



Dialogic® NaturalAccess™ Video Access Toolkit 3G-324M over IP Developer's Manual

Copyright and legal notices

Copyright © 2008-2010 Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies 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 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.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

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 may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does 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 and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Dialogic 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.

Any use case(s) shown and/or described herein represent one or more examples of the various ways, scenarios or environments in which Dialogic® products can be used. Such use case(s) are non-limiting and do not represent recommendations of Dialogic as to whether or how to use Dialogic products.

Dialogic, Dialogic Pro, Brooktrout, Diva, Cantata, SnowShore, Eicon, Eicon Networks, NMS Communications, NMS (stylized), Eiconcard, SIPcontrol, Diva ISDN, TruFax, Exnet, EXS, SwitchKit, N20, Making Innovation Thrive, Connecting to Growth, Video is the New Voice, Fusion, Vision, PacketMedia, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation or its subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. The names of actual companies and product mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Revision history

Revision	Release date	Notes
9000-62722-01	September 2008	PLR, See original document for dates and revision notes.
64-1132-01 Rev A	October 2010	LBG, Video Access 3.3
Last modified: 2010-10-15		

Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

Table of Contents

1. Introduction.....	6
2. 3G-324M Over IP Overview.....	9
Network Architecture	9
System Requirements	10
BICC Call Control and IP Bearer Control Overview	10
Call Flow: Forward Bearer Establishment with Fast Forward Tunneling.....	11
Call Flow: Forward Bearer Establishment with Delayed Forward Tunneling	12
3G-324M over IP Components.....	13
3. Bearer Control Function Library	14
Bearer Control Function Library Overview.....	14
Establishing a 3G-324M over IP Session	15
BCF Call Flows	16
Forward Bearer Establishment with Fast Forward Tunneling	16
Forward Bearer Establishment with Delayed Forward Tunneling.....	17
BCF Library Limitations	18
Building a Bearer Control Message.....	19
Initialize the Bearer Control Event structure.....	19
Build the Bearer Control Message	21
Parsing a Bearer Control Message.....	21
4. BCF Library Functions	22
BCF Library Function Summary	22
bcfBuildBearerControlMessage	22
bcfInitFastForwardBearerSetupRequest	23
bcfInitFastForwardBearerSetupAnswer	24
bcfInitDelayedForwardBearerInfoRequest	26
bcfInitDelayedForwardBearerInfoAnswer	28
bcfInitDelayedForwardBearerSetupRequest.....	30
bcfInitDelayedForwardBearerSetupAnswer.....	31
bcfParseBearerControlMessage.....	32
bcfInitIpbcParam	33
5. Bearer Control Event Structures.....	34
BCF Library Structures	34
SiBearerControlEvt Structure.....	34
SiBearerControlInformation Structure	34
SiActionIndicator Structure	34
SiBearerNetworkCharacteristics Structure.....	34
SiBearerControlTunnelling Structure	34
SiBearerControlUnitIdentifier Structure	35
SiAmrConfig Structure.....	35
SiAmrCodec Structure	35
SiG711Codec Structure	35
SiMuMeCodec Structure.....	35
SiCodecs Structure	35
SiBackboneNetworkIdentifier Structure	35
SiInterworkingFunctionAddress Structure	36
IPBCP_PARAM Structure	36
6. Demonstration Program.....	37
3G-324M Over IP Architecture.....	37

vmsamp_ip324m_bicc Limitations	38
3G-324M Parameters	38
H.223 Gateway-Related Settings for vmsamp_ip324m_bicc	39
NbUP-Related Settings for vmsamp_ip324m_bicc	40
ISUP-Related Settings for vmsamp_ip324m_bicc	41
BCF-Related Settings for vmsamp_ip324m_bicc	42
Configuration Overview	43
Configuring the Software	44
Configuration Procedure	45
Running vmsamp_ip324m_bicc	48
NbUP RTP Endpoint statistics	49

1. Introduction

The *Dialogic® NaturalAccess™ Video Access Toolkit 3G-324M over IP Developer's Manual* describes how to use, install, configure, and run extensions to Video Access 3.3 that enables the 3G-324M over IP feature. The 3G-324M over IP feature supports the following functionality:

- H223 framed with NbUP over RTP for the bearer data
- BICC and Bearer Control Functions for signaling

BICC and NbUP are typically used in a 3GPP Release 4 (or later) core network. By supporting these protocols, you can terminate calls in the core IP network and use Video Access to develop advanced multimedia applications such as Video RingBack tone and IVVR.

Terminology

Note: The product to which this document pertains is part of the NMS Communications Platforms business that was sold by NMS Communications Corporation ("NMS") to Dialogic Corporation ("Dialogic") on December 8, 2008. Accordingly, certain terminology relating to the product has been changed. Below is a table indicating both terminology that was formerly associated with the product, as well as the new terminology by which the product is now known. This document is being published during a transition period; therefore, it may be that some of the former terminology will appear within the document, in which case the former terminology should be equated to the new terminology, and vice versa.

Former terminology	Dialogic terminology
CG 6060 Board	Dialogic® CG 6060 PCI Media Board
CG 6060C Board	Dialogic® CG 6060C CompactPCI Media Board
CG 6565 Board	Dialogic® CG 6565 PCI Media Board
CG 6565C Board	Dialogic® CG 6565C CompactPCI Media Board
CG 6565e Board	Dialogic® CG 6565E PCI Express Media Board
CX 2000 Board	Dialogic® CX 2000 PCI Station Interface Board
CX 2000C Board	Dialogic® CX 2000C CompactPCI Station Interface Board
AG 2000 Board	Dialogic® AG 2000 PCI Media Board
AG 2000C Board	Dialogic® AG 2000C CompactPCI Media Board
AG 2000-BRI Board	Dialogic® AG 2000-BRI Media Board
NMS OAM Service	Dialogic® NaturalAccess™ OAM API

Former terminology	Dialogic terminology
NMS OAM System	Dialogic® NaturalAccess™ OAM System
NMS SNMP	Dialogic® NaturalAccess™ SNMP API
Natural Access	Dialogic® NaturalAccess™ Software
Natural Access Service	Dialogic® NaturalAccess™ Service
Fusion	Dialogic® NaturalAccess™ Fusion™ VoIP API
ADI Service	Dialogic® NaturalAccess™ Alliance Device Interface API
CDI Service	Dialogic® NaturalAccess™ CX Device Interface API
Digital Trunk Monitor Service	Dialogic® NaturalAccess™ Digital Trunk Monitoring API
MSPP Service	Dialogic® NaturalAccess™ Media Stream Protocol Processing API
Natural Call Control Service	Dialogic® NaturalAccess™ NaturalCallControl™ API
NMS GR303 and V5 Libraries	Dialogic® NaturalAccess™ GR303 and V5 Libraries
Point-to-Point Switching Service	Dialogic® NaturalAccess™ Point-to-Point Switching API
Switching Service	Dialogic® NaturalAccess™ Switching Interface API
Voice Message Service	Dialogic® NaturalAccess™ Voice Control Element API
NMS CAS for Natural Call Control	Dialogic® NaturalAccess™ CAS API
NMS ISDN	Dialogic® NaturalAccess™ ISDN API
NMS ISDN for Natural Call Control	Dialogic® NaturalAccess™ ISDN API
NMS ISDN Messaging API	Dialogic® NaturalAccess™ ISDN Messaging API
NMS ISDN Supplementary Services	Dialogic® NaturalAccess™ ISDN API Supplementary Services
NMS ISDN Management API	Dialogic® NaturalAccess™ ISDN Management API

Former terminology	Dialogic terminology
NaturalConference Service	Dialogic® NaturalAccess™ NaturalConference™ API
NaturalFax	Dialogic® NaturalAccess™ NaturalFax™ API
SAI Service	Dialogic® NaturalAccess™ Universal Speech Access API
NMS SIP for Natural Call Control	Dialogic® NaturalAccess™ SIP API
NMS RJ-45 interface	Dialogic® MD1 RJ-45 interface
NMS RJ-21 interface	Dialogic® MD1 RJ-21 interface
NMS Mini RJ-21 interface	Dialogic® MD1 Mini RJ-21 interface
NMS Mini RJ-21 to NMS RJ-21 cable	Dialogic® MD1 Mini RJ-21 to MD1 RJ-21 cable
NMS RJ-45 to two 75 ohm BNC splitter cable	Dialogic® MD1 RJ-45 to two 75 ohm BNC splitter cable
NMS signal entry panel	Dialogic® Signal Entry Panel
Video Access Utilities	Dialogic® NaturalAccess™ Video Access Toolkit Utilities
Video Mail Application Demonstration Program	Dialogic® NaturalAccess™ Video Access Toolkit Video Mail Application Demonstration Program
Video Messaging Server Interface	Dialogic® NaturalAccess™ Video Access Toolkit Video Messaging Server Interface
3G-324M Interface	Dialogic® NaturalAccess™ Video Access Toolkit 3G-324M Interface

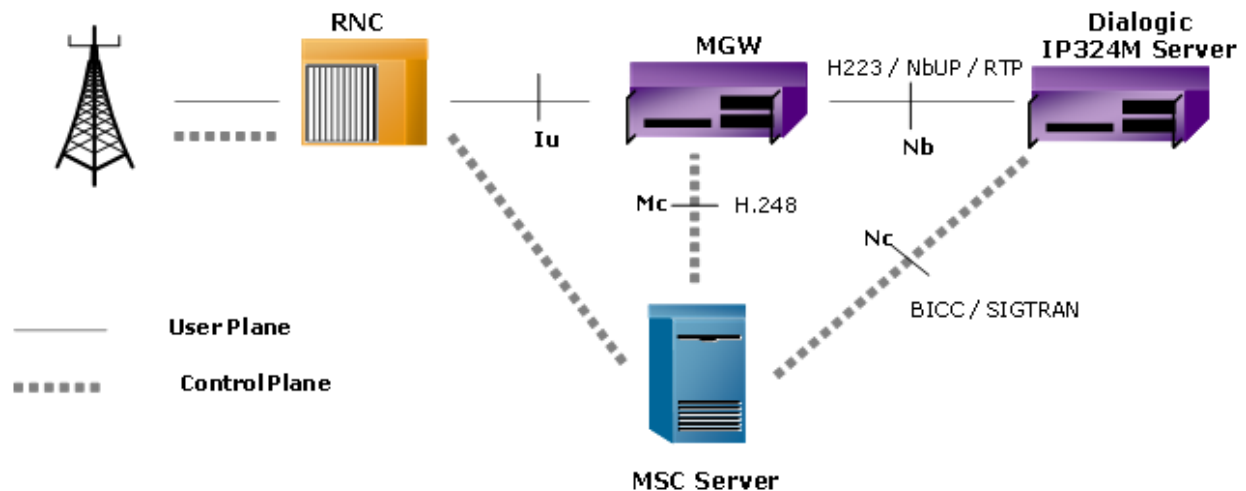
2. 3G-324M Over IP Overview

Network Architecture

The 3G-324M over IP feature provides the ability to transmit and receive H.223 over IP data streams in Video Access, using NbUP as a framing protocol. This functionality is provided through the introduction of several new components, integrated in NaturalAccess and Video Access frameworks:

- NbUP Endpoint and Clear Channel are added to NaturalAccess MSPP Service.
- BICC Call Control Signaling is performed through a TX 4000 board and functions.
- A BCF library is provided for IP Bearer Control on top of BICC.
- A new version of Video Access video mail sample application (*vmsamp_ip324m_bicc*) is provided to demonstrate these components.

The following illustration shows where a server based on the Video Access Toolkit 3G-324M over IP technology could be connected in a 3GPP Release4 core Network:



With Video Access Toolkit 3G-324M over IP technology, you can build servers that terminate:

- The Nb Interface: H223 framed with NbUP over RTP bearer
- The Nc Interface: BICC Call Control and BICC Bearer Control over SIGTRAN

Note: An IP324M Server based on Dialogic technology could also be configured as a gateway between the 3GP Release 4 BICC network and a SIP network.

System Requirements

Video Access Toolkit 3G-324M over IP requires a system with at least a CG 6565 board and a TX 4000 board.

The following software products are also required:

- Dialogic® NaturalAccess™ Development Environment R9.0.4 or later
- Dialogic® NaturalAccess™ Signaling Software 5.3 or later (be sure to install the SIGTRAN feature)

BICC Call Control and IP Bearer Control Overview

The NaturalAccess BICC implementation included with Dialogic® NaturalAccess™ Signaling Software supports:

- BICC Call Control, as an ISUP Call Control variant
- BICC APM Transport Mechanism

For more information, refer to the *Dialogic® NaturalAccess™ ISUP Layer Developer's Reference Manual*.

To perform IP Bearer control, BICC messages (Call Control or APM) are used to tunnel the IP bearer-related information, such as Bearer Characteristics, Codec, and IPBCP. IPBCP (IP Bearer Control Protocol) is a protocol based on SDP, and is used to exchange Bearer IP Addresses, RTP port numbers, and Packetization Time.

To format and parse Bearer Control Messages, a new library called BCF (Bearer Control Functions) is provided in the Video Access 3.3 package.

Two Bearer Control Signaling Call Flows are implemented by *vmsamp_ip324m_bicc* and demonstrate the usage of the BCF library:

- Forward bearer establishment with fast forward tunneling
- Forward bearer establishment with delayed forward tunneling (and optional codec negotiation)

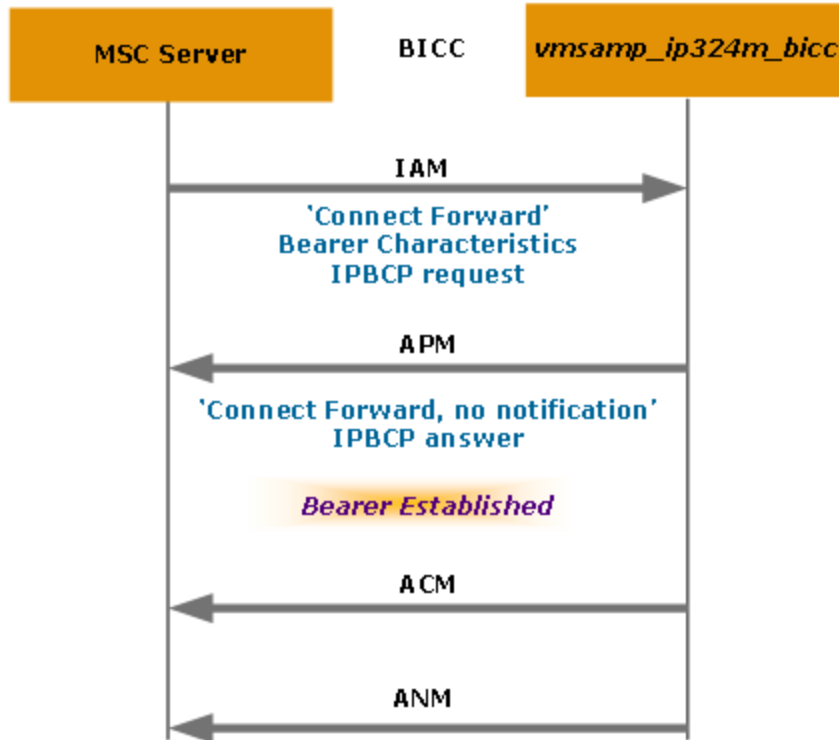
These two call flows are commonly used call flows in a 3GPP BICC IP Core Network.

Several other call flows are possible, but are not implemented in *vmsamp_ip324m_bicc*, even though the BCF library would allow implementing them.

In particular, backward bearer establishment call flows are not implemented by *vmsamp_ip324m_bicc*. Scenarios involving COT message are also not implemented.

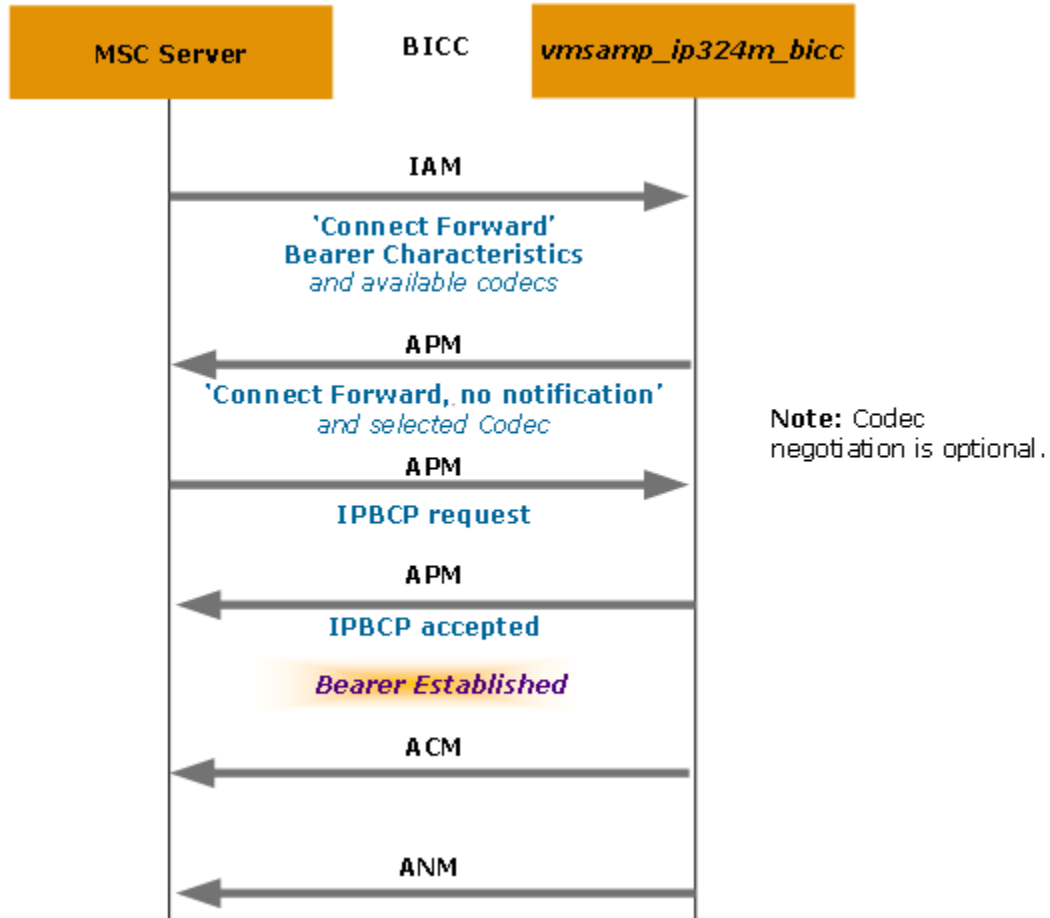
Call Flow: Forward Bearer Establishment with Fast Forward Tunneling

The following illustration shows forward bearer establishment with fast forward tunneling Call Flow:



Call Flow: Forward Bearer Establishment with Delayed Forward Tunneling

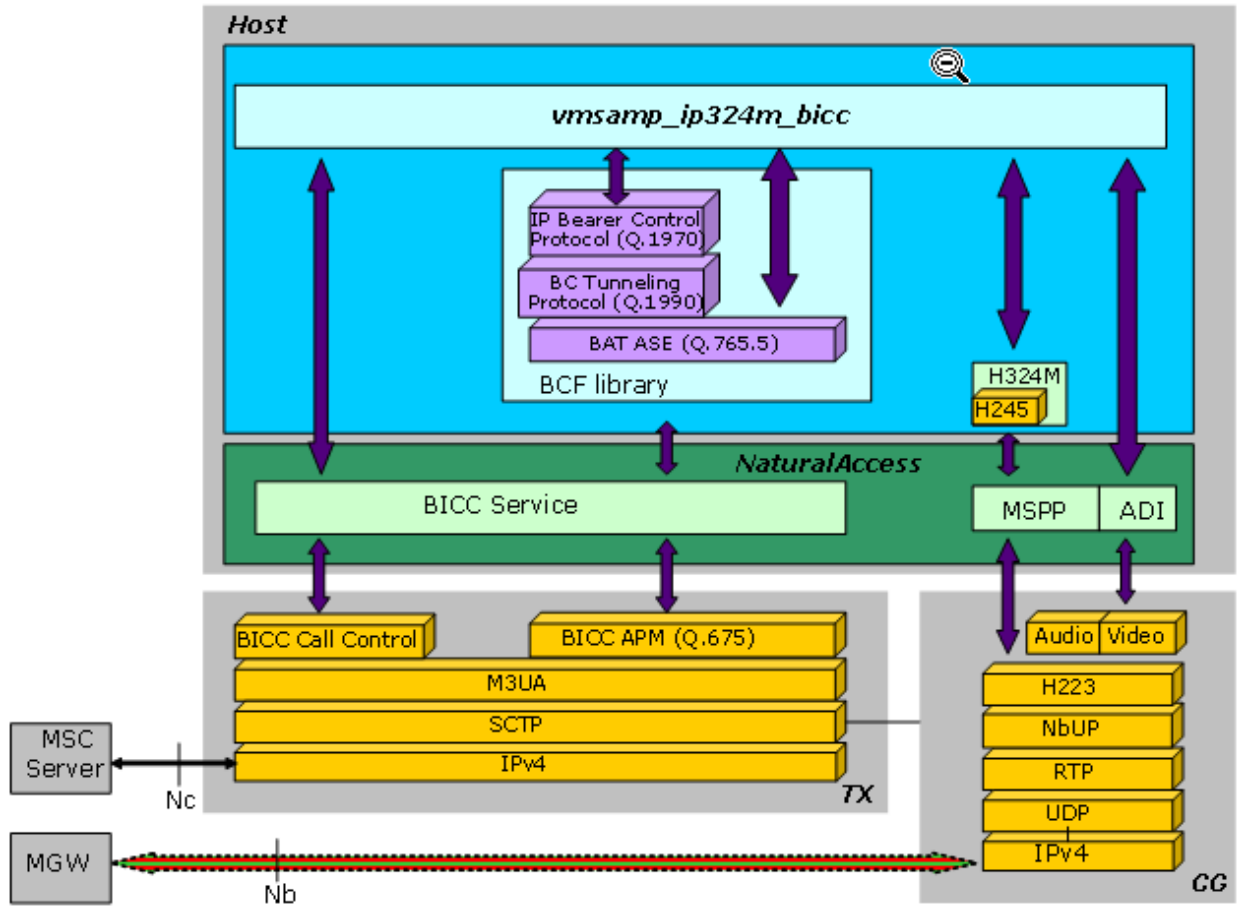
The following illustration shows forward bearer establishment with delayed forward tunneling Call Flow:



vmsamp_ip324m_bicc supports both inbound and outbound BICC call control, so the same (reverse) call flow is implemented when *vmsamp_ip324m_bicc* initiates the call.

3G-324M over IP Components

The following illustration shows the 3G-324M over IP components from a stack perspective and highlights the BCF library:



3. Bearer Control Function Library

Bearer Control Function Library Overview

The application uses the Bearer Control Functions (BCF) library to format and parse Bearer Control messages. In the context of the BCF library, a Bearer Control (BCF) Message consists of the following field sequence:

- APM'2000 Originating Address
- APM'2000 Destination Address
- Bearer Associated Transport (BAT) Application Service Element (ASE) Encapsulated Application Information, as specified in ITU-T Q765.5

The Bearer Control Message types that can be formatted and parsed using the BCF library and the main fields they encapsulate are:

Bearer Control Message Type	Main Fields Encapsulated
FAST_FORWARD_BEARER_SETUP_REQUEST	IPBCP Request Bearer Characteristics
FAST_FORWARD_BEARER_SETUP_ANSWER	IPBCP Answer
DELAYED_FORWARD_BEARER_INFO_REQUEST	Optional Codec List Bearer Characteristics
DELAYED_FORWARD_BEARER_INFO_ANSWER	Optional available Codec List Optional Selected Codec
DELAYED_FORWARD_BEARER_SETUP_REQUEST	IPBCP Request
DELAYED_FORWARD_BEARER_SETUP_ANSWER	IPBCP Answer

The BCF library complies with the following specifications:

- ITU-Q1902.4
- ITU-Q1970, IETF RFC 4566 (SDP), 3GPP TS 29.414
- BICC Bearer Control Tunneling Protocol (ITU-Q.1990)
- BAT ASE (Q.765.5)

The BCF can build and parse Bearer Control messages according to APM'2000 User Application. To send a Bearer Control message, the application needs to:

- Build the Bearer Control message using the BCF library
- Use the Dialogic® NaturalAccess™ BICC Layer to attach the Bearer Control message to a BICC Call Control (usually IAM) or APM message using the Application Transport Parameter IE

To receive a Bearer Control message, the application needs to:

- Use the Dialogic® NaturalAccess™ BICC Layer to get the Bearer Control message attached to a BICC Call Control or APM message
- Parse the Bearer Control message using the BCF library

For more information, refer to the *Dialogic® NaturalAccess™ ISUP Layer Developer's Reference Manual*.

Establishing a 3G-324M over IP Session

After configuring the boards, and creating and connecting all components shown in [Network Architecture](#), the host application must perform the following tasks to establish a 3G-324M session with an NbUP RTP Bearer:

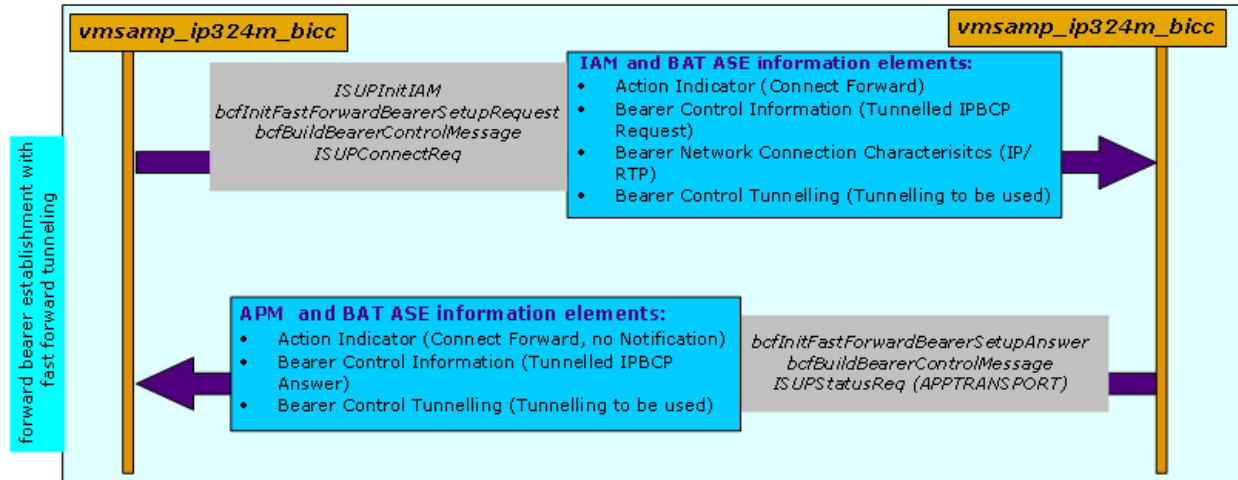
Task	Use	For more information, see...
1. Initiate BICC Call Control (either inbound or outbound).	SS7 ISUP	<i>Dialogic® NaturalAccess™ ISUP Layer Developer's Reference Manual</i>
2. Proceed with Bearer IP Control, and get the remote IPBCP parameters.	SS7 ISUP and BCF API	<i>Dialogic® NaturalAccess™ ISUP Layer Developer's Reference Manual</i> and Bearer Control Function Library Overview .
3. Configure the H223 GW RTP NbUP Endpoint and Channel according to remote IPBCP parameters. (IP address, UDP port, Payload ID, packetization time)	NaturalAccess MSPP	<i>Dialogic® NaturalAccess™ Media Stream Protocol Processing API Developer's Manual</i>
4. Establish the NbUP session.	NaturalAccess Fusion NbUP Endpoint	<i>Dialogic® NaturalAccess™ Fusion NbUP Endpoint and Clear Channel Reference Manual</i>
5. Set up a 3G-324M session.	3G-324M Interface	<i>Dialogic® NaturalAccess™ Video Access Toolkit 3G-324M Interface Developer's Reference Manual</i>

BCF Call Flows

The following illustrations show the function sequence for the two bearer control call flows implemented by *vmsamp_ip324m_bicc*. The ISUP and BCF sequence (in grey) is used to build and send each Bearer Control Message (in blue). A list of information elements that constitute each Bearer Control Message is also shown.

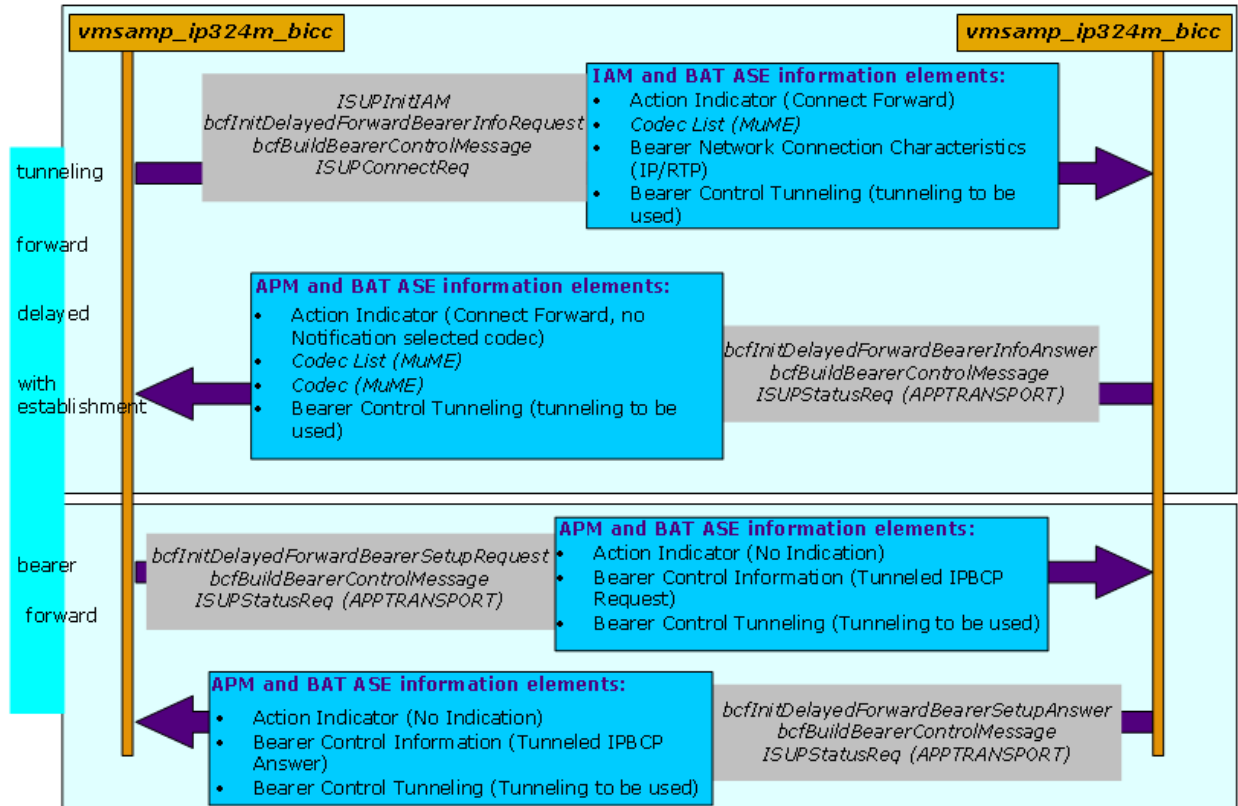
Forward Bearer Establishment with Fast Forward Tunneling

The following call flow shows Forward bearer establishment with fast forward tunneling Bearer Control Call Flow:



Forward Bearer Establishment with Delayed Forward Tunneling

The following call flow shows Forward bearer establishment with delayed forward tunneling Bearer Control Call Flow:



BCF Library Limitations

The following limitations apply to BCF Library:

Limitation	Description
IPBCP (Q1970)	<ul style="list-style-type: none"> • Messages: Request and Accepted Messages only (no Confused Message, no Rejected Message) • No IPBCP Timers • No IP bearer modification • No Alternative Network Address Type (no IPv6)
Q1990 (Tunneling Protocol)	No support for abnormal procedures.
Q765.5 (BAT ASE)	<p>Support for the following BAT ASE information elements:</p> <ul style="list-style-type: none"> • Action Indicator IE • Codec List IE • Single Codec IE • Bearer Network Connection Characteristics IE • Bearer Control Tunneling IE • Bearer Control Unit Identifier IE • Bearer Control Information IE (Tunneled IPBCP) • Backbone Network Connection Identifier • Inter-working Function Address <p>No support for the following BAT ASE information elements (the BCF library ignores these information elements if they are received in a BCF message):</p> <ul style="list-style-type: none"> • BAT Compatibility Report • Signal • Bearer Redirection Capability • Bearer Redirection Indicators • Signal Type • Duration
Q765.5	No segmentation/re-assembly.
APM User Application	APM'2000 (not APM'98).

Building a Bearer Control Message

To build a bearer control message:

1. Initialize the Bearer Control Event structure.
2. Build the Bearer Control message.

This topic provides instructions for completing these tasks.

Initialize the Bearer Control Event structure

The following structure defines the information elements that are required in the Bearer Control message to be formatted:

```
typedef struct _siBearerControlEvt
{
    SiActionIndicator          actionIndicator;
    SiCodecs                   codecsList;
    SiBearerNetworkCharacteristics networkCharacteristics;
    SiBearerControlInformation  bearerControlInfo;
    SiBearerControlTunnelling   bearerTunnelling;
    SiBearerControlUnitIdentifier bearerUnitId;
    SiCodecs                   singleCodec;
    SiBackboneNetworkIdentifier backboneNetworkId;
    SiInterworkingFunctionAddress iwfAddress;
} SiBearerControlEvt;
```

Information Elements

Name	Description
actionIndicator	Action indicator. Specifies the action associated with the message, and in particular, the bearer setup direction (forward, backward).
codecsList	Codecs list offered for codec negotiation.
networkCharacteristics	Bearer network connection characteristics. Supports IP/RTP only.
bearerControlInfo	Bearer control information. Contains the bearer control tunneling protocol (IPBCP) data unit.
bearerTunnelling	Bearer control tunneling indicator. Specifies that tunneling is to be used.
bearerUnitId	Bearer control unit identifier. Consists of the Network ID field (not currently supported) and the local BCU-ID field.
singleCodec	Single codec selected for codec negotiation.
backboneNetworkId	Backbone network connection identifier. Bearer specific and is specified in the relevant <i>Supplement 23 to Q-Series Recommendations</i> .
iwfAddress	Interworking function address. In NSAP format according to <i>Annex A/X.213 ISO/IEC 8348 and its Amendment 1</i> .

For more information, refer to [BCF Library Structures](#).

Init Functions

Several init functions are offered to initialize Bearer Control event structures with default values, specifically to support the Bearer Control reference call flows supported by *vmsamp_ip324m_bicc*:

Init functions for...	Prototypes
Fast Forward Bearer Setup Request and Answer	DWORD bcfInitFastForwardBearerSetupRequest (SiBearerControlEvt * <i>bearerControlEvt</i> , IPBCP_PARAM * <i>ipbciParam</i>); DWORD bcfInitFastForwardBearerSetupAnswer (SiBearerControlEvt * <i>bearerControlEvt</i> , IPBCP_PARAM * <i>ipbciParam</i>);
Delayed Forward Bearer Info Request and Answer	DWORD bcfInitDelayedForwardBearerInfoRequest (SiBearerControlEvt * <i>bearerControlEvt</i> , U8 codecList) DWORD bcfInitDelayedForwardBearerInfoAnswer (SiBearerControlEvt * <i>bearerControlEvt</i> , U8 codecList, U8 selectedCodec)
Delayed Forward Bearer Setup Request and Answer	DWORD bcfInitDelayedForwardBearerSetupRequest (SiBearerControlEvt * <i>bearerControlEvt</i> , IPBCP_PARAM * <i>ipbciParam</i>); DWORD bcfInitDelayedForwardBearerSetupAnswer (SiBearerControlEvt * <i>bearerControlEvt</i> , IPBCP_PARAM * <i>ipbciParam</i>);

After calling init functions, any *SiBearerControlEvt* structure field can be customized by the application implementing the bearer control logic.

Build the Bearer Control Message

After initializing the Bearer Control Event structure, use the following function to build the Bearer Control Message:

DWORD **bcfBuildBearerControlMessage** (U8 **outBuf*, U32 *outBufSize*,
SiBearerControlEvt **inBearerControl*, U32 **actualBcfSize*)

Parsing a Bearer Control Message

Use the **bcfParseBearerControlMessage** function to parse a Bearer Control Message into a Bearer Control Event structure:

```
DWORD bcfParseBearerControlMessage (
    U8 *inBuf,
    U32 inBufSize,
    tMimeType ipbcpMimeType,
    SiBearerControlEvt *outBearerControl)
```

This function fills the Bearer Control Event structure and allows the application to access the information element fields from the received Bearer Control message.

After parsing a Bearer Control Message, use the following macros to figure out the received message type:

- BCF_IS_FAST_FORWARD_BEARER_SETUP_REQUEST
- BCF_IS_FAST_FORWARD_BEARER_SETUP_ANSWER
- BCF_IS_DELAYED_FORWARD_BEARER_INFO_REQUEST
- BCF_IS_DELAYED_FORWARD_BEARER_INFO_ANSWER
- BCF_IS_DELAYED_FORWARD_BEARER_SETUP_REQUEST
- BCF_IS_DELAYED_FORWARD_BEARER_SETUP_ANSWER

If the received Bearer message is a Bearer Setup Request or a Bearer Setup Answer message, the application can use the **bcfInitIpbcParam** function:

```
DWORD bcfInitIpbcParam (
    IPBCP_PARAM *ipbcParam
    SiBearerControlEvt *bearerControlEvt)
```

To retrieve the following IPBCP/SDP parameters:

- payloadType of the RTP media stream
- Source and sink IP address of the RTP media stream
- Source and sink UDP Port of the RTP media stream
- Supported packetization time

4. BCF Library Functions

BCF Library Function Summary

BCF Library functions have similar coding conventions and usage to Dialogic® NaturalAccess™ Signaling Software.

For more information, refer to the data structure definitions in the *Dialogic® NaturalAccess™ ISUP Layer Developer's Reference Manual*.

bcfBuildBearerControlMessage

Builds a Bearer Control Message.

Prototype

DWORD **bcfBuildBearerControlMessage**(U8 **outBuf*, U32 *outBufSize*, SiBearerControlEvt **inBearerControl*, U32 **actualBcfSize*)

Argument	Description
<i>outBuf</i>	Output buffer to contain the BCF message.
<i>outBufSize</i>	Size (in bytes) of the output buffer used to contain the BCF message.
<i>inBearerControl</i>	Input Bearer Control Evt structure used to build the BCF message.
<i>actualBcfSize</i>	Actual size (in bytes) of BCF message.

Return Values

Argument	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.
BCF_ERROR	Error occurred while building the BCF message.
BCF_BUFFER_TOO_SMALL	The output buffer is too small.

Details

bcfBuildBearerControlMessage builds an outbound Bearer Control message based on SiBearerControlEvt structure.

APM'2000 Originating Address and Destination Address are set with Length = 0.

bcfInitFastForwardBearerSetupRequest

Initializes a Bearer Control Evnt Structure with the following information elements:

- actionIndicator
- networkCharacteristics
- bearerControlInfo
- bearerTunnelling

The resulting BCF Message is used to originate a forward bearer establishment with fast forward tunneling (IP core network).

It conveys in particular the IPBCP request and Network characteristics.

Prototype

DWORD **bcfInitFastForwardBearerSetupRequest** (SiBearerControlEvt **bearerControlEvt*, IPBCP_PARAM **ipbcuParam*)

Argument	Description
<i>bearerControlEvt</i>	Bearer Control event structure to initialize.
<i>ipbcuParam</i>	IPBCP Request input parameters used to initialize the Bearer Control event structure.

Return Values

Argument	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.

Details

Use the IPBCP Request to advertise the following RTP settings to the remote party:

- Source and sink IP address of the RTP media stream.
- Source and sink UDP port of the RTP media stream.
- Dynamic Payload Type of the RTP media stream.
- Supported packetization time (5 or 20 ms).

The information elements are initialized with the following default settings:

Information element	Default setting
actionIndicator	Connect forward
networkCharacteristics	IP
bearerControlInfo	Contains the tunneled bearer control protocol data (IPBCP Request)
bearerTunnelling	Tunneling to be used

Codec negotiation cannot be performed with a fast forward Bearer establishment procedure.

bcflnitFastForwardBearerSetupAnswer

Initializes a Bearer Control Evnt Structure with the following information elements:

- actionIndicator
- bearerControlInfo
- bearerTunnelling

The resulting BCF Message is used to terminate a forward bearer establishment with fast forward tunneling (IP core network). It conveys in particular the IPBCP answer.

Prototype

DWORD **bcflnitFastForwardBearerSetupAnswer** (SiBearerControlEvt
**bearerControlEvt*, IPBCP_PARAM **ipbcuParam*)

Argument	Description
<i>bearerControlEvt</i>	Bearer Control event structure to initialize.
<i>ipbcuParam</i>	IPBCP Answer input parameters used to initialize the Bearer Control event structure.

Return values

Return value	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.

Details

The IPBCP Answer is used to answer an IPBCP Request with the following RTP settings:

- Source and sink IP address of the RTP media stream
- Source and sink UDP port of the RTP media stream
- Dynamic Payload Type of the RTP media stream (shall be identical to the one proposed in the IPBCP Request)
- Supported packetization time

Note: Support for 5 ms packetization time is mandatory, and is the default packetization time. If the remote support 20 ms packetization time, and advertises it in the IPBCP request, the IPBCP answer may either acknowledge 20 ms packetization time, or force 5 ms.

The information elements are initialized with the following default settings:

Information element	Default setting
actionIndicator	Connect forward, no notification.
bearerControlInfo	Contains the tunneled bearer control protocol data (IPBCP Answer).
bearerTunnelling	Tunneling to be used.

Codec Negotiation cannot be performed with a fast forward Bearer establishment procedure.

bcfInitDelayedForwardBearerInfoRequest

Initialize a Bearer Control Evnt Structure with the following information elements:

- actionIndicator
- codecsList (optional)
- networkCharacteristics
- bearerTunnelling

The resulting BCF Message is used to originate a forward bearer establishment with delayed forward tunneling (IP core network).

It conveys in particular the optional supported codecs and Network characteristics.

Prototype

DWORD **bcfInitDelayedForwardBearerInfoRequest** (SiBearerControlEvt
**bearerControlEvt*, U8 **codecList*)

Argument	Description
<i>bearerControlEvt</i>	Bearer Control Evnt Structure to initialize.
<i>codecList</i>	Codec List. Bitmask of the following codecs: <ul style="list-style-type: none"> • AMR • AMR2 • G711 • MuME (H223) • MuME2 (H223)

Return values

Return value	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.

Details

Codec Negotiation is optional with this delayed forward Bearer establishment procedure.

The BCF library supports the following codecs:

- NO_CODEC (no codec negotiation being performed)
- AMR
- AMR2
- G711
- MuME (H223)
- MuME2 (H223)

As *vmsamp_ip324m_bicc* supports only H223, it is recommended to specify MuME codec only in the proposed codec list, if codec negotiation is supported and performed. Otherwise, specify NO_CODEC so that no codec negotiation takes place.

The information elements are initialized with the following default settings:

Information element	Default setting
actionIndicator	Connect forward.
codecsList	Optional, as passed in bcfInitDelayedForwardBearerInfoRequest .
networkCharacteristics	IP/RTP.
bearerTunnelling	Set to the value tunneling to be used.

bcfInitDelayedForwardBearerInfoAnswer

Initialize a Bearer Control event structure with the following information elements:

- actionIndicator
- codecsList (optional)
- singleCodec (optional)
- bearerTunnelling

The resulting BCF Message is used to terminate a forward bearer establishment with delayed forward tunneling (IP core network).

It conveys in particular the optional selected codec.

Prototype

DWORD **bcfInitDelayedForwardBearerInfoAnswer**(SiBearerControlEvt
**bearerControlEvt*, U8 **codecList*, U8 **selectedCodec*)

Argument	Description
<i>bearerControlEvt</i>	Bearer Control event structure to initialize.
<i>codecList</i>	Available codec list. Bitmask of the following codecs: <ul style="list-style-type: none"> • AMR • AMR2 • G711 • MuME (H223) • MuME2 (H223)
<i>selectedCodec</i>	Selected codec. Either: <ul style="list-style-type: none"> • NO_CODEC • AMR • AMR2 • G711 • MuME (H223) • MuME2 (H223)

Return Values

Return value	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.

Details

Codec Negotiation is optional with this delayed forward Bearer establishment procedure.

If codec Negotiation is performed, the application shall pass MuME codec to **bcflnitDelayedForwardBearerInfoAnswer** in order to force H223 as selected codec for the call.

The party answering codec negotiation shall also pass the available codec list.

If no codec negotiation is performed, then the application shall pass NO_CODEC.

The information elements are initialized with the following default settings:

Information element	Default setting
actionIndicator	If Codec Negotiation is performed: connect forward, no notification + selected codec. Otherwise: 'connect forward, no notification'
codecsList	Optional –available codec list, as passed in bcflnitDelayedForwardBearerInfoAnswer .
singleCodec	Optional – Selected codec, as passed in bcflnitDelayedForwardBearerInfoAnswer .
networkCharacteristics	IP/RTP.
bearerTunnelling	Set to the value tunneling to be used.

bcfInitDelayedForwardBearerSetupRequest

Initializes a Bearer Control event structure with the following information elements:

- actionIndicator
- bearerControlInfo
- bearerTunnelling

The resulting BCF Message is used to complete origination of a forward bearer establishment with delayed forward tunneling (IP core network). It conveys in particular the IPBCP request.

Prototype

DWORD **bcfInitDelayedForwardBearerSetupRequest** (SiBearerControlEvt
**bearerControlEvt*, IPBCP_PARAM **ipbcuParam*)

Argument	Description
<i>bearerControlEvt</i>	Bearer Control event structure to initialize.
<i>ipbcuParam</i>	IPBCP Request input parameters used to initialize the Bearer Control Evnt Structure.

Return Values

Return value	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.

Details

The IPBCP Request advertises the following RTP settings to the remote party:

- Source and sink IP address of the RTP media stream
- Source and sink UDP port of the RTP media stream
- Dynamic Payload Type of the RTP media stream
- Supported packetization time (5 or 20 ms)

The information elements are initialized with the following default settings:

Information element	Default setting
actionIndicator	No indication.
bearerControlInfo	Contains the tunneled bearer control protocol data (IPBCP Request).
bearerTunnelling	Set to the value tunneling to be used.

bcfInitDelayedForwardBearerSetupAnswer

Initializes a Bearer Control event structure with the following information elements:

- actionIndicator
- bearerControlInfo
- bearerTunnelling

The resulting BCF Message is used to complete termination of a forward bearer establishment with delayed forward tunneling (IP core network). It conveys in particular the IPBCP answer.

Prototype

DWORD **bcfInitDelayedForwardBearerSetupAnswer**(SiBearerControlEvt
**bearerControlEvt*, IPBCP_PARAM **ipbcuParam*)

Argument	Description
<i>bearerControlEvt</i>	Bearer Control event structure to initialize.
<i>ipbcuParam</i>	IPBCP Answer input parameters used to initialize the Bearer Control Evnt Structure.

Return Values

Return value	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.

Details

The IPBCP Answer is used to answer an IPBCP Request with the following RTP settings:

- Source and sink IP address of the RTP media stream
- Source and sink UDP port of the RTP media stream
- Dynamic Payload Type of the RTP media stream (shall be identical to the one proposed in the IPBCP Request)
- Supported packetization time

Note: Support for 5 ms packetization time is mandatory and the default. If the remote support 20 ms packetization time, and advertises it in the IPBCP request, the IPBCP answer acknowledges 20 ms packetization time or forces 5 ms.

The information elements are initialized with the following default settings:

Information element	Default setting
actionIndicator	No indication.
bearerControlInfo	Contains the tunneled bearer control protocol data (IPBCP Answer).
bearerTunnelling	Set to the value tunneling to be used.

bcfParseBearerControlMessage

Parses a Bearer Control Message.

Prototype

DWORD **bcfParseBearerControlMessage** (U8 **inBuf*, U32 *inBufSize*, tMimeType *ipbcpMimeType*, SiBearerControlEvt **outBearerControl*)

Argument	Description
<i>inBuf</i>	Input buffer containing the BCF message to be parsed.
<i>inBufSize</i>	Size of the input buffer containing the BCF message.
<i>ipbcpMimeType</i>	Expected Mime Type ID: <ul style="list-style-type: none"> • DEFAULT_MIME_TYPE=0 • NBUP_MIME_TYPE =1 • PCMU_MIME_TYPE =2 • PCMA_MIME_TYPE =3 • CLEARMODE_MIME_TYPE =4
<i>outBearerControl</i>	Bearer Control output parameters extracted from the BCF message.

Return Values

Return value	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.
BCF_UNKNOW_IE	Unknown information element.
BCF_BUFFER_TOO_SMALL	The input buffer is too small.

Details

bcfParseBearerControlMessage parses an inbound Bearer Control message and populates SiBearerControlEvt structure in order to easily access the relevant Information Elements fields.

Note: The default expected Mime Type ID is NBUP_MIME_TYPE (MimeType "VND.3GPP.IUFP"), which is the only payload specified by 3GPP for BICC RTP Bearer. NBUP_MIME_TYPE is the only MIME type supported in the context of Video Access Toolkit 3G-324M over IP feature, but other values are offered, for demo or testing purpose. See section: Disabling NbUP.

bcfInitIpbcpParam

Extracts the IPBCP/SDP parameters from a Bearer Control event structure.

Prototype

DWORD **bcfInitIpbcpParam** (IPBCP_PARM **ipbcparam* , SiBearerControlEvt **bearerControlEvt*)

Argument	Description
<i>ipbcparam</i>	IPBCP parameter structure to initialize.
<i>bearerControlEvt</i>	Bearer Control event structure to extract IPBCP parameters from.

Return values

Return value	Description
BCF_SUCCESS	
BCF_PARAM_ERROR	Parameter error.

5. Bearer Control Event Structures

BCF Library Structures

The BCF Library uses the following structures to define the Bearer Control event structure.

SiBearerControlEvt Structure

```
typedef struct _siBearerControlEvt /* Bearer Control event */
{
    SiActionIndicator actionIndicator; /* Action Indicator */
    SiCodecs codecsList; /* Codecs List */
    SiBearerNetworkCharacteristics networkCharacteristics; /*Bearer Network Connection
Characteristics*/
    SiBearerControlInformation bearerControlInfo; /* Bearer Control Information */
    SiBearerControlTunnelling bearerTunnelling; /* Bearer Control Tunnelling */
    SiBearerControlUnitIdentifier bearerUnitId; /* Bearer Control Unit Identifier */
    SiCodecs singleCodec; /* Single Codec */
    SiBackboneNetworkIdentifier backboneNetworkId; /* Backbone Network Connection
Identifier */
    SiInterworkingFunctionAddress iwfAddress; /* Interworking Function Address */
} SiBearerControlEvt;
```

SiBearerControlInformation Structure

```
typedef struct _bearerControlInformation /* Bearer Control Information*/
{
    ElmtHdr eh; /* element header */
    TknU8 bctpBvei; /* BCTP Version Error Indicator */
    TknU8 bctpTpei; /* BCTP Tunnelled Protocol Error Indicator */
    TknU8 ipbcpMessageType; /* IPBCP Message Type*/
    TknU8 srcPayloadType; /* PayloadType of the RTP media stream range = 96 to 127 */
    TknU8 ipbcpMimeType; /* Mime Type Identifier used to build an IPBCP message
*/
    TknU8 spare; /* spare */
    TknU16 srcRtpPort; /* source and sink UDP port of the RTP media stream */
    TknU16 srcPacketizationTime; /* supported packetization time */
    TknStr srcIpAddress; /* source and sink IP address of the RTP media stream */
} SiBearerControlInformation;
```

SiActionIndicator Structure

```
typedef struct _actionIndicator /* Action Indicator */
{
    ElmtHdr eh; /* element header */
    TknU8 indicator; /* action indicator */
    TknU8 spare; /* spare */
} SiActionIndicator;
```

SiBearerNetworkCharacteristics Structure

```
typedef struct _bearerNetworkCharacteristics /* Bearer Network Connection Characteristics */
{
    ElmtHdr eh; /* element header */
    TknU8 characteristics; /* Bearer Network Connection Characteristics */
    TknU8 spare; /* spare */
} SiBearerNetworkCharacteristics;
```

SiBearerControlTunnelling Structure

```
typedef struct _bearerControlTunnelling /* Bearer Control Tunnelling */
{
    ElmtHdr eh; /* element header */
    TknU8 tunnellingIndicator; /* Bearer Control Tunnelling Indicator */
    TknU8 spare; /* spare */
} SiBearerControlTunnelling;
```

SiBearerControlUnitIdentifier Structure

```
typedef struct _bearerControlUnitIdentifier /* Bearer Control Unit Identifier */
{
    ElmtHdr eh; /* element header */
    TknU32 networkId; /* Network ID - Not supported in the current version */
    TknU32 localBcuId; /* local BCU-ID */
} SiBearerControlUnitIdentifier;
```

SiAmrConfig Structure

```
typedef struct _amrConfig /* AMR Codec Config */
{
    TknU8 amrCodecConfigACS; /* Active Codec Set */
    TknU8 amrCodecConfigSCS; /* Supported Codec Set */
    TknU8 amrCodecConfigMACS; /* Maximal number of codec modes in the ACS */
    TknU8 amrCodecConfigOM; /* Optimisation Mode */
} SiAmrConfig;
```

SiAmrCodec Structure

```
typedef struct _amrCodec /* AMR Codec */
{
    ElmtHdr eh; /* element header */
    SiAmrConfig amrConfig; /* AMR Codec Config */
} SiAmrCodec;
```

SiG711Codec Structure

```
typedef struct _g711Codec /* G711 Codec */
{
    ElmtHdr eh; /* element header */
} SiG711Codec;
```

SiMuMeCodec Structure

```
typedef struct _mumeCodec /* MuMe Codec */
{
    ElmtHdr eh; /* element header */
    TknU8 bwm; /* B/W Multiplier */
} SiMuMeCodec;
```

SiCodecs Structure

```
typedef struct _codecs /* Supported Codecs */
{
    ElmtHdr eh; /* element header */
    SiAmrCodec amrCodec; /* Amr Codec */
    SiAmrCodec amr2Codec; /* Amr2 Codec */
    SiG711Codec g711Codec; /* G711 Codec */
    SiMuMeCodec mumeCodec; /* MuMe Codec */
    SiMuMeCodec mume2Codec; /* MuMe2 Codec */
} SiCodecs;
```

SiBackboneNetworkIdentifier Structure

```
typedef struct _backboneNetworkIdentifier /* Backbone Network Connection Identifier */
{
    ElmtHdr eh; /* element header */
    TknU8 spare; /* spare */
    TknU8 length; /* length of Connection ID field (max is 4 octets) */
    TknU32 connectionId; /* Connection ID */
} SiBackboneNetworkIdentifier;
```

SiInterworkingFunctionAddress Structure

```
typedef struct _interworkingFunctionAddress /* Interworking Function Address */
{
    ElmtHdr eh; /* element header */
    TknStr x213Address; /* Interworking Function Address formatted according to */
    /* Annex A/X.213 | ISO/IEC 8348 */
} SiInterworkingFunctionAddress;
```

IPBCP_PARAM Structure

Use the IPBCP_PARAM structure to describe the IPBCP parameters:

```
typedef struct
{
    int srcPayloadType; /* RTP PayloadType of media stream */
    char srcIpAddress[20]; /* source and sink IP address of the RTP media
stream */
    int srcRtpPort; /* source and sink UDP Port of the RTP media
stream */
    int srcPacketizationTime; /* supported packetization time */
    int ipbcpMessageType; /* IPBCP Messages Type */
    tMimeType ipbcpMimeType; /* Mime Type used to build an IPBCP message
*/
} IPBCP_PARAM ;
```

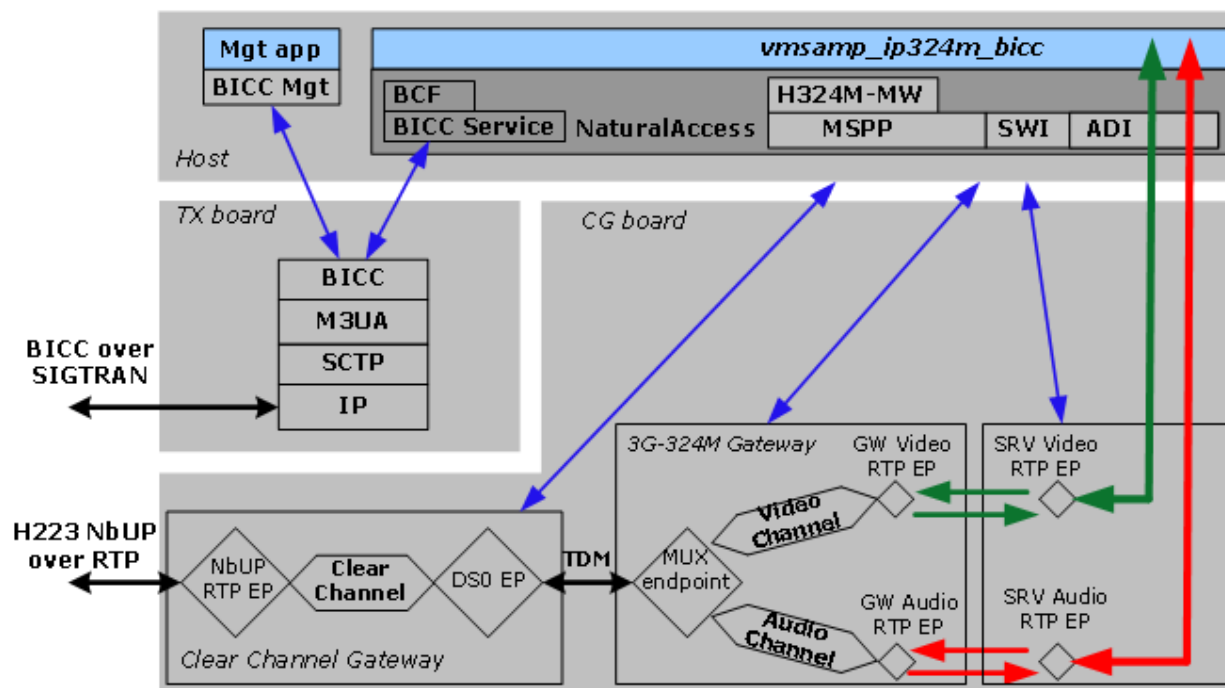
6. Demonstration Program

3G-324M Over IP Architecture

The following hardware is required to run the *vmsamp_ip324m_bicc* application:

- CG 6565 board for bearer processing: RTP packetization, NbUP framing protocol, H223 Multiplexer. The CG Ethernet interface is used to send and receive User Plane traffic.
- TX 4000 board for BICC over SIGTRAN signaling. The TX Ethernet interface is used to send and receive control plane traffic.
- Host for management and application control logic.

The *vmsamp_ip324m_bicc* application runs on the host, and accesses the board functionality through NaturalAccess and Video Access services:



From a software perspective, the 3G-324M over IP feature uses a Clear Channel gateway in front of Video Access 3G-324M gateway interface to tunnel H.223 data streams between TDM and RTP.

In the context of 3G-324M over IP, the clear channel gateway is also referred as H223 GW.

ITU BICC over SIGTRAN is used for call control and for establishing the RTP bearer channel settings and characteristics.

The *vmsamp_ip324m_bicc* application implements two configurable Bearer Control variants:

- Forward bearer establishment with fast forward tunneling.
- Forward bearer establishment with delayed forward tunneling.

vmsamp_ip324m_bicc Limitations

vmsamp_ip324m_bicc does not support IPv6 or RTCP. Although RTCP is supported by NaturalAccess endpoints, an application limitation prevents it from being supported.

3G-324M Parameters

The *vmsamp_ip324m_bicc* application provides new parameters (from the legacy *vmsamp* application) that control the settings for NbUP, Clear Channel Gateways, and BICC Call and Bearer Control. Original *vmsamp* parameters still apply.

This topic provides information about the new *vmsamp_ip324m_bicc* parameters you must configure in the gateway configuration files. For information about the original *vmsamp* parameters, refer to the *Video Mail Application Demonstration Program Manual*.

H.223 Gateway-Related Settings for vmsamp_ip324m_bicc

The following table describes the H.223 gateway related settings in the gateway configuration files (*g1.cfg* and *g2.cfg*) for *vmsamp_ip324m_bicc*:

Setting	Type	Default	Description
h223Gw_srcIPAddr	string	10.118.7.225	IP address of this board.
h223Gw_destIPAddr	string	10.118.7.226	IP address of the remote board to receive the H.223 over IP data. This address is re-negotiated during call setup using IPBCP.
h223Gw_srcPort	int	2000	UDP port to use on this board.
h223Gw_destPort	int	3000	UDP port of the remote User Plane Endpoint. This port is re-negotiated during call setup using IPBCP.
h223Gw_DspStream	int	64	DSP stream (64 for CG6X6X boards, 16 for CG 6000 boards).
h223Gw_ds0Timeslot	int	300	First timeslot that is configured for H.223 over IP.
h223Gw_JitterBufferDepth	int	4	H223 Gateway Jitter Buffer Depth
h223Gw_frameDuration	int	5	Frame Duration in ms (re-negotiated during call setup using IPBCP). Valid values: 5 ms or 20 ms
h223Gw_payloadType	int	99	Dynamic Payload Type (re-negotiated during call setup using IPBCP).
h223Gw_enable	int	0	Enables sending H.223 over IP. If this setting is not enabled, the other settings in this table do not take effect, and the bearer data is sent over TDM.

NbUP-Related Settings for vmsamp_ip324m_bicc

The following table describes the new NbUP related settings in the gateway configuration files (*g1.cfg* and *g2.cfg*) for *vmsamp_ip324m_bicc*:

Setting	Type	Default	Description
h223Gw_enableNbUP	int	1	NbUP is enabled by default.
h223Gw_modeNbUP	int	0	NbUP Slave Mode (0) or Master Mode (1).
h223Gw_pduTypeNbUP	int	0	NbUP PDU Type: 0 (with payload error detection) 1 (without payload error detection)
h223Gw_NbUPInitTimer	int	200	NbUP (Master or Slave) Timer (in ms)
h223Gw_NbUPInitRetry	int	20	Max number of Init Retransmissions for NbUP Master

ISUP-Related Settings for vmsamp_ip324m_bicc

The following table describes the ISUP related settings in the gateway configuration files (*g1.cfg* and *g2.cfg*) for *vmsamp_ip324m_bicc*:

Setting	Type	Default	Description
isupGw_board	int	1	TX board number.
isupGw_circuitId	int	1	First BICC circuit ID.
isupGw_switchType	int	12	ST_BICC
isupGw_logBcfMessages	int	0	0: log the received BCF messages 1: don't log the received BCF messages

BCF-Related Settings for vmsamp_ip324m_bicc

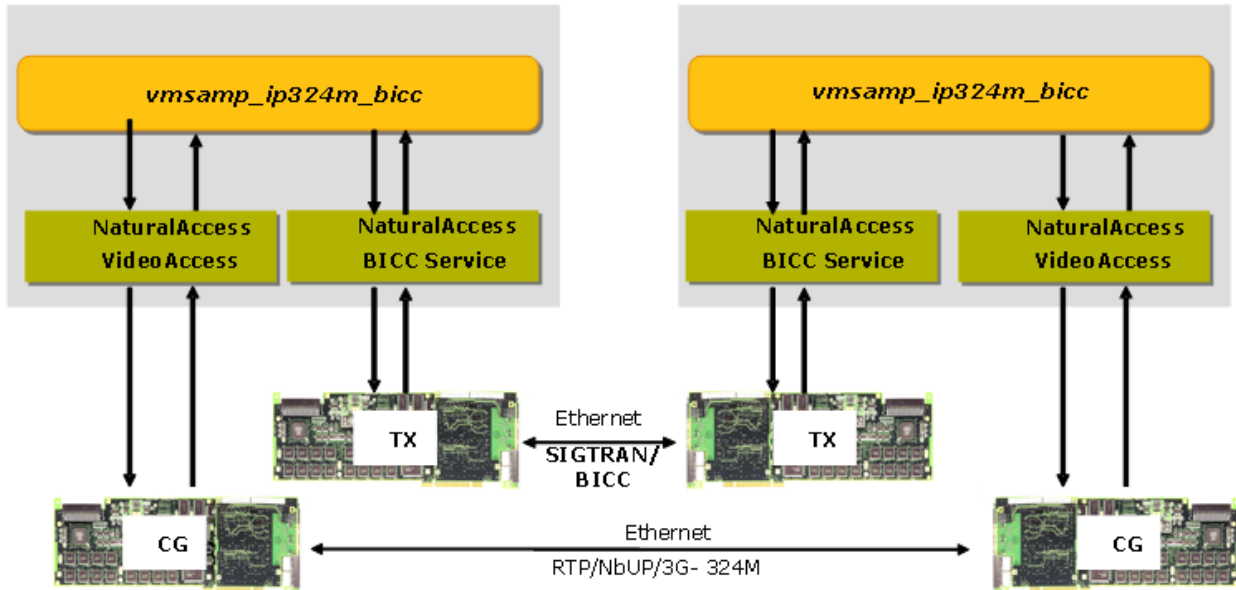
The following table describes the new BCF related settings in the gateway configuration files (*g1.cfg* and *g2.cfg*) for *vmsamp_ip324m_bicc*:

Setting	Type	Default	Description
bcfMask	int	340	BCF Library Trace Mask: 0 = TRACE_NONE 0x154 = TRACE_ALL_ERR 0xffff = TRACE_ALL
bcfCodecRequest	int	0	Optional Codec used for Codec Negotiation (only available for Delayed Bearer Setup Variant) 0 = BCF_NO_CODEC 1 = BCF_CODEEC_AMR 2 = BCF_CODEEC_AMR2 4 = BCF_CODEEC_G711 8 = BCF_CODEEC_MUME 16 = BCF_CODEEC_MUME2
bcfBearerSetupVariant	int	0	0 = Forward bearer establishment with fast forward tunneling 1 = Forward bearer establishment with delayed forward tunneling

Configuration Overview

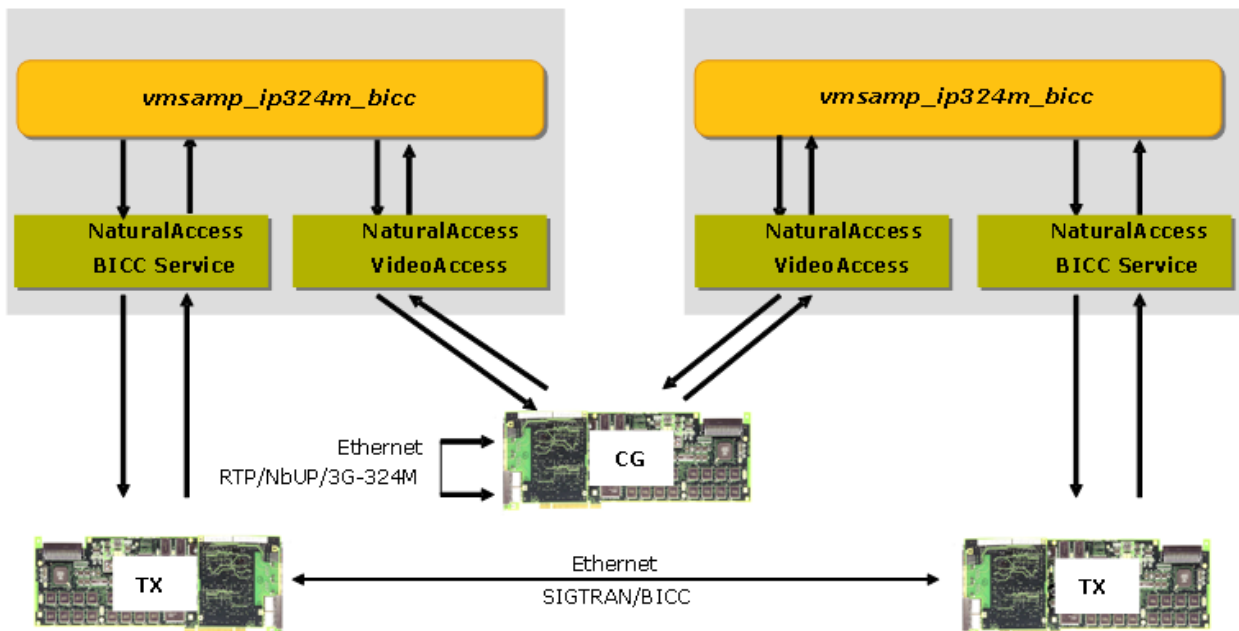
This topic describes how to configure Video Access Toolkit 3G-324M over IP to run in a back-to-back configuration.

A back-to-back configuration requires two TX 4000 boards and two CG 6565 boards, as shown:



Use either two Ethernet crossover cables to connect both pairs, or use a hub to connect each board. If you use a hub, you will be able to use a packet analyzer tool (for example, WireShark) to snoop the signaling and media traffic.

You can also use a two TX boards and one CG 6565 board with media looped back over both CG 6565 Ethernet interfaces:



Configuring the Software

The configuration files shipped with Video Access Toolkit 3G-324M over IP correspond to this configuration (two TX boards and one CG 6565 board).

SS7 configuration files and boot script files are provided with Video Access Toolkit 3G-324M over IP to help configure the TX boards. The files are located in the following directories:

Operating system	Directory	Description
Windows	<code>\nms\vaccess\demos\vmsamp\bicc_config\bicc</code>	TX board configuration files
	<code>\nms\vaccess\demos\vmsamp\bicc_config\boot</code>	Boot script files
Unix	<code>opt/nms/vaccess/demos/vmsamp/bicc_config/bicc</code>	TX board configuration files
	<code>opt/nms/vaccess/demos/vmsamp/bicc_config/boot</code>	Boot script files

To use these files, copy the bicc directory to:

- *Program Files\dialogic\tx\config\standalone* directory (Windows)
- */opt/dialogic/tx/etc/standalone* directory (Unix)

Copy the batch file located in the boot directory into:

- *Program Files\dialogic\tx\bin* directory (Windows)
- */opt/dialogic/tx/bin* directory (Unix)

File	Description
<i>biccsigtranload.bat</i>	Configures and loads the TX boards with SIGTRAN and BICC. This is the configuration that is supported in the context of 3G-324M over IP feature.

Configuration Procedure

Complete the following procedure after the boards are connected:

1. Change the IP address, subnet mask, and gateway IP address settings in the CG 6565 board configuration file (*vm_6565_ip324m.cfg*) to match the settings in your network. Note that two different IP addresses are used (one per Ethernet Interface).

The following example shows the IP address settings whose values you must change:

```
#####
#
# IP Address, subnet mask, and gateway IP address.
# Note: the IP configuration below is for a Ethernet Failover
# THIS CONFIGURATION FILE WILL FAIL UNLESS THE VARIABLE STRINGS
# BELOW ARE REPLACED WITH REAL IP ADDRESSES.

IPC.AddRoute[0].DestinationAddress = 10.51.8.14
IPC.AddRoute[0].Mask = 255.255.0.0
IPC.AddRoute[0].Interface = 1

IPC.AddRoute[1].DestinationAddress = 0.0.0.0
IPC.AddRoute[1].Mask = 0.0.0.0
IPC.AddRoute[1].GatewayAddress = 10.51.0.1

IPC.AddRoute[2].DestinationAddress = 10.51.8.15
IPC.AddRoute[2].Mask = 255.255.0.0
IPC.AddRoute[2].Interface = 2
```

Note: Do not change values in the RESOURCE MANAGEMENT section of the configuration file. Changing these values could cause the demo to fail.

2. Modify the *oamsys* configuration file to point to the proper Product Type, Logical board number (0), PCI bus and slot, and CG6565 board configuration file (*vm_6565_ip324m.cfg*).
3. Run *oamsys* to reboot the CG board.
4. Change the IP address, subnet mask, and gateway IP address settings for the TX boards to match the settings in your network by modifying the BICC configuration files:
 - *ipcfg1.txt* and *ipcfg2.txt*
 - *M3UAcp1.cfg* and *M3UAcp2.cfg*
5. Run *txalarm* in a console window, to monitor the TX boards status:

```
txalarm
```

6. Boot both of the TX boards with BICC and SIGTRAN configurations:

```
biccsigtranload 1
biccsigtranload 2
```

7. Check that both TX boards send an Association established successfully message to *txalarm* indicating that they are properly configured.

8. Modify the *vmsamp_ip324m_bicc* 3G-324M Interface configuration files (*g1.cfg* and *g2.cfg*, located in *nms/vaccess/demos/vmsamp*) by replacing the values in **bold** with values appropriate for your network.

Verify that *g1.cfg* is configured with Ethernet Interface 1 IP address, and *g2.cfg* with Ethernet Interface 2 IP address.

```

srcIPAddr_video = 10.51.8.14
destIPAddr_video = 10.51.8.14
srcIPAddr_audio = 10.51.8.14
destIPAddr_audio = 10.51.8.14

terminalType=254

nPorts=1
useVideoXc = 0
#video format: MPEG4(1), H263(2), H263+(3), H264(4)
Video_Format = 2

notifyIframe = 1

h223Gw_enable = 1

h223Gw_srcIPAddr = 10.51.8.14
h223Gw_destIPAddr = 10.51.8.14

h223Gw_srcPort = 7000
h223Gw_destPort = 8000

h223Gw_UseIPv6 = 0
h223Gw_DspStream = 64
h223Gw_ds0Timeslot = 300

h223Gw_JitterBufferDepth = 4
h223Gw_frameDuration = 5
h223Gw_payloadType = 123

isupGw_board = 1
isupGw_circuitId = 1
isupGw_switchType = 12

h223Gw_enableNbUP = 1
h223Gw_modeNbUP = 1

h223Gw_NbUPInitTimer = 200
h223Gw_NbUPInitRetry = 20

h223Gw_pduTypeNbUP = 0

bcfCodecRequest = 0

# Apply to outbound scenario
# BCF_FAST_FORWARD_BEARER_SETUP_VARIANT 0
# BCF_DELAYED_FORWARD_BEARER_SETUP_VARIANT 1
bcfBearerSetupVariant=0

```

For a back-to-back test configuration, note that one side must be NbUP Master (*h223Gw_modeNbUP = 1*) and the other side NbUP Slave (*h223Gw_modeNbUP = 0*). For more information, refer to the *Dialogic® NaturalAccess™ Video Access Toolkit 3G-324M Interface Developer's Reference Manual*.

9. Modify the *vmsamp_ip324m_bicc* Video Messaging Server Interface configuration files (*s1.cfg* and *s2.cfg*) by replacing the values in bold with values appropriate for your network.

Verify that *s1.cfg* is configured with Ethernet Interface 1 IP address and *s2.cfg* with Ethernet Interface 2 IP address:

```

srcIPAddr_video = 10.51.8.14
destIPAddr_video = 10.51.8.14
srcIPAddr_audio = 10.51.8.14
destIPAddr_audio = 10.51.8.14

nPorts=1
AutoPlayRec = 0
AutoPlay = 0
AutoRecord = 0

3gpFileName = test.3gp
    
```

For more information, refer to the *Dialogic® NaturalAccess™ Video Access Toolkit Video Messaging Server Interface Developer's Reference Manual*.

Running `vmsamp_ip324m_bicc`

Complete the following steps to run `vmsamp_ip324m_bicc` for a simple back-to-back play/record scenario:

1. Optionally set up WireShark to collect the H.223 RTP streams and BICC signaling messages in both directions (UDP and SCTP traffic).
2. Navigate to the `\nms\vaccess\demos\vmsamp` directory on the source side.
3. Enter the following commands at the command line:

```
vmsamp_ip324m_bicc -g g1.cfg -s s1.cfg
vmsamp_ip324m_bicc -g g2.cfg -s s2.cfg
```

4. Place a call by entering the following commands in the first `vmsamp_ip324m_bicc` instance:

```
x
c
123
```

5. Start audio and video recording on the first `vmsamp_ip324m_bicc` instance:
 - **z** = play and record commands.
 - **r** = start audio and video recording.
6. Navigate to the `\nms\vaccess\demos\vmsamp` directory on the destination side.
7. Play a 3GP file from the second `vmsamp_ip324m_bicc` instance:
 - **z** = play and record commands.
 - **s** = play a 3GP file (you must have an H263 3GP file named `test.3gp` in the `vmsamp` directory).
8. Stop audio and video recording on the first `vmsamp_ip324m_bicc` instance:
 - **z** = play and record commands.
 - **u** = stop audio recording. Video recording automatically stops at the same time.
9. Check that the recorded media on the first `vmsamp_ip324m_bicc` instance matches the original 3GP file played on the second `vmsamp_ip324m_bicc` instance.

NbUP RTP Endpoint statistics

After establishing the RTP Bearer and NbUP session, type **d** in the 1 port *vmsamp_ip324m_bicc* menu. *vmsamp_ip324m_bicc* retrieves NbUP RTP Endpoint configured information and statistics.

The following example shows the output:

```

RTP FDX Endpoint Data:
  Filter ID           =      412
  IP Filter State     =        3
  Local IP Addr       =    10.51.8.14
  Local Port          =     7000
  Remote IP Addr      =    10.51.8.15
  Remote Port         =     8000
  rtcp events         =        0
  link events         =        0

RTP Endpoint Stats:
  Rx Drops            =        0
  Rx Pkts             =    12388
  Rx Bytes            =   2031474
  Tx Drops            =     185
  Tx Fail             =        0
  Tx Pkts             =    12388
  Tx Bytes            =   2031479

RTP Assembler Stats:
  frame quota         =        1
  dtmf frames         =        0

RTP Disassembler Stats:
  last PID            =        0
  mismatches          =        0
  last map vocoder    =     123
  last map PID        =     123
  dtmf frames         =        0
  dtmf event control  =        3

NBUP Endpoint NBUP status:
  NbUP Mode configured =        2
  current NbUP state   =        2
  current NbUP Media   =        2
  Received NbUP PDU message count = 12387
  Received NbUP Control message count = 1
  Transmitted NbUP PDU message count = 12387
  Transmitted NbUP Control message count = 1
  Rate control failure count = 0
  Time Alignment Error count = 0
  Number of frames lost = 0
  Unexpected frame error count = 0
  Unexpected PDU error count = 0
  Unexpected Control Procedure count = 0
  Unknown PDU error count = 0
  Header CRC checksum error count = 0
  Payload CRC checksum error count = 0

```