701


Send Response Terminal Capabilities Set
----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 7
      H245: Flags     = 21
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 3 (Terminal Capability Set Ack)
      H245: .0.. .... = No extension value(s) present in
   TerminalCapabilitySetAck
      H245: Sequence number = 1


Receive Response Terminal Capabilities Set

----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 7
      H245: Flags     = 21
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 3 (Terminal Capability Set Ack)
      H245: .0.. .... = No extension value(s) present in
   TerminalCapabilitySetAck
      H245: Sequence number = 1
```

```
Send Response Master Slave Determination
----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 6
      H245: Flags     = 20
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 1 (Master Slave Determination Ack)
      H245: .0.. .... = No extension value(s) present in
   MasterSlaveDeterminationAck
      H245: Decision  = 0 (Master)


Receive Response Master Slave Determination
----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 6
      H245: Flags     = 20
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 1 (Master Slave Determination Ack)
      H245: .0.. .... = No extension value(s) present in
   MasterSlaveDeterminationAck
      H245: Decision  = 1 (Slave)


Receive Request Open Logical Channel
----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 24
      H245: Flags     = 03
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 3 (Open Logical Channel)
      H245:
      H245:
      H245: Flags     = 00
```

```
   H245: 0... .... = No extension value(s) present in
OpenLogicalChannel
  H245: .0.. .... = reverseLogicalChannelParameters is
not present
   H245: Forward logical channel number = 100
   H245:
   H245: Flags     = 0C
   H245: 0... .... = No extension value(s) present in
forwardLogicalChannelParameters
   H245: .0.. .... = portNumber is not present
   H245:
   H245: ..0. .... = No extension value(s) present in
dataType
   H245: Data type = 3 (Audio data)
   H245: .... ..0. = No extension value(s) present in
audioData
   H245: Audio capability = 3 (G.711 ulaw 64k)
   H245: Value = 20
   H245:
   H245: Flags     = 80
   H245: 1... .... = Extension value(s) present in
MultiplexParameters
   H245: .0.. .... = Extension index format
  H245: Multiplex parameters = 0 (H225 logical channel
parameters)
   H245:
   H245: Flags     = 0D
   H245: 0... .... = No extension value(s) present in
h2250LogicalChannelParameters
   H245: .0.. .... = nonStandard is not present
  H245: ..0. .... = associatedSessionID is not present
   H245: ...0 .... = mediaChannel is not present
  H245: .... 1... = mediaGuaranteedDelivery is present
   H245: .... .1.. = mediaControlChannel is present
  H245: .... ..0. = mediaControlGuaranteedDelivery is
not present
   H245: .... ...1 = silenceSuppression is present
   H245: Flags     = 00
   H245: 0... .... = destination is not present
   H245: .0.. .... = dynamicRTPPayloadType is not
present
   H245: ..0. .... = mediaPacketization is not present
   H245: Session ID = 1
   H245: Flags     = 00
   H245: 0... .... = Media guaranted delivery is OFF
   H245:
   H245: *** Media Control Channel ***
   H245: .0.. .... = No extension value(s) present in
mediaControlChannel(RTCP)
   H245: Transport Address = 0 (Unicast address)
```

```
      H245: ...0 .... = No extension value(s) present in
   UnicastAddress
      H245: Unicast address = 0 (IP address)
      H245: .... ...0 = No extension value(s) present in
   iPAddress
      H245: Network = 135.119.55.170
      H245: TSAP indentifier = 5001
      H245: Flags     = 80
      H245: 1... .... = Silence suppression is ON

Send Response Open Logical Channel
----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 27
      H245: Flags     = 22
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 1 (Response)
      H245:
      H245: ...0 .... = No extension value(s) present in
   ResponseMessage
      H245: Response  = 5 (Open Logical Channel Ack)
      H245:
      H245: .1.. .... = Extension value(s) present in
   openLogicalChannelAck
      H245: ..0. .... = reverseLogicalChannelParameters is
   not present
      H245: Forward logical channel number = 100
      H245:
      H245: Flags     = 04
      H245: 0... .... = Extension index format
      H245: Number of extension = 3
      H245: .... ...0 = separateStack is not present
      H245: Flags     = 80
      H245: 1... .... = forwardMultiplexAckParameters is
   present
      H245: .0.. .... = encryptionSync is not present
      H245: Flags     = 1C
      H245: 0... .... = No extension value(s) present in
   forwardMultiplexAckParameters
      H245: .0.. .... = No extension value(s) present in
   H2250LogicalChannelAckParameters
      H245: ..0. .... = nonStandard is not present
      H245: ...1 .... = sessionID is present
      H245: .... 1... = mediaChannel is present
      H245: .... .1.. = mediaControlChannel is present
      H245: .... ..0. = dynamicRTPPayloadType is not
   present
      H245: Session ID = 1
      H245: .... ...0 = No extension value(s) present in
   mediaChannel
```

```
      H245: Transport Address = 0 (Unicast address)
      H245: .0.. .... = No extension value(s) present in
   UnicastAddress
      H245: Unicast address = 0 (IP address)
      H245: .... .0.. = No extension value(s) present in
   iPAddress
      H245: Network = 135.119.55.183
      H245: TSAP indentifier = 14656
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   mediaControlChannel (forward RTCP channel)
      H245: Transport Address = 0 (Unicast address)
      H245: ..0. .... = No extension value(s) present in
   UnicastAddress
      H245: Unicast address = 0 (IP address)
      H245: .... ..0. = No extension value(s) present in
   iPAddress
      H245: Network = 135.119.55.183
      H245: TSAP indentifier = 14657


Send Request Open Logical Channel
----- Control Protocol for Multimedia Communication -----
      H245:
      H245: Message length = 24
      H245: Flags     = 03
      H245: 0... .... = No extension value(s) present in
   H.245 control message
      H245: H.245 call control message type = 0 (Request)
      H245:
      H245: ...0 .... = No extension value(s) present in
   request
      H245: Request type = 3 (Open Logical Channel)
      H245:
      H245:
      H245: Flags     = 00
      H245: 0... .... = No extension value(s) present in
   OpenLogicalChannel
        H245: .0.. .... = reverseLogicalChannelParameters is
   not present
      H245: Forward logical channel number = 1763
      H245:
      H245: Flags     = 0C
      H245: 0... .... = No extension value(s) present in
   forwardLogicalChannelParameters
      H245: .0.. .... = portNumber is not present
      H245:
      H245: ..0. .... = No extension value(s) present in
   dataType
      H245: Data type = 3 (Audio data)
```

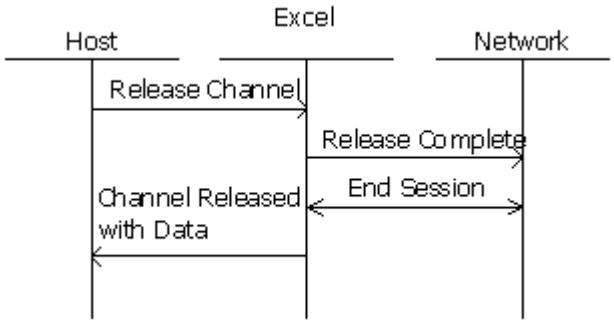
```
   H245: .... ..0. = No extension value(s) present in
audioData
   H245: Audio capability = 3 (G.711 ulaw 64k)
   H245: Value = 20
   H245:
   H245: Flags     = 80
   H245: 1... .... = Extension value(s) present in
MultiplexParameters
   H245: .0.. .... = Extension index format
  H245: Multiplex parameters = 0 (H225 logical channel
parameters)
   H245:
   H245: Flags     = 05
   H245: 0... .... = No extension value(s) present in
h2250LogicalChannelParameters
   H245: .0.. .... = nonStandard is not present
  H245: ..0. .... = associatedSessionID is not present
   H245: ...0 .... = mediaChannel is not present
   H245: .... 0... = mediaGuaranteedDelivery is not
present
   H245: .... .1.. = mediaControlChannel is present
  H245: .... ..0. = mediaControlGuaranteedDelivery is
not present
   H245: .... ...1 = silenceSuppression is present
   H245: Flags     = 00
   H245: 0... .... = destination is not present
   H245: .0.. .... = dynamicRTPPayloadType is not
present
   H245: ..0. .... = mediaPacketization is not present
   H245: Session ID = 1
   H245:
   H245: *** Media Control Channel ***
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaControlChannel(RTCP)
   H245: Transport Address = 0 (Unicast address)
   H245: ..0. .... = No extension value(s) present in
UnicastAddress
   H245: Unicast address = 0 (IP address)
   H245: .... ..0. = No extension value(s) present in
iPAddress
   H245: Network = 135.119.55.183
   H245: TSAP indentifier = 14657
   H245: Flags     = 00
   H245: 0... .... = Silence suppression is OFF


Receive Response Open Logical Channel

----- Control Protocol for Multimedia Communication -----
      H245:
```

```
   H245: Message length = 30
   H245: Flags     = 22
   H245: 0... .... = No extension value(s) present in
H.245 control message
   H245: H.245 call control message type = 1 (Response)
   H245:
   H245: ...0 .... = No extension value(s) present in
ResponseMessage
   H245: Response  = 5 (Open Logical Channel Ack)
   H245:
   H245: .1.. .... = Extension value(s) present in
openLogicalChannelAck
  H245: ..0. .... = reverseLogicalChannelParameters is
not present
   H245: Forward logical channel number = 1763
   H245:
   H245: Flags     = 02
   H245: 0... .... = Extension index format
   H245: Number of extension = 2
   H245: .... ...0 = separateStack is not present
   H245: Flags     = 80
   H245: 1... .... = forwardMultiplexAckParameters is
present
   H245: .0.. .... = encryptionSync is not present
   H245: Flags     = 5C
   H245: 0... .... = No extension value(s) present in
forwardMultiplexAckParameters
   H245: .1.. .... = Extension value(s) present in
H2250LogicalChannelAckParameters
   H245: ..0. .... = nonStandard is not present
   H245: ...1 .... = sessionID is present
   H245: .... 1... = mediaChannel is present
   H245: .... .1.. = mediaControlChannel is present
   H245: .... ..0. = dynamicRTPPayloadType is not
present
   H245: Session ID = 1
   H245: .... ...0 = No extension value(s) present in
mediaChannel
   H245: Transport Address = 0 (Unicast address)
   H245: .0.. .... = No extension value(s) present in
UnicastAddress
   H245: Unicast address = 0 (IP address)
   H245: .... .0.. = No extension value(s) present in
iPAddress
   H245: Network = 135.119.55.170
   H245: TSAP indentifier = 5000
   H245: Flags     = 00
   H245: 0... .... = No extension value(s) present in
mediaControlChannel (forward RTCP channel)
   H245: Transport Address = 0 (Unicast address)
   H245: ..0. .... = No extension value(s) present in
UnicastAddress
```

```
   H245: Unicast address = 0 (IP address)
   H245: .... ..0. = No extension value(s) present in
iPAddress
   H245: Network = 135.119.55.170
   H245: TSAP indentifier = 5001
   H245:
   H245: Flags     = 01
   H245: 0... .... = Extension index format
   H245: Number of extension = 1
   H245: .... ...1 = flowControlToZero is present
   H245: Flags     = 00
   H245: 0... .... = Flow control to zero is OFF
```

Call Release Call Flows

Host Initiated Call Release



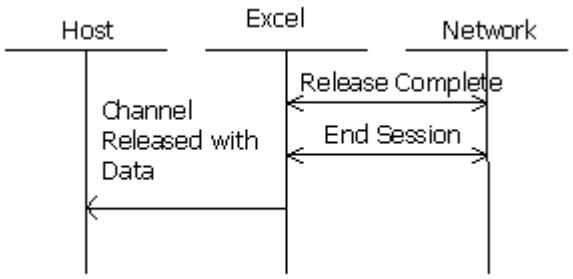
Message Trace

```
Receive Release Channel
H->X
    [00 11 00 08 00 00 01 00 02 0d 03 00 a8 14 0d 03 00 00
     01]

X->H
    [00 07 00 08 00 00 01 00 10]


Send Release Complete

----- H.225 Call Signaling -----
      H225:
      H225: Protocol discriminator   = 8
      H225: Length of call reference = 2
      H225: Call reference field = FCD8
      H225:  1... ....  .... .... = Message to originator
     H225:  .111 1100  1101 1000 = Call reference value =
   31960
      H225: Message type = 90 (Release Complete)
      H225:
      H225:
```

```
H225:
      H225: ----User-User Information----
      H225:
      H225: Information element identifier = 126 (User-
   User)
      H225: Length        = 31
      H225: Discriminator = 5 (X.208/X.209 (ASN.1))
      H225:
      H225: Flags     = 05
      H225: 0... .... = No extension values present in
   H323-UserInformation
      H225: .0.. .... = user-data is not present
      H225: ..0. .... = No extension values present in
   h323-uu-pdu
      H225: ...0 .... = nonStandardData is not present
      H225: .... 0... = No extension values present in
   h323-message-body
      H225: h323-message = 5 (Release complete)
      H225: Flags     = C0
      H225: 1... .... = Extension values present in
   ReleaseComplete-UUIE
      H225: .1.. .... = ReleaseCompleteReason is present
      H225: Protocol id = {0.0.8.2250.0.2}
      H225: Reason = 0 (No bandwidth)
      H225: .1.. .... = Extension length determinant
      H225: Number of extension = 25
      H225: Flags     = 08
      H225: 0... .... = callIdentifier is not present
      H225: Length of callIdentifier = 17
      H225:
      H225: Flags     = 00
      H225: 0... .... = No extension value(s) present in
   callIdentifier
      H225: Guid = 5D0D95950952D61198CE0050DAD08480
      H225:


Receive End Session
H245 Msg RCVD
4a 40


Send End Session
H245 Msg SENT
4a 40


Send Channel Released with Data

X->H
```

```
 [00 5d 00 69 00 14 01 00 01 0d 03 00 a8 14 02 02 1e 2a
00 05
  01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 00 01 11
  00 04 00 00 00 de 01 10 00 04 00 00 00 00 01 12 00 04
  00 00 0a f8 03 00 33 00 1e 00 04 27 4e 00 02 00 10 27
  92 00 04 87 77 37 b7 27 93 00 04 00 00 3c 40 27 e3 00
   02 03 07]
```

Network Initiated Call Release



Message Trace

```
Receive Release Complete

--- Q.931 (CCITT Q.931 Call Control)---

 Protocol Discriminator: 8 [0x08] Q.931 User-Network Call
   Control Msgs
 Call Ref Value Length: 2 bytes
 Call Reference Field: 0x7CDB
   :  0... .... : Flag=0, Origination Side of CRV
   :  .101 1011 : CRV=31963, [0x7CDB]
 Message Type: Release Complete
 Information Element: 126, User-User
```

```
Length of Info Element: 30 bytes
Octet 3: 0x05
  :  0000 0101 : Protocol = X.208/X.209 (ASN.1)

--- H.323 (ITU-T H.323 User Information)---

H323-UserInformation:
  h323-uu-pdu:
    h323-message-body:
      releaseComplete:
        protocolIdentifier: itu-t
  : recommendation
  : h
  : 2250
  : version
  : 2
        reason: undefinedReason
        callIdentifier:
          guid: Hex( 63 0D 95 95 09 52 D6 11 98 CE 00 50
  DA D0 84 80 )
  : [end of pdu]
  :



Send Release Complete

--- Q.931 (CCITT Q.931 Call Control)---

Protocol Discriminator: 8 [0x08] Q.931 User-Network Call
  Control Msgs
Call Ref Value Length: 2 bytes
Call Reference Field: 0x7CDA
  :  0... .... : Flag=0, Origination Side of CRV
  :  .101 1010 : CRV=31962, [0x7CDA]
Message Type: Release Complete
Information Element: 8, Cause
Length of Info Element: 3 bytes
Octet 3: 0x00
  :  0... .... : Extension Bit = 0
  :  .00. .... : Coding Standard = ITU-T
  :  ...0 .... : Spare Bit
  :  .... 0000 : Location = User
Octet 3a: 0x00
  :  0... .... : Extension Bit = 0 (Invalid Ext Bit)
  :  .000 0000 : Recommendation = Q.931
Octet 4: 0x90
  :  1... .... : Extension Bit = 1
  :  .001 .... : Cause Class = Normal Event
  :  .... 0000 : Cause Code = Normal Call Clearing
```

```
  :
Information Element: 126, User-User
Length of Info Element: 30 bytes
Octet 3: 0x05
  :  0000 0101 : Protocol = X.208/X.209 (ASN.1)

--- H.323 (ITU-T H.323 User Information)---

H323-UserInformation:
  h323-uu-pdu:
    h323-message-body:
      releaseComplete:
        protocolIdentifier: itu-t
  : recommendation
  : h
  : 2250
  : version
  : 2
        reason: undefinedReason
        callIdentifier:
          guid: Hex( 61 0D 95 95 09 52 D6 11 98 CE 00 50
  DA D0 84 80 )
  : [end of pdu]
  :


Receive End Session

H245 Msg RCVD
4a 40

Send End Session
H245 Msg SENT
4a 40

Send Channel Release with Data
X->H
    [00 5d 00 69 00 17 01 00 01 0d 03 00 a8 16 02 02 1e 2a
   00 05
     01 04 00 04 00 00 00 00 01 05 00 04 00 00 00 00 01 11
     00 04 00 00 1e 9f 01 10 00 04 00 00 00 00 01 12 00 04
     00 02 22 8c 03 00 33 00 1e 00 04 27 4e 00 02 00 10 27
     92 00 04 87 77 37 b7 27 93 00 04 00 00 3c 80 27 e3 00
     02 02 0c]
```