=

TESTING

EXCENTIS

IPAT requirements for Euro-PacketCable certification

--- Project Reference ---

Document Reference : IPAT requirements for Euro-PacketCable certification
Revision : 7.0
Author(s) : testing@excentis.com
Date : November 15, 2006
Distribution : www.excentis.com

1 Introduction

This document specifies and clarifies the requirements for an IPAT (also called V5.2 gateway) that is submitted for Euro-PacketCable certification. This document is dynamic in nature and vendors should be checking for updates regularly. Specific document version numbers are however not mentioned. The version numbers applicable for a specific Euro-PacketCable Certification Wave are published in the guidelines for that wave.

2 Normative documents

2.1 PKT-SP-EC-MGCP

This document defines the call signaling protocol. Some parts of appendix A of this document are redefined in [Euro-PacketCable L-Package Clarification]. If something is mentioned in both documents, it is the latter document that has priority. If something is mentioned in both [PKT-SP-EC-MGCP] and in section 3, it is section 3 that has priority.

2.2 PKT-SP-DQOS

This document defines the dynamic quality of service protocol. Section 3 of this document specifies or alters requirements applicable for IPATs. If something is mentioned in both [PKT-SP-DQOS] and in section 3, it is section 3 that has priority.

2.3 PKT-SP-SEC

This document defines the security protocols. Specific requirements for Euro-PacketCable certificates are defined in [Euro-PacketCable Certificate Requirements].

Section 3 of this document specifies or alters requirements applicable for IPATs. If something is mentioned in both [PKT-SP-SEC] and in section 3, it is section 3 that has priority.

2.4 PKT-SP-CODEC

This document defines the requirements for CODEC handling and support. Section 3 of this document specifies or alters requirements applicable for IPATs. If something is mentioned in both [PKT-SP-CODEC] and in section 3, it is section 3 that has priority.

2.5 Euro-PacketCable Certificate Requirements

This document defines the requirements for Euro-PacketCable certificates.

2.6 Euro-PacketCable L-Package Clarification

This document defines the requirements for the L-package for Euro-PacketCable.

2.7 ETSI TS 101 909-23

This document defines the specifics for IPATs. Sections 6.6, 6.10 and Annex D of [ETSI TS 101 909-23] are not applicable. Annex C is replaced by section 4 of this document. Statements in [ETSI TS 101 909-23] concerning the use of Radius for Gate Coordination are not applicable, since all DQoS messaging is done as per section 2.2 of this document.

Section 3 of this document specifies or alters requirements applicable for IPATs. If something is mentioned in both [ETSI TS 101 909-23] and in section 3, it is section 3 that has priority.

3 Requirement clarification/alteration

3.1 Introduction

If a requirement is specified in this section (the whole section 3) it takes priority over a conflicting requirement in any other document. The goal of this section 3 is to make sure the IPAT-requirements are:

- clear and unambiguous
- require no modification to MTA, CMTS or any other component in the system
- complete

3.2 NCS related

3.2.1 Port numbers

The default UDP port for call signaling towards the IPAT is 2727.

3.2.2 ETSI "E" package

Support for the ETSI "E" package, as defined in ETSI TS 101 909, is NOT required.

3.2.3 Codec and packetization period selection

An IPAT must perform codec and packetization period selection before sending any of the following:

a) a CreateConnection command with a RemoteConnectionDescriptor parameter (i.e. the SDP session description of the IPAT)

b) a ModifyConnection command with a RemoteConnectionDescriptor parameter (i.e. the SDP session description of the IPAT)

An IPAT must also perform codec and packetization period selection when processing any of the following:

c) a response to a CreateConnection or ModifyConnection command containing a LocalConnectionDescriptor (i.e. the SDP session description of the MTA)

An IPAT must not perform codec and packetization period selection in any other case.

Furthermore, this codec and packetization period selection process must only use the information present in the current response (if it was a response that triggered the selection process) and not retain any of the values that may have been received in a previous response. As well, if the LocalConnectionDescriptor is omitted in a response to a MDCX command the existing negotiated codecs and packetization periods will remain intact.

In determining which codec(s) and packetization period(s) to provide in the RemoteConnectionDescriptor, there are three lists of codecs and packetization periods that an IPAT needs to consider:

- A list of codecs and packetization periods allowed by operator provisioning. This list can be thought of as the "virtual" LocalConnectionOptions that apply to the IPAT. A codec is allowed by operator provisioning if it satisfies the constraints specified by the encoding method, packetization period and multiple packetization periods. If one or more of these properties are not specified, then they do not impose any constraints on the allowed codecs. This list can be provisioned for all endpoints or on a per-endpoint basis.
- A list of codecs and packetization periods in the LocalConnectionDescriptor.
- An internal list of codecs and packetization periods that the IPAT can support for the connection. An IPAT may support one or more codecs and packetization periods for a given connection.

Codec selection (including all relevant media parameters) involves the following:

1. An approved list of codecs/packetization periods is formed by taking the intersection of the internal list of codecs/packetization periods and codecs/packetization periods allowed by operator provisioning. If there is no operator provisioned list of allowed codecs/packetization periods, the approved list of codecs/packetization periods thus contains the internal list. If there is an operator provisioned list of allowed media parameters, and the codecs are not part of it, then all codecs in the internal list are allowed, provided they are not incompatible with any packetization period(s) specified. Similarly, if the provisioned list only contains codecs, but no packetization period(s), the provisioned list implicitly contains the set of packetization periods supported by the internal list.
2. If the approved list of codecs/packetization periods is empty, an error condition exists that must be brought to operator attention using the IPAT's error reporting mechanism.
3. Otherwise, a negotiated list of codecs/packetization periods is formed by taking the intersection of the approved list of codecs/packetization periods and codecs/packetization periods allowed by the LocalConnectionDescriptor parameter (if present; this is the SDP session description of the MTA). If a LocalConnectionDescriptor was not provided or has not yet been received, the negotiated list of codecs/packetization periods thus contains the approved list of codecs/packetization periods. If the LocalConnectionDescriptor is present and does not contain any media stream lines, a codec negotiation failure has occurred, and the IPAT MUST send a DeleteConnection command to the MTA. If the LocalConnectionDescriptor contains multiple media streams, the IPAT SHOULD only accept one of these and reject the others by setting their port to zero in its RemoteConnectionDescriptor. If the LocalConnectionDescriptor was provided but the packetization period(s) was omitted, the negotiated list of packetization periods contains the set of packetization periods from the approved list. The IPAT MUST choose reasonable defaults [6] if the packetization period is explicitly omitted from both the operator provisioned list of allowed media attributes and the LocalConnectionDescriptor.

4. If the negotiated list of codecs/packetization periods is empty, a codec negotiation failure has occurred and the IPAT MUST send a DeleteConnection command to the MTA, except when no other commands regarding the current connection or the connection currently being established were sent to the MTA before.
5. Otherwise, codec negotiation has succeeded. If the codec negotiation took place because the IPAT was about to send a CreateConnection or ModifyConnection command, then the negotiated list of codecs/packetization periods is sent in the RemoteConnectionDescriptor of this command.

Note that both the operator provisioned list of allowed codecs and the LocalConnectionDescriptor can contain a list of codecs ordered by preference. When both are supplied, the IPAT should adhere to the preferences provided by the operator.

In the case that an IPAT supports more than one codec per connection, there are two options the IPAT can use in deciding how many codecs it wants to support for that connection:

1. IPAT supports multiple codecs and can switch between different codecs in real-time. The IPAT returns all negotiated codecs in the SDP media stream line and authorizes the Least-Upper-Bound (LUB) as per the DQoS specification. The LUB is authorized to guarantee that a switch to any of these codecs will succeed. Multiple codecs in the m= line means the device must be ready to receive media packets from any of the negotiated codecs. As well, the IPAT may send media packets from any of the negotiated codecs and switch between them as required.
2. IPAT supports one or more codecs but cannot switch between different codecs in real-time. The IPAT therefore negotiates and returns only one codec in the SDP media stream line, (optionally, IPAT also puts additional supported codecs in the SDP 'X-pc-codecs' attribute) and authorizes the bandwidth for the single negotiated codec in the media stream line as per the DQoS specification. With this method, a codec change must be initiated by the IPAT in order to change codecs at which time the resulting change in bandwidth is re-established as per the DQoS specification.

3.2.4 Security associations and signaling

The IPAT MUST accept signaling and bearer channel requests from a MTA that has an active security association.

The IPAT MUST NOT accept signaling and bearer channel requests from a MTA that does not have an active security association unless provisioned to do so with information corresponding to the "pktcMtaDevCmsIpsecCtrl" MIB Object.

3.2.5 Interaction with V5.2 BCC protocol

3.2.5.1 Allocation of a bearer channel

In response to a BCC ALLOCATION message from the Local Exchange (LE), the IPAT will try to create a connection to an MTA. If this connection can not be made, then the IPAT MUST make sure that the LE receives a notification of this, so that the LE can decide to release the allocated resources.

3.2.5.2 Deallocation of a bearer channel

Upon reception of a BCC DEALLOCATION message from the Local Exchange (LE), the IPAT MUST make sure that the call leg (for which the bearer channel is to be deallocated) and all associated resources are deleted. This means that the IPAT will have to send a DeleteConnection message to the MTA and that the IPAT will have to be conformant to all related DQoS requirements. Timer T5 applies as per clause 6.6 of [PKT-SP-DQOS].

3.3 DQoS related

The goal of the DQoS-specification is to prevent theft-of-service by the client. The IPAT has always to be aware of the current state of the endpoint of device. The IPAT is responsible for assuring that a gate has been closed after the call has finished. If the gate was not closed, the IPAT must force the CMTS to close the gate by sending a GATE-DELETE command for the gate that was left open. The IPAT must use the information provided by the CMTS in the GATE-CLOSE message to determine that a gate has been closed, if the gate was not closed within the required time, the GATE-DELETE must be sent.

3.4 Security related

3.4.1 Principal identifier

As the MTA is not aware if an IPAT or CMS is used, the principal identifier (needed for requesting a ticket from the KDC) has to be constructed in the same way as for a CMS. This means the principal identifier for an IPAT MUST be:

cms/<FQDN>@<realm>

3.4.2 Euro-PacketCable Server Certificates

The value of <Sub-System Name> for IP Access Terminals is “ipat”.

3.4.3 Null ciphersuite combinations and ordering

If an IPAT receives a LocalConnectionDescriptor from an MTA with a list of ciphersuites that meets the prohibited combinations listed in clause 7.6.2.3.1.1 of [PKT-SP-SEC], or that includes the NULL authentication and NULL encryption combination (60/50 for RTP, and 80/70 for RTCP), but this combination is not last, the IPAT MUST send a DeleteConnection command to the MTA for the affected connection.

3.4.4 Ciphersuite negotiation for RTP and RTCP streams

An IPAT MUST perform RTP and RTCP ciphersuite negotiation before sending any of the following:

- a CreateConnection command with a RemoteConnectionDescriptor parameter (i.e. the SDP session description of the IPAT)
- a ModifyConnection command with a RemoteConnectionDescriptor parameter (i.e. the SDP session description of the IPAT)

An IPAT MUST also perform RTP and RTCP ciphersuite negotiation when processing any of the following:

- a response to a CreateConnection or ModifyConnection command containing a LocalConnectionDescriptor (i.e. the SDP session description of the MTA)

An IPAT MUST NOT perform ciphersuite negotiation in any other case. The steps involved in ciphersuite negotiation are the following:

1. An approved list of ciphersuites is formed by taking the intersection of the internal list of ciphersuites and ciphersuites allowed by operator provisioning. The internal list of ciphersuites contains the ciphersuites that the IPAT supports and which this specification requires. If there is no operator provisioned list of allowed ciphersuites, the approved list of ciphersuites contains the previously agreed upon approved list, or if no such list exists, the internal list of ciphersuites.
2. If the approved list of ciphersuites is empty, an error condition exists that must be brought to operator attention using the IPAT's error reporting mechanism.
3. Otherwise, a negotiated list of ciphersuites is formed by taking the intersection of the approved list of ciphersuites and ciphersuites allowed by the LocalConnectionDescriptor parameter (if present; this is the SDP session description of the MTA), subject to the constraints specified in section 7.6.2.3.1.1. If a LocalConnectionDescriptor was not provided, the negotiated list of ciphersuites thus contains the approved list of ciphersuites. If a LocalConnectionDescriptor parameter is provided without fields containing the RTP and RTCP ciphersuite lists, then the RTP_AUTH_NULL/RTP_ENCR_NULL and RTCP_AUTH_NULL/RTCP_ENCR_NULL ciphersuites are assumed for the remote endpoints, and the regular ciphersuite negotiation process continues (i.e., the negotiated list of ciphersuites is formed by taking the intersection of the approved list of ciphersuites and the RTP_AUTH_NULL/RTP_ENCR_NULL and RTCP_AUTH_NULL/RTCP_ENCR_NULL ciphersuites).
4. If the negotiated list of ciphersuites is empty, a ciphersuite negotiation failure has occurred, and the IPAT MUST send a DeleteConnection command to the MTA, except when no other commands regarding the current connection or the connection currently being established were sent to the MTA before.
5. Otherwise, ciphersuite negotiation has succeeded. If the ciphersuite negotiation took place because the IPAT was about to send a CreateConnection or ModifyConnection command, then the negotiated list of ciphersuites is sent in the RemoteConnectionDescriptor parameter of this command.

The following requirements apply during ciphersuite negotiation:

- If the IPAT receives a LocalConnectionDescriptor parameter (i.e. the SDP session description of an MTA) with RTP_AUTH_NULL/RTP_ENCR_NULL for RTP or RTCP_AUTH_NULL/RTCP_ENCR_NULL for RTCP that is not last in the list, it MUST send a DeleteConnection command to the MTA.
- An IPAT MUST be capable of sending the allowable lists of ciphersuites for RTP and/or RTCP in the LocalConnectionOptions parameter of a CreateConnection command (CRCX) or a ModifyConnection command (MDCX) in the order of preference specified by the operator subject to the constraints specified in section 7.6.2.3.1.1.

- Whenever possible, an IPAT SHOULD select the first supported ciphersuite for RTP and the first supported ciphersuite for RTCP in the LocalConnectionDescriptor parameter. This allows the IPAT to immediately start sending RTP and RTCP packets to the MTA. An IPAT MAY instead select alternate ciphersuites specified by the MTA.
- When sending a RemoteConnectionDescriptor and the negotiated list of RTP and RTCP ciphersuites is NULL, an IPAT MUST NOT include an End-End Secret or Pad.
- When sending a RemoteConnectionDescriptor and the negotiated list of RTP and RTCP ciphersuites contains at least one non-NULL selection each, an IPAT MUST include an End-End Secret (for incoming RTP and RTCP packets) and MAY include a Pad value (for outgoing RTP and RTCP packets). The following rules apply:
    1. The IPAT MUST generate a new End-End Secret when first sending a CreateConnection command with a RemoteConnectionDescriptor.
    2. The IPAT MUST generate a new End-End Secret when sending a ModifyConnection command where the connection address on the IPAT side (e.g., IP Address) or the transport address on the IPAT side (e.g., port) are not identical to what was previously sent.
    3. The IPAT MUST generate a new Pad when first sending a CreateConnection command with a RemoteConnectionDescriptor.
    4. If not otherwise required, the IPAT MAY generate a new Pad when generating a new End-End Secret.
    5. The IPAT MUST NOT generate a new Pad when not generating a new End-End Secret.
- If, when sending a CreateConnection command with a RemoteConnectionDescriptor, the list of ciphersuites selected for RTP contains at least one non-NULL encryption or authentication algorithm, before sending the command, an IPAT MUST:
    1. Establish inbound RTP security based on the preferred (first) RTP ciphersuite and its End-End Secret (which it generated), as described in section 7.6.2.3.3.1 of this specification.
    2. If applicable for the connection mode specified in the command, be ready to receive RTP packets, which may arrive any time after the command is sent.
- If, when sending a CreateConnection command with a RemoteConnectionDescriptor, the list of ciphersuites selected for RTCP contains at least one non-NULL encryption or authentication algorithm, before sending the command, an IPAT MUST:
    1. Establish inbound RTCP security based on the preferred (first) RTCP ciphersuite and its End-End Secret (which it generated), as described in section 7.6.2.3.3.1 of this specification.
    2. If applicable for the connection mode specified in the command, be ready to receive RTCP packets, which may arrive any time after the command is sent.
- If, when sending a ModifyConnection command with a RemoteConnectionDescriptor, the list of ciphersuites selected for RTP contains at least one non-NULL encryption or authentication algorithm, before sending the command, an IPAT MUST:
    1. If a Pad is included in the RemoteConnectionDescriptor and it is different than a Pad that may have previously been sent, remove any existing outbound RTP keys and

generate new ones, based on the keys that are generated from both the End-End Secret (generated by the MTA) and the Pad (generated locally). The IPAT MUST re-initialize the RTP timestamp if new keys are generated. The ciphersuites used for these outbound keys are those from the RemoteConnectionDescriptor parameter that is to be sent to the MTA.

2. If a Pad is included in the RemoteConnectionDescriptor and it is different than a Pad that may have previously been sent, remove any existing outbound RTCP keys and generate new ones, based on the keys that are generated from both the End-End Secret (generated by the MTA) and the Pad (generated locally). The IPAT MUST re-initialize the RTCP sequence numbers if new keys are generated. The ciphersuites used for these outbound keys are those from the RemoteConnectionDescriptor parameter that is to be sent to the MTA.
3. If the RemoteConnectionDescriptor parameter is sent without a Pad, check if the first RTP ciphersuite field in the RemoteConnectionDescriptor parameter differs from the one that the IPAT originally selected. Also, check to see if a Pad has been previously sent. If the ciphersuites differ, or if a Pad has been previously sent, perform the following steps:
    a. Remove any existing outbound RTP key.
    b. If the new RTP ciphersuite is non-NULL, generate new outbound RTP keys and RTP timestamp from the same End-End Secret (generated by the MTA) as the last time, as specified in section 7.6.2.3.3.1
4. If the RemoteConnectionDescriptor parameter is sent without a Pad, check if the first RTCP ciphersuite field in the RemoteConnectionDescriptor parameter differs from the one that the IPAT originally selected. Also, check to see if a Pad has been previously sent. If the ciphersuites differ, or if a Pad has been previously sent, perform the following steps:
    a. Remove any existing outbound RTCP key.
    b. If the new RTCP ciphersuite is non-NULL, generate new outbound RTCP keys from the same End-End Secret (generated by the MTA) as the last time, as specified in section 7.6.2.3.3.1, and reset the RTCP sequence number to 0.
5. If the End-End Secret included in the RemoteConnectionDescriptor has changed or the negotiated RTP ciphersuite has changed, perform the following steps:
    a. Remove any existing inbound RTP keys.
    b. If the new list of RTP ciphersuites is non-NULL, generate new inbound RTP keys, based on the End-End Secret (generated locally) and the Pad (generated by the MTA), and generate a new RTP timestamp.
6. If the End-End Secret included in the RemoteConnectionDescriptor has changed or the negotiated RTCP ciphersuite has changed, perform the following steps:
    a. Remove any existing inbound RTCP keys.
    b. If the new list of RTCP ciphersuites is non-NULL, generate new inbound RTCP keys, based on the End-End Secret (generated locally) and the Pad (generated by the MTA), and reset the RTCP sequence number to 0.

7. Be ready to send RTCP messages to and receive RTCP messages from the MTA. If applicable under the current connection mode, be ready to send and receive RTP messages with the MTA.

- If, when receiving a response to a CreateConnection or ModifyConnection command that includes a LocalConnectionDescriptor, and the negotiated lists of ciphersuites for RTP and RTCP contain at least one non-NULL encryption or authentication algorithm each, an IPAT MUST:
    1. If a Pad was included in the LocalConnectionDescriptor and it is different than a Pad that may have previously been received, remove any existing inbound RTP keys and generate new ones, based on the keys that are generated from both the End-End Secret (generated locally) and the Pad (generated by the MTA). The IPAT MUST re-initialize the RTP timestamp if new keys are generated. The ciphersuites used for these inbound keys are taken from the LocalConnectionDescriptor parameter just received from the MTA.
    2. If a Pad was included in the LocalConnectionDescriptor and it is different than a Pad that may have previously been received, remove any existing inbound RTCP keys and generate new ones, based on the keys that are generated from both the End-End Secret (generated locally) and the Pad (generated by the MTA). The IPAT MUST re-initialize RTCP sequence numbers if new keys are generated. The ciphersuites used for these inbound keys are taken from the LocalConnectionDescriptor parameter just received from the MTA.
    3. If the LocalConnectionDescriptor parameter was received without a Pad, check if the first RTP ciphersuite field in the LocalConnectionDescriptor parameter differs from the one that the MTA originally selected. Also, check to see if a Pad had been previously received. If the ciphersuites differ, or if a Pad had been previously received, perform the following steps:
        a. Remove any existing inbound RTP key.
        b. If the new RTP ciphersuite is non-NULL, generate new inbound RTP keys and RTP timestamp from the same End-End Secret (generated locally) as the last time, as specified in section 7.6.2.3.3.1
    4. If the LocalConnectionDescriptor parameter was received without a Pad, check if the first RTCP ciphersuite field in the LocalConnectionDescriptor parameter differs from the one that the MTA originally selected. Also, check to see if a Pad had been previously received. If the ciphersuites differ, or if a Pad had been previously received, perform the following steps:
        a. Remove any existing inbound RTCP key.
        b. If the new RTCP ciphersuite is non-NULL, generate new inbound RTCP keys from the same End-End Secret (generated locally) as the last time, as specified in section 7.6.2.3.3.1, and reset the RTCP sequence number to 0.
    5. If the End-End Secret included in the LocalConnectionDescriptor has changed or the negotiated RTP ciphersuite has changed, perform the following steps:
        a. Remove any existing outbound RTP keys.

b. If the new list of RTP ciphersuites is non-NULL, generate new outbound RTP keys, based on the End-End Secret (generated by the MTA) and the Pad (generated locally), and generate a new RTP timestamp.

6. If the End-End Secret included in the LocalConnectionDescriptor has changed or the negotiated RTCP ciphersuite has changed, perform the following steps:
   a. Remove any existing outbound RTCP keys.
   b. If the new list of RTCP ciphersuites is non-NULL, generate new outbound RTCP keys, based on the End-End Secret (generated by the MTA) and the Pad (generated locally), and reset the RTCP sequence number to 0.
7. Be ready to send RTCP messages to and receive RTCP messages from the MTA. If applicable under the current connection mode, be ready to send and receive RTP messages with the MTA.

- If ciphersuite negotiation results in NULL encryption and authentication algorithms for RTP and RTCP, then the IPAT MUST remove any existing RTP and RTCP keys and do not perform security on the RTP and RTCP packets.

If an IPAT sends a RemoteConnectionDescriptor parameter, it MUST send the latest negotiated list of ciphersuites.

4 Call flow examples (informative)

4.1 Introduction

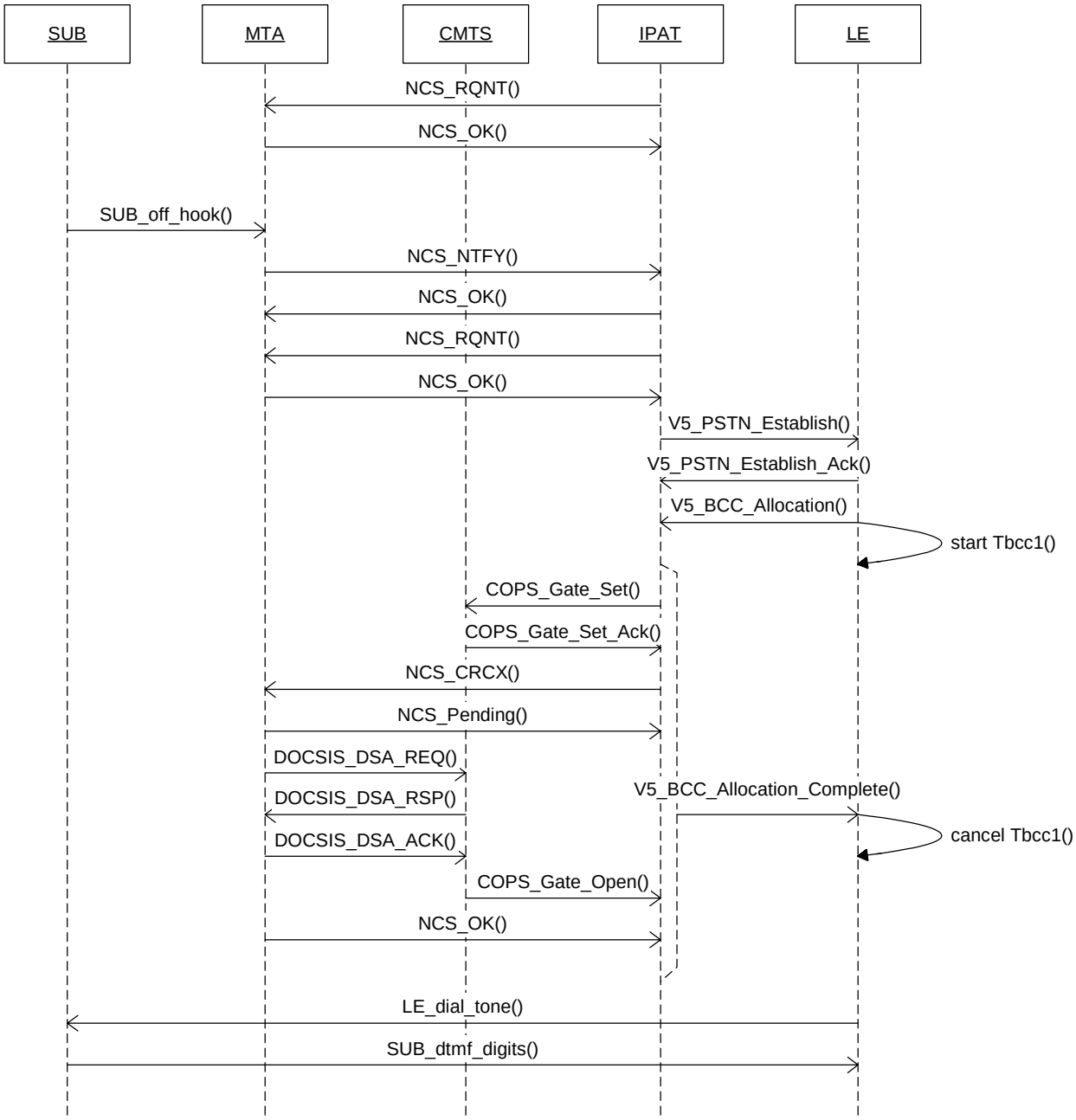
The following call flows identify messages that are exchanged between the subscriber (SUB), the media terminal adapter (MTA), the cable modem termination system (CMTS), the internet protocol access terminal (IPAT) and the local exchange (LE). The protocols used in these signals are identified in the first part of the message label when appropriate. For example, NCS_RQNT, means that the RQNT message is defined in the NCS protocol. The timers associated with the V5 PSTN signals are defined in ETSI EN 300 324-1. The timers associated with the V5 BCC signals are defined in ETSI EN 300 347-1.

A split component lifeline in the call flow diagrams stresses that there is no specified time relationship between the messaging on both lifeline paths.

4.2 Call origination

Call origination is the scenario in which the near-end subscriber initiates a call to a telephone user elsewhere.

4.2.1 Origination call flow



4.2.2 Origination call flow description

NCS_RQNT: Before the origination begins, the IPAT has sent a request for notification to the MTA telling it to report off hook events.

NCS_OK: The MTA acknowledges the request. These first two messages may occur any time before the subscriber goes off hook.

SUB_off_hook: The subscriber takes the phone off hook.

NCS_NTFY: The MTA sends an NCS message to the IPAT notifying it that it has observed the off hook event (O: hd).

NCS_OK: The IPAT acknowledges the notification.

NCS_RQNT: The IPAT sends a request to the MTA for notification of on hook and hook flash events detected by the MTA (R: hu, hf). This message may be piggybacked with the previous acknowledgement.

NCS_OK: The MTA acknowledges the request.

V5_PSTN_Establish: The IPAT sends a V5 PSTN Establish message to the LE. The Establish message contains the L3 address of the subscriber (the calling party).

V5_PSTN_Establish_Ack: The LE sends a V5 PSTN Establish Ack message to the IPAT, acknowledging receipt of the Establish message.

V5_BCC_Allocation: The LE sends a V5 BCC Allocation message to the IPAT. The message identifies the bearer channel that the IPAT and LE will use for the media portion of the call. The LE starts timer Tbcc1.

V5_BCC_Allocation_Complete: The IPAT sends a V5 BCC Allocation Complete message to the LE indicating that the bearer channel for the call has been reserved. This can happen at any time after the receipt of the BCC Allocation message. It is not mandatory, but it is allowed, to wait for the bandwidth to be established on the HFC network. The LE cancels timer Tbcc1 upon receipt of the Allocation Complete message.

COPS_Gate_Set: The IPAT sends a Gate-Set message to the CMTS over the managed IP network. This allocates and authorizes a gate in the CMTS.

COPS_Gate_Set_Ack: The CMTS acknowledges the Gate-Set message. The message contains the gate ID that the CMTS will use for the call.

NCS_CRCX: The IPAT sends a CreateConnection command to the MTA. The message includes the gate ID.

NCS_Pending: The MTA sends the IPAT a provisional response, acknowledging that it is working on the creation of the connection.

DOCSIS_DSA_REQ: The MTA sends the CMTS a DSA request. The message includes the gate ID. It is part of the three-way handshake used in the Dynamic Service Addition process.

DOCSIS_DSA_RSP: The CMTS sends the MTA a DSA response. It is part of the three-way handshake used in the Dynamic Service Addition process.

DOCSIS_DSA_ACK: The MTA sends the CMTS a DSA acknowledgement. It is part of the three-way handshake used in the Dynamic Service Addition process.

COPS_Gate_Open: The CMTS sends the IPAT a Gate-Open message, indicating that the MTA has committed the gate resources.

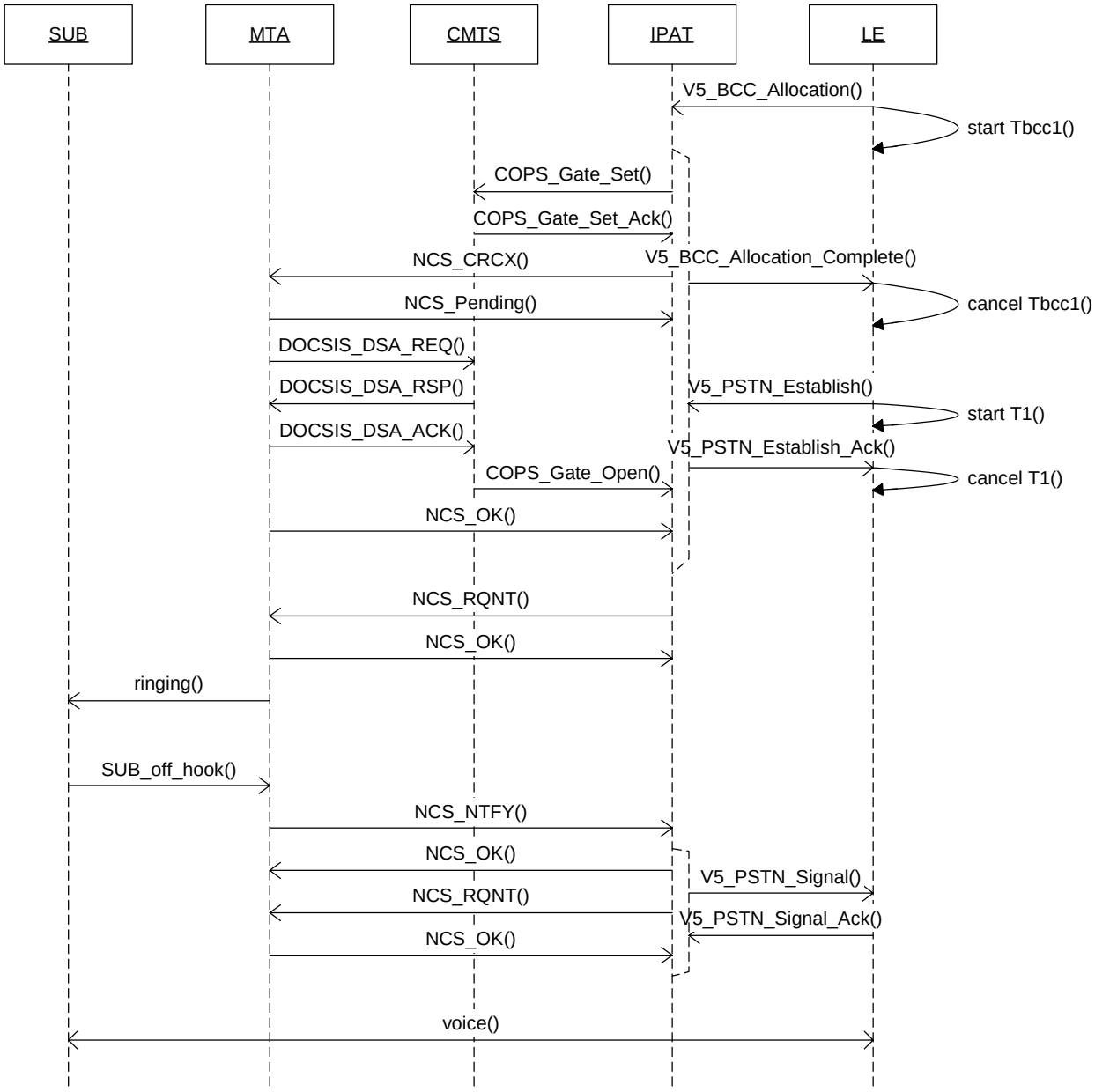
NCS_OK: The MTA sends the IPAT a final response, indicating that the connection previously requested has been created.

A two way voice path has now been established. The LE sends a dial tone over this path, and the subscriber, upon hearing the dial tone, begins to send DTMF digits toward the LE.

4.3 Call termination

Call termination is the scenario in which the subscriber receives a call which was initiated by a telephone user elsewhere.

4.3.1 Termination call flow



4.3.2 Termination call flow description

V5_BCC_Allocation: In response to an incoming call from a far-end caller, the LE begins the process of allocating bandwidth for the call by sending the IPAT a V5 BCC Allocation message. The message identifies the bearer channel that the IPAT and LE will use for the media portion of the call. The LE starts timer Tbcc1.

V5_BCC_Allocation_Complete: The IPAT sends a V5 BCC Allocation Complete message to the LE indicating that the bearer channel for the call has been reserved. This can happen at any time after the receipt of the BCC Allocation message. It is not mandatory, but it is allowed, to wait for the bandwidth to be established on the HFC network. The LE cancels timer Tbcc1 upon receipt of the Allocation Complete message.

V5_PSTN_Establish: Some time after receiving the BCC Allocation Complete message, the LE sends a V5 PSTN Establish message to the IPAT. The Establish message contains the L3 address of the subscriber (the called party), and an indication to start ringing. Ringing will only happen after HFC bandwidth is reserved and committed. The LE starts timer T1.

V5_PSTN_Establish_Ack: The IPAT sends a V5 PSTN Establish Ack message to the LE, acknowledging receipt of the Establish message. It is not mandatory, but it is allowed, to wait for the bandwidth to be established on the HFC network before sending this message. The LE cancels timer T1.

COPS_Gate_Set: The IPAT sends a Gate-Set message to the CMTS over the managed IP network. This allocates and authorizes a gate in the CMTS.

COPS_Gate_Set_Ack: The CMTS acknowledges the Gate-Set message. The message contains the gate ID that the CMTS will use for the call.

NCS_CRCX: The IPAT sends a CreateConnection command to the MTA. The message includes the gate ID.

NCS_Pending: The MTA sends the IPAT a provisional response, acknowledging that it is working on the creation of the connection.

DOCSIS_DSA_REQ: The MTA sends the CMTS a DSA request. The message includes the gate ID. It is part of the three-way handshake used in the Dynamic Service Addition process.

DOCSIS_DSA_RSP: The CMTS sends the MTA a DSA response. It is part of the three-way handshake used in the Dynamic Service Addition process.

DOCSIS_DSA_ACK: The MTA sends the CMTS a DSA acknowledgement. It is part of the three-way handshake used in the Dynamic Service Addition process.

COPS_Gate_Open: The CMTS sends the IPAT a Gate-Open message, indicating that the MTA has committed the gate resources.

NCS_OK: The MTA sends the IPAT a final response, indicating that the connection previously requested has been created.

NCS_RQNT: The IPAT sends the MTA a notification request to report off hook events. The message also contains a signal request that instructs the MTA to ring the subscriber's phone (S:<ringing signal>, where <ringing signal> can be an rx signal defined in the L package).

NCS_OK: The MTA acknowledges the request, and rings the subscriber's phone.

SUB_off_hook: The subscriber takes the phone off hook. The MTA must stop ringing locally without waiting for a message from the IPAT.

NCS_NTFY: The MTA sends an NCS message to the IPAT notifying it that it has observed the off hook event (O: hd).

NCS_OK: The IPAT acknowledges the notification. At the same or a later time, the IPAT will send a PSTN Signal message to the LE (see further).

NCS_RQNT: The IPAT sends a request to the MTA for notification of on hook and hook flash events detected by the MTA (R: hu, hf). This message may be piggybacked with the previous acknowledgement.

NCS_OK: The MTA acknowledges the request.

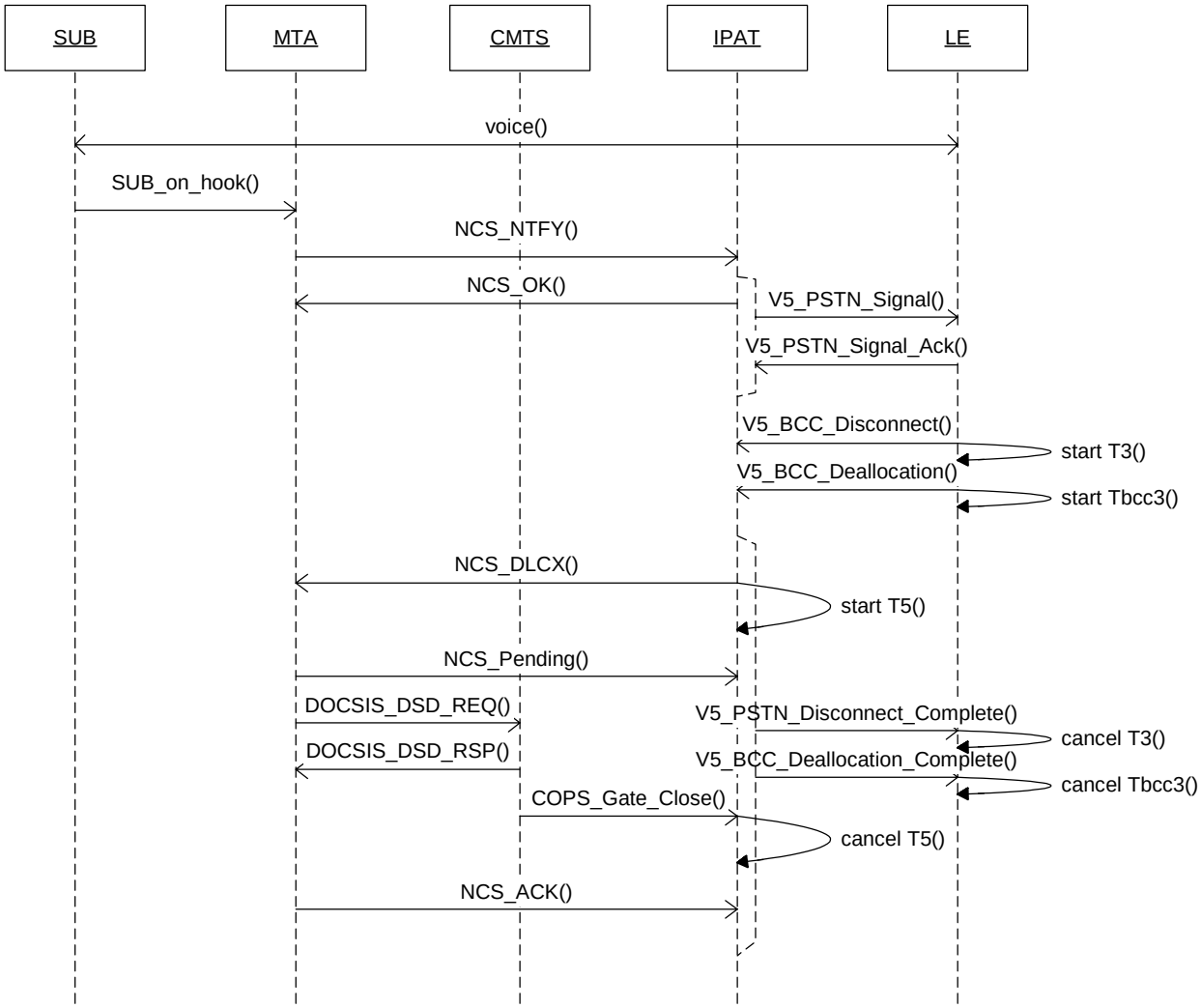
V5_PSTN_Signal: After the MTA notified the IPAT of the off hook event, the IPAT sends a PSTN Signal message to the LE. The message notifies the LE that the called party has gone off hook.

V5_PSTN_Signal_Ack: The LE acknowledges receipt of the Signal message.

At this point, a two way audio path exists between the calling and the called parties.

4.4 Call disconnection

4.4.1 Disconnection call flow



4.4.2 Disconnection call flow description

SUB_on_hook: A media path exists between the called and the calling party. The subscriber goes on hook.

NCS_NTFY: The MTA detects the on hook event and sends a notification to the IPAT as it was previously instructed to do in a RQNT message from the IPAT (not shown here). The NTFY message contains the observed event (O:hu).

NCS_OK: The IPAT acknowledges receipt of the notification. At the same time, the IPAT will have to notify the LE of the on hook state.

V5_PSTN_Signal: The IPAT sends a V5 PSTN Signal message to the LE indicating that the subscriber is on hook. The message contains the L3 address of the subscriber.

V5_PSTN_Signal_Ack: The LE acknowledges the receipt of the Signal message.

V5_PSTN_Disconnect: The LE sends a PSTN Disconnect message to the IPAT instructing it to disconnect the call. The message contains the L3 address of the subscriber. The LE starts timer T3.

V5_BCC_Deallocation: The LE sends a BCC Deallocation message to the IPAT instructing it to deallocate the bearer channel for the call. The LE starts timer Tbcc3.

V5_PSTN_Disconnect_Complete: The IPAT sends a V5 PSTN message to the LE indicating that the state of the call has been returned to idle. It is not mandatory, but it is allowed, to wait for the bandwidth to be released on the HFC network before sending this message. The LE cancels timer T3.

V5_BCC_Deallocation_Complete: The IPAT sends a V5 BCC message to the LE indicating that the time slot allocated to the call has been released. It is not mandatory, but it is allowed, to wait for the bandwidth to be released on the HFC network before sending this message. The LE cancels timer Tbcc3.

NCS_DLCX: The IPAT sends an NCS DeleteConnection command to the MTA. The IPAT also starts timer T5. If the IPAT does not receive a Gate-Close message from the CMTS within this time, it must delete the gate on the CMTS itself by using the appropriate Gate-Delete message.

NCS_Pending: The MTA sends a provisional response to acknowledge the receipt of the DeleteConnection command, and to indicate that it is in the process of releasing HFC bandwidth.

DOCSIS_DSD_REQ: The MTA sends the CMTS a DSD request. The message identifies the service flow to be deleted.

DOCSIS_DSD_RSP: The CMTS sends a DSD response message to the MTA to indicate the deletion of the service flow.

COPS_Gate_Close: The CMTS sends the IPAT a Gate-Close message, indicating that the gate has been deleted. The IPAT cancels timer T5.

NCS_OK: The MTA sends the IPAT a final response, indicating that the connection has been deleted as requested.