INTERNATIONAL TELECOMMUNICATION UNION

ITU-T H.323 System Implementers Guide

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(30 May 2003)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Communication procedures

Implementors Guide for Recommendations of the H.323 System:

H.323, H.225.0, H.245, H.246, H.283, H.235, H.341, H.450 Series, H.460 Series, and H.500 Series

Summary

This document is a compilation of reported defects identified in the versions of ITU-T Recommendation H.323 and its related Recommendations currently in force. It must be read in conjunction with the Recommendations to serve as an additional authoritative source of information for implementers. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.323-series Recommendations.

This revision contains all updates submitted upto and including those at Study Group 16 meeting in May, 2003, in Geneva. This Implementor's Guide provides corrections and clarifications for implementations of the H.323 system V.4 and supersedes the earlier version approved by Study Group 16 in October 2002.

Change Log

N/A

Contact Information

ITU-T Study Group 16 / Rapporteur Question 2/16	Paul E. Jones Cisco Systems, Inc. 7025 Kit Creek Road Research Triangle Park, NC 27709. USA	Tel: Fax: E-mail:	+1 919 392 6948 +1 919 392 2177 paulej@packetizer.com
ITU-T Study Group 16 / Rapporteur Question 3/16	Christian Groves Ericsson Australia Pty. Ltd. 37/360 Elizabeth Street Victoria 3000. Australia	Tel: Fax: E-mail:	+61 3 9301 6116 +61 3 9301 1499 Christian.Groves@ericsson.com
ITU-T Study Group 16 / Rapporteur Question G/16 Editor ITU-T Rec. H.235	Martin Euchner Siemens AG / ICN M NT 18 Hofmannstr. 51 D-81359 Muenchen. Germany	Tel: Fax: E-mail:	+49 89 722 5 57 90 +49 89 722 4 68 41 martin.euchner@ties.itu.int
Editor ITU-T Rec. H.341	Craig Blasberg Cisco Systems, Inc. 7025 Kit Creek Road Research Triangle Park, NC 27709. USA	Tel: Fax: E-mail:	+1 919 392 5760 +1 919 392 6801 blasberg@cisco.com
Editor ITU-T Rec. H.225.0 Editor ITU-T Rec. H.323 and Implementer's Guide	Vivek Bhargava Cisco Systems, Inc. 7025 Kit Creek Road Research Triangle Park, NC 27709. USA	Tel: Fax: E-mail:	+1 919 392 6823 +1 919 392 2177 vbhargava@cisco.com
Editor ITU-T Rec. H.225.0 Annex G	Miner Gleason Cisco Systems, Inc. 7025 Kit Creek Road Research Triangle Park, NC 27709. USA	Tel: Fax: E-mail:	+1 919 392 8752 +1 919 392 7065 mgleason@cisco.com
Editor ITU-T Rec. H.245	Mike Nilsson BT Labs Ipswitch. United Kingdom	Tel: Fax: E-mail:	+44 1 473 645413 +44 1 473 643791 mike.nilsson@bt.com
ITU-T Study Group 16 / Rapporteur Question 5/16 & Editor ITU-T Rec. for H.450.{1,2,3,4,5, 6,9,10,11,12}	Ernst Horvath Siemens Austria Gudrunstrasse 11 A-1101 Vienna. Austria	Tel: Fax: E-mail:	+43 5 1707 45897 +43 5 1707 56992 ernst.horvath@siemens.at
Editor ITU-T Rec. H.450.7	Dave Walker SS8 Networks 135 Michael Cowpland Drive, Suite 200 Kanata, Ontario, K2M 2E9. Canada	Tel: Fax: E-mail:	+1 613 592 8450 +1 613 592 9634 dwalker@ss8networks.com
Editor ITU-T Rec. H.450.8	Glen Freundlich Avaya Communication 1300 W. 120th Avenue Westminster, CO 80234. USA	Tel: Fax: E-mail:	+1 303 538 2899 +1 303 538 3007 ggf@avaya.com
Editor ITU-T Rec. H.460.1	P. Cordell	E-mail:	pete@tech-know-ware.com
Editor ITU-T Rec. H.460.4	Gary Thom	E-mail:	gthom@delta-info.com
Editor ITU-T Rec. H.460.5	Sasha Ruditsky	E-mail:	sasha@radvision.com
Editor ITU-T Rec. H.460.6	Bob Gilman	E-mail:	rrg@avaya.com
Editor ITU-T Rec. H.460.{2,7,8}	Paul Jones (see above for Q.2/16)	E-mail:	paulej@packetizer.com
Editor ITU-T Rec. H.460.3	Louis Fourie	E-mail:	lfourie@cisco.com
Editor ITU-T Rec. H.460.9	Ernst Horvath (see above for Q.5/16)	E-mail:	ernst.horvath@siemens.at

Note: Not all Recommendations indicated above have IG issues in this document. The information above is provided for completeness.

Table of Contents

REVISED IMPLEMENTERS GUIDE FOR ITU-T H.323, H.225.0, H.245, H.246, H.283, H.235, H.341, H.450 SERIES, H.460 SERIES, AND H.500 SERIES RECOMMENDATIONS

1 Scope

This guide resolves defects in the following categories:

- editorial errors
- technical errors, such as omissions and inconsistencies
- ambiguities

In addition, the Implementers Guide may include explanatory text found necessary as a result of interpretation difficulties apparent from the defect reports.

This Guide will not address proposed additions, deletions, or modifications to the Recommendations that are not strictly related to implementation difficulties in the above categories. Proposals for new features should be made in through contributions to the ITU-T.

2 Introduction

This document is a compilation of reported defects identified in the versions of ITU-T Recommendation H.323 and its related Recommendations currently in force. It must be read in conjunction with the Recommendations to serve as an additional authoritative source of information for implementers. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.323-series Recommendations.

Upon discovering technical defects with any components of the H.323 Recommendations series, please provide a written description directly to the editors of the affected Recommendations with a copy to the Q13/16 or Q14/16 Rapporteur. The template for a defect report is located at the end of the Guide. Contact information for these parties is included at the front of the document. Return contact information should also be supplied so a dialogue can be established to resolve the matter and an appropriate reply to the defect report can be conveyed. This defect resolution process is open to any interested party. Formal membership in the ITU is not required to participate in this process.

4 References

This document refers to the following H.323 series Recommendations:

– ITU-T Recommendation H.323 (2000), Packet-Based multimedia communications systems

– ITU-T Recommendation H.323 – Annex R (2001), Robustness Methods for H.323 Entities

– ITU-T Recommendation H.225.0 (2000), Call signaling protocols and media stream packetization for packet based multimedia communications Systems

– ITU-T Recommendation H.225.0 – Annex G (2002), Communication Between Administrative Domains

– ITU-T Recommendation H.245 (2003), Control protocol for multimedia communication

– ITU-T Recommendation H.246 (1998), Interworking of H-Series multimedia terminals with H-Series multimedia terminals and voice/voiceband terminals on GSTN and ISDN

– ITU-T Recommendation H.246 – Annex C (2000), ISDN User Part Function - H.225.0 Interworking

– ITU-T Recommendation H.235 (2000), Security and encryption for H Series (H.323 and other H.245 based) multimedia terminals

- ITU-T Recommendation H.450.1 (1998), Generic functional protocol for the support of supplementary services in H.323
- ITU-T Recommendation H.450.2 (1998), Call transfer supplementary service for H.323
- ITU-T Recommendation H.450.3 (1998), Call diversion supplementary service for H.323
- ITU-T Recommendation H.450.4 (1999), Call hold supplementary service for H.323
- ITU-T Recommendation H.450.5 (1999), Call park and call pickup supplementary services for H.323
- ITU-T Recommendation H.450.6 (1999), Call waiting supplementary service for H.323
- ITU-T Recommendation H.450.7 (1999), Message waiting indication supplementary service for H.323
- ITU-T Recommendation H.450.8 (2000), Name identification supplementary service for H.323
- ITU-T Recommendation H.450.9 (2000), Call Completion Supplementary Services For H.323
- ITU-T Recommendation H.450.10 (2001), Call offer supplementary service for H.323
- ITU-T Recommendation H.450.11 (2001), Call intrusion supplementary services
- ITU-T Recommendation H.450.12 (2001), Call Information Additional Network Feature for H.323
- ITU-T Recommendation H.460.5 (2002), H.225.0 transport of multiple Q.931 IE of the same type
- ITU-T Recommendation H.460.6 (2002), Extended Fast Connect Feature
- ISO/IEC 11571 (1998), Information technology – Telecommunications and information exchange between systems – Private Integrated Services Networks – Addressing
- ITU-T Recommendation Q.931 (1998), ISDN user-network interface layer 3 specification for basic call control
- ITU-T Recommendation H.283, Remote device control logical channel transport

5 Nomenclature

In addition to traditional revision marks, the following marks and symbols are used to indicate to the reader how changes to the text of a Recommendation should be applied:

Symbol	Description
[Begin Correction]	Identifies the start of revision marked text based on extractions from the published Recommendations affected by the correction being described.
[End Correction]	Identifies the end of revision marked text based on extractions from the published Recommendations affected by the correction being described.
...	Indicates that the portion of the Recommendation between the text appearing before and after this symbol has remained unaffected by the correction being described and has been omitted for brevity.

--- SPECIAL INSTRUCTIONS --- {instructions}	Indicates a set of special editing instructions to be followed.

6 Technical and Editorial Corrections to H.323 Series Recommendations

6.1 Technical and Editorial Corrections to ITU-T Recommendation H.323 (2000)

6.1.1 H.323 Annex L Section 3.4

Description:	Modification to section 3.4 of H.323 Annex L to use PER/Text encoding scheme for h248Message.

[Begin Correction]

3.4 Encoding

...

H.248 signalling may be either binary (H.248 Annex A syntax, but using PER for encoding) or text (H.248 Annex B) based. The default is binary encoding. The presence of the isText field shall be used to indicate that H.248 Annex B encoding has been used for the H.248 descriptors in the StimulusControl structure.

...

[End Correction]

6.1.2 Calling party address information Correction

Description:	The H.323v4 specification is not clear on how to transport calling party address when the address is of the form of a number belonging to a private numbering plan. The text below clarifies the issue. This clarification applies to H.323v2 and H.323v3 also.

[Begin Correction]

7.8.2.1 Calling party address information

Calling party address information appears in the Setup message.

When address information represents a telephone number, the relevant information may appear in the Calling Party Number IE. This IE contains the caller's number, information about the number, and presentation and screening indicators found in octet 3a. This is the recommended mode of operation for the case where a PSTN Gateway sends a Setup message on the packet network.

Alternatively, calling party information may appear in the **sourceAddress**, **presentationIndicator**, and **screeningIndicator** fields of the Setup message. This mode of operation is required when the **sourceAddress** is not in any form of telephone number (i.e., **sourceAddress** is not type a **dialedDigits** or **partyNumber**). In accordance with 7.2.2.6 of H.225.0, it is also required when the address information is in the form of a telephone number belonging to a Private Numbering Plan.

[End Correction]

[Begin Correction]

7.8.3.1 Gateway as originating endpoint

In the case of a Setup message received by a Gateway from the ISDN, the caller's number and presentation information reside in the Calling Party Number IE. The Gateway shall send a Setup message on the packet network with the Calling Party Number IE containing the same information as was found in the Setup message from the SCN with the following exception. If the Numbering Plan Identification field contains value Private Numbering Plan, the digits shall be omitted from the Calling Party Number IE in accordance with 7.2.2.6 of H.225.0. In this exception case the Gateway shall place the received caller identification information in the sourceAddress, presentationIndicator and screeningIndicator fields in the Setup message. If the Gateway has the knowledge to send both a PNP Number and an E.164 Number, the Calling Party Number IE shall convey the E.164 Number (and not the "empty" PNP number).

[End Correction]

[Begin Correction]

7.8.4.1 Gateway as terminating endpoint

A PSTN Gateway in receipt of a Setup message from the packet network shall copy the information found in the Calling Party Number IE from the Setup message to the signalling format supported in the PSTN. For example, this information would be copied to the Calling Party Number IE of the Q.931 Setup message for ISDN. If the Calling Party Number IE is not present in the Setup message, or if the Numbering Plan Identification field contains the value Private Numbering Plan, the Gateway shall form the Calling Party Number IE using the **sourceAddress** (assuming it is one of the telephone number alias types), **presentationIndicator**, and **screeningIndicator** from the Setup message.

[End Correction]

6.1.3 Status/Status Inquiry messages without explicit Call Identifiers

Description:	Clarification is needed with respect to handling Status and Status Inquiry messages that do not have an explicit call identifier or which are not related to a specific call.

[Begin Correction]

7.3 Call signalling channel

...

An entity that is capable of processing multiple concurrent calls on the Call Signalling Channel may indicate that it will support no additional calls on the signalling channel by sending Release Complete with **newConnectionNeeded** as the reason. An entity that receives Release Complete with **newConnectionNeeded** can attempt to connect a new Call Signalling Channel.

An entity may transmit a Status Inquiry message that is not related to a specific call. In such cases, the entity shall set the **callIdentifier** field to all zeros. An entity shall not omit the **Status-UUIE** in the Status message or the **StatusInquiry-UUIE** in the Status Inquiry message when transmitting those messages, but entities shall be prepared to receive messages not containing those message elements in order to maintain backward compatibility.

[End Correction]

6.1.4 Corrections to the H.323 URL Syntax

Description:	The syntax currently used in H.323 version 4 for the H.323 URL contains several syntax errors. In addition, as a result of a number of editorial changes, two productions defined in the ABNF are no longer used and should be removed. The syntax errors are in the productions "user" and "url-parameter". The intent was to allow one or more characters to be used, but the syntax currently restricts those to a single character. In addition, the syntax for value ranges is incorrect. The productions that are no longer used are "unreserved" and "mark".

[Begin Correction]

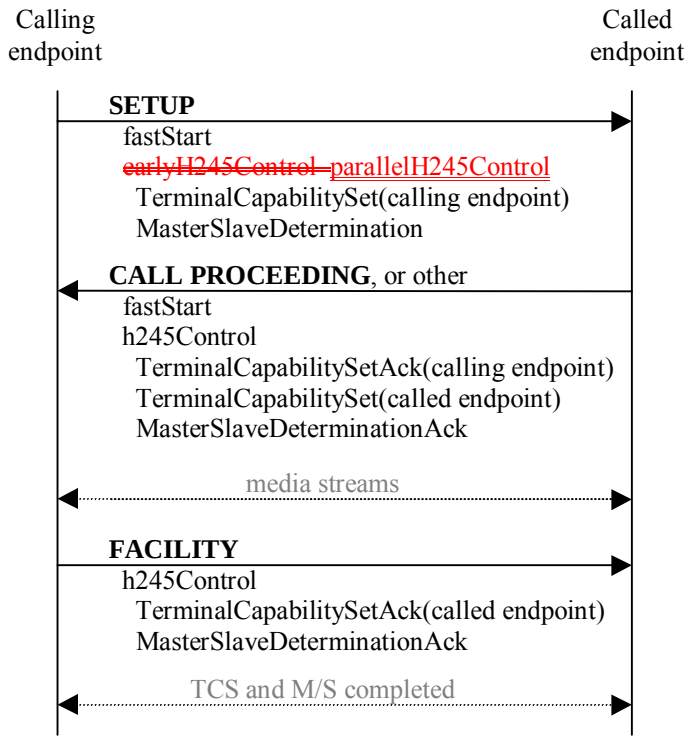
```
H323-URL              =     "h323:" address [ url-parameters ]
address               =     user / "@" hostport / user "@" hostport
user                  =     1*(%x21-24 / %x26-3F / %x41-7F / escaped)
                                  ; The symbols "%", "@", and symbols with a
                                  ; character value below 0x21 may be represented
                                  ; as escaped sequences.
hostport              =     host [ ":" port]
host                  =     hostname / IPv4address / IPv6reference
hostname              =     *( domainlabel "." ) toplabel [ "." ]
domainlabel           =     alphanum / alphanum *( alphanum / "-" ) alphanum
toplabel              =     ALPHA / ALPHA *( alphanum / "-" ) alphanum
IPv4address           =     1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
IPv6reference         =     "[" IPv6address "]"
IPV6address           =     hexpart [ ":" IPv4address ]
hexpart               =     hexseq / hexseq "::" [ hexseq ] / "::" [ hexseq ]
hexseq                =     hex4 *( ":" hex4 )
hex4                  =     1*4HEXDIG
port                  =     1*DIGIT
url-parameters        =     *( ";" url-parameter )
url-parameter         =     1*(%x21-24 / %x26-3A / %x3C-7F / escaped)
                                  ; Specific parameter definitions are for further
                                  ; study.    The symbols "%", ";", and symbols with
                                  ; a character value below 0x21 may be
                                  ; represented as escaped sequences.
unreserved            =     alphanum / mark
alphanum              =     ALPHA / DIGIT
mark                  =     "-" / "_" / "." / "!" / "~" / "*" / "'"
                      /     "(" / ")" / "&" / "=" / "+" / "$" / ","
escaped               =     "%" HEXDIG HEXDIG
```

[End Correction]

6.1.5 H.323v4 Editorial Correction

Description:	The SETUP message in Figure 41 includes the term "earlyH245Control". The term "earlyH245Control" would be incorrect and should be replaced with the correct term "parallelH245Control".

[Begin Correction]



[End Correction]

6.1.6 Pairing of RTP streams for a common bi-directional RTCP channel

Description:	The existing text in the document is ambiguous in describing how a slave endpoint can open an RTP channel other than the defined values for primary audio, video, and data. The following text clarifies this procedure.

[Begin Correction]

6.2.8.2 Logical Channel Signalling

…

If a corresponding reverse channel is opened for a given existing RTP session (identified by the RTP **sessionID**), the **mediaControlChannel** Transport Addresses exchanged by the **openLogicalChannel** process shall be identical to those used for the forward channel. **sessionID** values of 1, 2 and 3 are pre-assigned for primary audio, video and data sessions, respectively. Even the slave endpoint can open logical channels for these primary sessions without negotiating the **sessionID** value with the master endpoint. The master endpoint can open any additional session with a particular **sessionID** value greater than 3. The slave endpoint can open a corresponding session with the given **sessionID**. Otherwise, the slave endpoint can open additional sessions with **sessionID**=0 in the **openLogicalChannel** message, but it shall acquire the actual **sessionID** value from the master endpoint's **openLogicalChannelAck** message. Should a collision occur where both ends attempt to establish conflicting RTP sessions at the same time, the master endpoint shall reject the conflicting attempt as described in Recommendation H.245. The rejected **openLogicalChannel** attempt may then be retried at a later time.

[End Correction]

6.1.7 H.323 Annex M.1 Section 3

Description:	If a tunneled QSIG PROGRESS message contains an indication of the presence of tones or announcements it should be possible to convey these tones/announcements over a logical channel from called to calling side. This requirement is signalled by including a Progress Indicator IE in a backward message (e.g. ALERTING), or in a PROGRESS message if no other message is appropriate. H.323 Annex M.1 recommends tunneling a QSIG PROGRESS message in an H.225.0 FACILITY message, but the FACILITY message cannot contain a Progress Indicator. Therefore the QSIG PROGRESS message should be tunneled in an H.225.0 PROGRESS message in this case.

[Begin Correction]

3. Endpoint Procedures

…

Also since the NOTIFY and PROGRESS messages are optional, they might not be delivered end-to-end and should be tunnelled in a FACILITY message unless tones or announcements are provided by the called side and no Progress indicator has been sent to the calling side so far. In this case a PROGRESS message (with Progress descriptor #1 or #8) should be used to tunnel a QSIG PROGRESS message.

...

Table 1/Annex M.1 – Mapping between QSIG messages and H.225.0 messages

QSIG message	H.225.0 message
SETUP	SETUP
ALERTING	ALERTING
CONNECT	CONNECT
RELEASE COMPLETE	RELEASE COMPLETE
CALL PROCEEDING	FACILITY
FACILITY	
PROGRESS (Note)	
NOTIFY	
DISCONNECT	
RELEASE	
all other messages…	
Note: If tones or announcements are provided by the called side this message should be tunnelled in a PROGRESS message rather than in FACILITY.	

...

[End Correction]

6.1.8 Editorial Correction

Description:	H.323v4 states that the H.245 recipient shall not alter the characteristics of the received OLC. However, H.235 practices such that the H.245 recipient includes encryptionSync with a conveyed session key as part of the returned OLC. While the included encryptionSync does not relate to the characteristics of the OLC, the OLC structure is in fact modified. This matter is clarified with the following text.

[Begin Correction]

8.1.7.1 Proposal, selection and opening of media channels

NOTE – The called endpoint is only allowed to alter fields in a proposed **OpenLogicalChannel** structure as specified in this section. An endpoint is not allowed, for example, to alter the number of frames per packet or other characteristics of the proposed channel not specifically stated in this section. If the calling endpoint wants to increase the likelihood that the Fast Connect can be accepted, it should include multiple proposals with different alternative parameters. This rule does not preclude an endpoint from including **encryptionSync** in the returned **OpenLogicalChannel**.

[End Correction]

6.1.9 Normative References Update

Description:	H.323v4 document refers to an old version of the H.235 Recommendation. This reference has been corrected by the text below.

[Begin Correction]

2 Normative references

…

[22] ITU-T Recommendation H.235 (~~1998~~2000), *Security and encryption for H-Series (H.323 and other H.245 based) multimedia terminals.*

[End Correction]

6.1.10 Sending of BRQ messages

Description:	H.323 recommends that the Gatekeeper should be informed whenever there is an increase or decrease in bandwidth usage. However, it mandates timing restrictions on when a decrease in bandwidth should be reported. This is deemed un-necessary and it should be left to the endpoint on how they want to support this reporting. The text below shows the corrections.

[Begin Correction]

8.4.1 Bandwidth changes

…

The endpoint shall send a BRQ message to the Gatekeeper whenever bandwidth utilization decreases below that which was specified in the original ARQ or the last BRQ or BCF message. The endpoint shall also send a BRQ message to the Gatekeeper whenever logical channel signalling results in the addition or removal of a unique multicast stream to or from the endpoint. ~~While an endpoint shall send a BRQ each time bandwidth needs to be increased, an endpoint shall wait for a period of five seconds before sending a BRQ message to indicate a decrease in bandwidth utilization. Only after the bandwidth utilization has not changed for a period of five seconds shall the endpoint transmit a BRQ to indicate lower bandwidth utilization.~~

…

[End Correction]

6.1.11 Third Party Initiated Pause and Re-routing

Description:	New paragraphs are added to clarify the sending and receiving of an empty capability set.

[Begin Correction]

8.4.6 Third party initiated pause and re-routing

…

While in the "transmitter side paused" state, an endpoint shall not initiate the opening of any logical channels, but shall accept the opening and closing of logical channels from the remote end based on the usual rules and shall continue to receive media on open logical channels opened by the remote endpoint. This allows endpoints to receive announcements (e.g. pre-connect call progress) where the announcing entity does not wish to receive media from the endpoint. A **terminalCapabilitySet** message may be sent whenever an endpoint's capabilities change, including when the endpoint is in the "transmitter side paused" state. This allows communication to be established between two endpoints that initially do not declare any capabilities.

An endpoint in “transmitter side paused” state may also put the other endpoint in the call into a “transmitter side paused” state by transmitting an empty capability set message. Upon reception of the empty capability set message, the receiver shall adhere to the procedures defined in this section.

An endpoint shall leave the "transmitter side paused" state on reception of any **terminalCapabilitySet** message, other than an empty capability set. On leaving this state, an endpoint shall reset its H.245 state to that which it was in just after the H.245 transport connection was made at call establishment time (i.e. the beginning of phase B), but shall preserve state information relating to any logical channels that are open. This puts the endpoint in a known H.245 state after the pause. This allows an endpoint to be connected to a different endpoint when it is released from the paused state.

After leaving the "transmitter side paused" state, an endpoint shall proceed with normal H.245 procedures: it shall take part in master/slave determination signalling and may proceed with normal open logical channel signalling procedures. When an MC leaves the "transmitter side paused" state, it shall act as if a new endpoint has entered the conference.

If an endpoint in a “transmitter side paused” state had also transmitted an empty capability set in order to put the other end in “transmitter side paused” state, it shall assume that it is still in a paused state until it receives a non-empty capability set from the other side when it releases the other endpoint from the paused state. The paused endpoint shall be prepared to receive OLCs from the other endpoint.

Unless its capabilities have changed, an endpoint need not resend a capability set as the Gatekeeper will have supplied this to the remote endpoint to remove any paused state in the remote endpoint. This option of not sending a capability set enables faster reconnection. If the first **terminalCapabilitySet** message sent by an endpoint after leaving the "transmitter side paused" state differs from the capability set that the Gatekeeper provided to the remote endpoint, the Gatekeeper shall signal the remote endpoint to remove capabilities which were not indicated by the initiating endpoint.

[End Correction]

6.1.12 Fast Connect Session IDs

Description:	During the work on Modem over IP, it became apparent that there are legitimate cases wherein an H.323 entity may need to use session IDs that are outside the range of the “well known” session IDs (1, 2, and 3) in a Fast Connect proposal. Unfortunately, the current text in H.323 prohibits using any but the well known session IDs. This is corrected as below.

[Begin Correction]

8.1.7.2 Switching to H.245 procedures

When H.245 procedures are activated, all mandatory procedures of H.245 that normally occur upon initiation of an H.245 connection shall be completed prior to initiation of any additional H.245 procedures. The media channels that were established in the Fast Connect procedure are "inherited" as though they had been opened using normal H.245 **openLogicalChannel** and **openLogicalChannelAck** procedures. ~~In order for such "inheritance" to succeed, media sessions opened during the Fast Connect procedure shall use only well-known **sessionID** values defined in ITU-T Rec. H.245.~~

[End Correction]

6.1.13 Restart in RRQ

Description:	The current text in H.323 implies that the Registration Request may be sent only when an endpoint powers up. However, the intent is that endpoints can send RRQ at any time. This is corrected in the text as follows below.

[Begin Correction]

7.2.2 Endpoint Registration

…

An endpoint shall send a Registration Request (RRQ) to a Gatekeeper. This is sent to the Gatekeeper's RAS Channel Transport Address. The endpoint has the Network Address of the Gatekeeper from the Gatekeeper discovery process and uses the well-known RAS Channel TSAP

Identifier. The Gatekeeper shall respond with either a Registration Confirmation (RCF) or a Registration Reject (RRJ). See Figure 24. An endpoint shall only register with a single Gatekeeper.

The RRQ may be repeated periodically (~~i.e.~~e.g. at terminal power-up), so the Gatekeeper shall be able to handle multiple requests from the same endpoint. If a Gatekeeper receives an RRQ having the same alias address (or list of alias addresses) and the same Transport Addresses as an active registration, it shall respond with RCF. If a Gatekeeper receives an RRQ having the same alias address (or list of alias addresses) as an active registration and different Transport Addresses, it may confirm the request, if it complies with the Gatekeeper's registration policy. If the request does not comply with the Gatekeeper's registration policy, the Gatekeeper should reject the registration indicating a duplicate or invalid registration. If the Gatekeeper receives an RRQ having the same Transport Addresses as an active registration and a different alias address (or list of alias addresses) and the RRQ is not specified to be an additive RRQ, it should replace the translation table entries. The Gatekeeper may have a method to authenticate these changes.

[End Correction]

6.1.14 Alternate Gatekeeper Procedures in URQ

Description:	The **altGKisPermanent** field is not present in the URQ message. Its usage from the following text in H.323 should be deleted.

[Begin Correction]

7.2.6 Alternate gatekeeper procedures

…

A Gatekeeper may send a URQ to an endpoint with a list of Alternate Gatekeepers, in which case the endpoint shall respond with a UCF and attempt to communicate with an Alternate Gatekeeper. ~~The endpoint shall ignore the values of the **needToRegister** and **altGKisPermanent** fields and assume that those values are TRUE.~~ An endpoint shall not include a list of Alternate Gatekeepers in any URQ message that it sends.

…

[End Correction]

6.1.15 Clarification on usage of RFC 2833 in fast connect by using parallel H.245 procedure

Description:	The fact that the signalling of the capability of using RFC 2833 for carrying DTMF and other telephone tones in RTP can be supported with fast connect by using parallelH245 procedures is not obvious. The text below makes this possibility more obvious. In addition “receiveRTPAudioTelephoneEventCapability” should be “receiveRTPAudioTelephonyEventCapability”

[Begin Correction]

10.5 Use of RTP payload for DTMF digits, telephony tones and telephony signals

It is possible to carry DTMF tones, fax-related tones, standard subscriber line tones, country-specific tones and trunk events using a distinct dynamic RTP payload type in the same RTP stream as the media. Many applications, such as IVR systems and voice systems rely on synchronization of DTMF input.

RFC 2833 describes means for transporting these tones and events over RTP. An endpoint may indicate support for receiving these RFC 2833 tones and events by including the **receiveRTPAudioTelephonyeEventCapability** or the **receiveRTPAudioToneCapability** in the terminal capability set. When using fast connect procedures, these capabilities can be sent using parallelH245 procedures of section 8.2.4.

…

[End Correction]

6.2 Technical and Editorial Corrections to ITU-T Recommendation H.225.0 (2000)

6.2.1 Registration Request (RRQ) Corrections

Description:	TerminalAlias field in the RRQ message is inaccurately described in the document in case when this field is null. The following text provides the correction.

[Begin Correction]

1) Editorial - Clause 7.9.1, description of terminalAlias)
Change dialedDigits to terminalAlias as below.

terminalAlias – This optional value is a list of alias addresses, by which other terminals may identify this terminal. This field may be used in addition to or as an alternative to the **terminalAliasPattern** and **supportedPrefixes** fields. If the **terminalAlias** is null, a ~~dialedDigits~~ terminalAlias address may be assigned by the gatekeeper, and included in the RCF. If an email-ID is available for the endpoint, it should be registered. Note that multiple alias addresses may refer to the same transport addresses. All of the endpoint's aliases that it desires to register shall be included in this list unless the **additiveRegistration** option is specified in which case the endpoint aliases in an RRQ shall be added to the list of aliases currently registered for the endpoint.

[End Correction]

6.2.2 Section 7.6 H.225.0 Common Message Elements Correction

Description:	Modification to the text in Section 7.6 to define **H248SignalsDescriptor** and **H248PackagesDescriptor** as Octet Strings that represent ASN.1 PER encoded H.248 **SignalsDescriptor** and H.248 **PackagesDescriptor** respectively.

[Begin Correction]

…

The **H248PackagesDescriptor** structure is ~~a PackagesDescriptor as described in Recommendation H.248, in binary format.~~ an octet string, which will contain ASN.1 PER encoded H.248 **PackagesDescriptor**.

The **H248SignalsDescriptor** structure is ~~a SignalsDescriptor as described in Recommendation H.248, in binary format.~~ an octet string, which will contain ASN.1 PER encoded H.248 **SignalsDescriptor**.

…

[End Correction]

6.2.3 Annex H H.225.0 Message Syntax (ASN.1) Corrections

Description:	Changes in H.225 Version 4 ASN.1 syntax. Changes include removing dependencies on H.248 syntax and the addition of an invalid Call Identifier Release Complete reason.

[Begin Correction]

```
IMPORTS
                SIGNED{},
                ENCRYPTED{},
                HASHED{},
                ChallengeString,
                TimeStamp,
                RandomVal,
                Password,
                EncodedPwdCertToken,
                ClearToken,
                CryptoToken,
                AuthenticationMechanism
        FROM H235-SECURITY-MESSAGES
                DataProtocolCapability,
                T38FaxProfile
        FROM MULTIMEDIA-SYSTEM-CONTROL;
                ~~PackagesDescriptor~~
                ~~SignalsDescriptor~~
        ~~FROM MEDIA-GATEWAY-CONTROL;~~

~~H248PackagesDescriptor ::= PackagesDescriptor~~

~~H248SignalsDescriptor ::= SignalsDescriptor~~

H248PackagesDescriptor ::= OCTET STRING -- This octet string contains ASN.1 PER encoded H.248
                                        --PackagesDescriptor.
H248SignalsDescriptor ::= OCTET STRING -- This octet string contains ASN.1 PER encoded H.248
                                       -- SignalsDescriptor.

                                …
ReleaseCompleteReason ::= CHOICE
{
        noBandwidth                     NULL,   -- bandwidth taken away or ARQ denied
        gatekeeperResources             NULL,   -- exhausted
        unreachableDestination          NULL,   -- no transport path to the destination
        destinationRejection            NULL,   -- rejected at destination
        invalidRevision                 NULL,
        noPermission                    NULL,   -- called party's gatekeeper rejects
        unreachableGatekeeper           NULL,   -- terminal cannot reach gatekeeper for ARQ
        gatewayResources                NULL,
        badFormatAddress                NULL,
        adaptiveBusy                    NULL,   -- call is dropping due to LAN crowding
        inConf                          NULL,   -- no address in AlternativeAddress
        undefinedReason                 NULL,
        ...,
        facilityCallDeflection          NULL,   -- call was deflected using a Facility message
        securityDenied                  NULL,   -- incompatible security settings
        calledPartyNotRegistered        NULL,   -- used by gatekeeper when endpoint has
                                                -- preGrantedARQ to bypass ARQ/ACF
        callerNotRegistered             NULL,   -- used by gatekeeper when endpoint has
                                                -- preGrantedARQ to bypass ARQ/ACF
        newConnectionNeeded             NULL,   -- indicates that the Setup was not accepted on this
                                                -- connection, but that the Setup may be accepted on
                                                -- a new connection
        nonStandardReason               NonStandardParameter,
        replaceWithConferenceInvite     ConferenceIdentifier,  -- call dropped due to
subsequent
                                                               -- invitation to a conference
                                                               -- (see H.323 8.4.3.8)
        genericDataReason               NULL,
```

```
        neededFeatureNotSupported     NULL,
        tunnelledSignallingRejected   NULL,
        invalidCID                    NULL
}

                                      ...

CircuitIdentifier ::= ~~CHOICE~~SEQUENCE
{
        cic             CicInfo OPTIONAL,
        group           GroupID OPTIONAL,
        ...
}

                                      ...

LocationRejectReason ::= CHOICE
{
        notRegistered                 NULL,
        invalidPermission             NULL,   -- exclusion by administrator or feature
        requestDenied                 NULL,   -- cannot find location
        undefinedReason               NULL,
        ...,
        securityDenial                NULL,
        aliasesInconsistent           NULL,   -- multiple aliases in request identify distinct
people
        routeCalltoSCN                SEQUENCE OF PartyNumber,
        resourceUnavailable           NULL,
        genericDataReason             NULL,
        neededFeatureNotSupported     NULL,
        hopCountExceeded              NULL,
        incompleteAddress             NULL
}
```

[End Correction]

6.2.4 Clarification for the usage of rasAddress

Description:	There is no requirement that the GK should send back its responses to the GRQ and RRQ messages where they came from. The following clarifies the usage of rasAddress field in these messages.

In Section 7.8.1 GatekeeperRequest (GRQ) and Section 7.9.1 RegistrationRequest (RRQ), add the following line to the description for rasAddress.

[Begin Correction]

rasAddress – This is the transport address that this endpoint uses for registration and status messages. The Gatekeeper shall send RAS messages to this address and not to the address from which the message was sent, unless the **rasAddress** cannot be decoded.

[End Correction]

6.2.5 ReleaseCompleteReason to Cause IE mapping

Description:	The description for Cause IE for the release complete reason of noPermission is incorrect. The following text corrects it. Additionally, a new mapping is added to support the invalidCID reason added via this implementers guide.

[Begin Correction]

Table 5/H.225.0 – ReleaseCompleteReason to cause IE mapping

ReleaseCompleteReason code	Corresponding Q.931/Q.850 cause value
noBandwidth	34 – No circuit/channel available
gatekeeperResources	47 – Resource Unavailable
unreachableDestination	3 – No route to destination
destinationRejection	16 – Normal call clearing
invalidRevision	88 – Incompatible destination
noPermission	~~111~~ 127 – Interworking, unspecified
unreachableGatekeeper	38 – Network out of order
gatewayResources	42 – Switching equipment congestion
badFormatAddress	28 – Invalid number format
adaptiveBusy	41 – Temporary Failure
inConf	17 – User busy
undefinedReason	31 – Normal, unspecified
facilityCallDeflection	16 – Normal call clearing
securityDenied	31 – Normal, unspecified
calledPartyNotRegistered	20 – Subscriber absent
callerNotRegistered	31 – Normal, unspecified
newConnectionNeeded	47 – Resource Unavailable
nonStandardReason	127 – Interworking, unspecified
replaceWithConferenceInvite	31 – Normal, unspecified
genericDataReason	31 – Normal, unspecified
neededFeatureNotSupported	31 – Normal, unspecified
tunnelledSignallingRejected	127 – Interworking, unspecified
invalidCID	3 – No route to destination

[End Correction]

6.2.6 Clarification for sending PNP numbers in Information messages

Description:	The following text clarifies that PNP numbers shall be sent in the Called Party Number IE of the Information message.

[Begin Correction]

Table 9/H.225.0 – Information Message Content

Information element	H.225.0 status (M/F/O)	Length in H.225.0
Protocol discriminator	M	1

Call reference	M	3
Message type	M	1
Sending complete	O	1
Display	O	2-82
Keypad facility	O	2-34
Signal	O	2-3
Called party number	O (Note)	2-35
User-user	M	2-131
Note: The Called Party Number IE will be used to carry numbers from a Private Numbering Plan when performing overlapped sending according to 8.1.12/H.323.		

[End Correction]

6.2.7 Clarification for using Bearer Capability IE in Connect and Progress messages

Description:	The following text removes the requirement that Bearer Capability IE is mandatory in Connect and Progress messages if the connection is between a terminal and a gateway.

[Begin Correction]

Table 8/H.225.0 – Connect

Information element	H.225.0 status (M/F/O)	Length in H.225.0
Protocol discriminator	M	1
Call reference	M	3
Message type	M	1
Bearer capability	O ~~(Note)~~	5-6
Extended facility	O	8-*
Channel identification	FFS	NA
Facility	O	8-*
Progress indicator	O	2-4
Notification indicator	O	2-*
Display	O	2-82
Date/Time	O	8
Connected Number	O	2-*
Connected Sub-Address	O	2-23
Low layer compatibility	FFS	NA
High layer compatibility	FFS	NA
User-user	M	2-131
~~NOTE – Bearer capability is mandatory if the message is between a terminal and a gateway.~~		

[End Correction]

[Begin Correction]

Table 10/H.225.0 – Progress

Information element	H.225.0 status (M/F/O)	Length in H.225.0
Protocol discriminator	M	1
Call reference	M	3
Message type	M	1
Bearer capability	O ~~(Note)~~	5-6
Cause	O	2-32
Extended facility	O	8-*
Channel identification	FFS	NA
Facility	O	8-*
Progress indicator	M	2-4
Notification indicator	O	2-*
Display	O	2-82
High layer compatibility	FFS	NA
User-user	M	2-131
~~NOTE – The Bearer capability information element is mandatory if the message is between a terminal and a gateway.~~		

[End Correction]

6.2.8 Clarification on GK response to additive registration requests

Description:	There is some ambiguity in H.323 version 4 on how to handle the scenario where a gatekeeper only wants to acknowledge a subset of aliases proposed in an additive RRQ. The gatekeeper could return an RCF specifying the accepted aliases in the terminalAliasPattern field. The gateway would then assume that the other aliases were rejected. Alternately, the gatekeeper could return an RRJ specifying the rejected aliases in the invalidTerminalAliases field of the reject reason. In this case the gateway would assume that the other aliases were accepted. The following additions clarify the usage of aliases in RCF and RRJ messages.

[Begin Correction]

7.9.2 RegistrationConfirm (RCF)

terminalAlias – This optional value is a list of alias addresses, by which other terminals may identify this terminal. This field may be used in addition to or as an alternative to the **terminalAliasPattern** and **supportedPrefixes** fields. It specifies the alias addresses that

have been accepted from those proposed in the associated RRQ message. If none were proposed in the RRQ, this list gives aliases assigned by the Gatekeeper. If this field is not included and alias addresses were proposed in the RRQ, then the Gatekeeper has accepted all of the proposed alias addreses. If this field is included and specifies a subset of the alias addresses proposed in the RRQ, then the Gatekeeper has accepted only those addresses.

terminalAliasPattern – This optional value is a list of address patterns specifying aliases and addresses by which other endpoints may identify this endpoint. This field may be used in addition to or as an alternative to the **terminalAlias** and **supportedPrefixes** fields. It specifies the aliases and addresses that have been accepted from those proposed in the associated RRQ message. If none were proposed in the RRQ, this list gives aliases and addresses assigned by the Gatekeeper. If this field is not included and address patterns were proposed in the RRQ, then the Gatekeeper has accepted all of the proposed patterns. If this field is included and specifies a subset of the address patterns proposed in the RRQ, then the Gatekeeper has accepted only those patterns.

supportedPrefixes – This optional value is a list of prefixes by which other endpoints may identify this endpoint. This field may be used in addition to or as an alternative to the **terminalAlias** and **terminalAliasPattern** fields. It specifies the address prefixes that have been accepted from those proposed in the associated RRQ message. If none were proposed in the RRQ, this list gives prefixes assigned by the Gatekeeper. If this field is not included and addres prefixes were proposed in the RRQ, then the Gatekeeper has accepted all of the proposed prefixes. If this field is included and specifies a subset of the address prefixes proposed in the RRQ, then the Gatekeeper has accepted only those prefixes.

7.9.3 RegistrationReject (RRJ)

rejectReason – The reason for the rejection of the registration. This field may contain an invalidTerminalAliases value, in which case it contains a list of aliases, addresses and supported prefixes that were determined to be invalid in the associated RRQ message. In any event, all of the aliases, addresses and supported prefixes from the associated RRQ are rejected along with those specified in the invalidTerminalAliases field. A reason of **genericDataReason** indicates that the request was rejected as a result of a generic element or feature; in this case, additional information may be specified in the **genericData** field.

[End Correction]

6.2.9 Progress Indicator in Setup Message

Description:	In H.225.0v4, the use of progress indicator is forbidden in a SETUP message. However, C.7.1.1 of the Recommendation H.246 Annex C (2000) defines coding rules from the forward call indicators parameter and the access transport parameter of IAM (ISUP) to the progress indicator information element of SETUP (H.225), indicating that the notification of progress indicator is allowed to H.323 terminals. Thus use of progress indicator should be made optional as below. In addition, this table lists “Repeat indicator” twice. One instance of it is removed.

[Begin Correction]

Table 12/H.225.0 – Setup

Information element	H.225.0 status(M/F/O/CM)	Length in H.225.0
Protocol discriminator	M	1
Call reference	M (Note 2)	3
Message type	M	1
Sending complete	O	1
Repeat indicator	F	NA
Bearer capability	M	5-6
Extended facility	O	8-*
Channel identification	FFS	NA
Facility	O	8-*
Progress indicator	~~F~~O	~~NA~~2-4
Network specific facilities	F	NA
Notification indicator	O	2-*
Display	O	2-82
Keypad facility	O	2-34
Signal	O	2-3
Calling party number	O	2-131
Calling party subaddress	CM (Note 1)	NA
Called party number	O	2-131
Called party subaddress	CM (Note 1)	NA
Redirecting Number	O	2-*
Transit network selection	F	NA
~~Repeat indicator~~	~~F~~	~~NA~~
Low layer compatibility	FFS	NA
High layer compatibility	FFS	NA
User-user	M	2-131
NOTE 1 – Subaddresses are needed for some SCN call scenarios; they should not be used for packet-based network side only calls. NOTE 2 – If an ARQ was previously sent, the CRV used here shall be the same.		

[End Correction]

6.2.10 Additions to Q.931 timers usage in H.225.0

Description:	H.225.0 omitts several timers defined in Q.931. The following text adds these timers to H.225.0.

[Begin Correction]

7.5 Q.931 timer values

Two Q.931 timers shall be supported:

…

The following additional Q.931 timers should be supported:

- The "overlap sending timer" T302 (see Tables 9-1/Q.931 and 9-2/Q.931) defining after which time the called endpoint shall stop waiting for the dialed digits from calling endpoint while in the overlap sending. This timer starts when SETUP ACK is sent or INFORMATION received and normally terminates when sending complete indication is received. This timeout value shall be 10-15 seconds.
- The "overlap receiving timer" T304 (see Tables 9-1/Q.931 and 9-2/Q.931) defining after which time the calling endpoint shall stop waiting for the dialed digits from called endpoint user while in the overlap receiving. This timer starts when SETUP ACK is received restarts when INFORMATION is sent and normally terminates CALL PROCEEDING, ALERTING or CONNECT is received. This timeout value shall be at least 20 seconds.
- The "incoming call proceeding timer" T310 (see Tables 9-1/Q.931 and 9-2/Q.931) defining after which time the called endpoint shall stop waiting for the dialed digits from calling endpoint while in the overlap sending. This timer starts when CALL PROCEEDING is received and normally terminates on ALERTING, CONNECT or when the caller terminates the call attempt and sends Release Complete. This timeout value shall be at least 10 seconds.
- The "status timer" T322 (see Tables 9-1/Q.931 and 9-2/Q.931) defining after which time the called endpoint shall stop waiting STATUS message response to STATUS ENQUIRY it has sent. This timer starts when STATUS ENQUIRY is sent normally terminates when STATUS message is received. This timeout value shall be at least 4 seconds.

[End Correction]

6.2.11 ASN.1 specification error for URQ

Description:	There is an error in the ASN.1 definition for the URQ message in the published version of the H.225 (2000) document. The following text corrects this error.

[Begin Correction]

```
UnregistrationRequest ::= SEQUENCE --(URQ)
{
        requestSeqNum                   RequestSeqNum,
        callSignalAddress               SEQUENCE OF TransportAddress,
        endpointAlias                   SEQUENCE OF AliasAddress OPTIONAL,
        nonStandardData                 NonStandardParameter OPTIONAL,
        endpointIdentifier              EndpointIdentifier OPTIONAL,
        ...,
        alternateEndpoints              SEQUENCE OF Endpoint OPTIONAL,
        gatekeeperIdentifier            GatekeeperIdentifier OPTIONAL,
        tokens                          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ~~SUnregRequestReason~~ ICV OPTIONAL,
        reason                          UnregRequestReason OPTIONAL,
        endpointAliasPattern            SEQUENCE OF AddressPattern OPTIONAL,
        supportedPrefixes               SEQUENCE OF SupportedPrefix OPTIONAL,
        alternateGatekeeper             SEQUENCE OF AlternateGK OPTIONAL,
        genericData                     SEQUENCE OF GenericData OPTIONAL
}
```

[End Correction]

6.2.12 Clarifying the semantics of destCallSignalAddress in ACF for answerCall ARQ

Description:	The content of the destCallSignalAddress in ACF message from a Gatekeeper in response to an answerCall ARQ to the GK is not very clear. The following text clarifies its use.

[Begin Correction]

7.11.2 AdmissionConfirm (ACF)

…

destCallSignalAddress – The transport address to which to send Q.931 call signalling, but may be an endpoint or gatekeeper address depending on the call model in use. This field shall be ignored by an endpoint that receives an ACF in response to an ARQ to answer a call.

[End Correction]

6.2.13 Clarifying the semantics of sourceInfo in LRQ message

Description:	There is some misunderstanding over the usage of sourceInfo field in the LRQ – when originated from a gatekeeper – if it referred to the sending gatekeeper or to the endpoint on behalf of which the gatekeeper may have generated the request. The text below clarifies the usage of the sourceInfo field in LRQ messages.

[Begin Correction]

7.13.1 LocationRequest (LRQ)

…

sourceInfo – ~~Indicates~~ Identifies the ~~sender~~sending entity of the LRQ. For a gatekeeper generated LRQ this would identify the gatekeeper and for an endpoint generated LRQ this would identify the endpoint. The gatekeeper can use this information to decide how to respond to the LRQ.

[End Correction]

6.3 Technical and Editorial Corrections to ITU-T Recommendation H.245 (2/2003)

6.3.1 Annex B Section 3.1 Open Logical Channel

Description:	The textual description regarding encryptionSync is inconsistent with section B.3.2 and H.235 Section 8.5. The inconsistency persists since H.245 Version 3. The text below corrects this error.

[Begin Correction]

The encryptionSync field shall be ~~used by the master when acknowledging the opening of a channel by a slave. It is used~~ provided by the master in order to provide the encryption key value and the synchronization point at which the key should be used. For H.323, the syncFlag shall be set to the RTP dynamic payload number which matches the key.

[End Correction]

6.4 Technical and Editorial Corrections to ITU-T Recommendation H.246 (1998)

6.4.1 Annex A Corrections

Description:	The H.245 equivalents defined for H.230 commands MCV and Cancel-MCV were incorrectly defined in H.246. The following text corrects those table entries.

[Begin Correction]

A.5.2.4.1 Multipoint Control C&I

H.230 command/indication	**H.245 equivelent**
MCV	~~Send **broadcastMe**~~ Send either **conferenceRequest.broadcastMyLogicalChannel** or **conferenceCommand.broadcastMyLogicalChannel** with the LCN of the video channel in the direction from the gateway to the H.323 endpoint. If the gateway has previously both sent and received the MVC capability to/from the H.230 side (indicating that both ends of the terminal-MCU or inter-MCU link have declared the MVC capability or the H.245 equivalent), then the H.245 side shall use the **conferenceRequest** form of the message. Otherwise, it shall use the **conferenceCommand** form of the message.
Cancel-MCV	~~Send **cancelBroadcastMe**~~ Send **conferenceCommand.cancelBroadcastMyLogicalChannel**

[End Correction]

Description:	New H.243 codepoints MVC, MVA, and MVR were approved in February

	2000. To support those new codepoints, the following additions shall be added to the table in A.5.2.4.1 as shown below

[Begin Correction]

A.5.2.4.1 Multipoint Control C&I

H.230 command/indication	**H.245 equivelent**
MVC	Send **conferenceCapability.multipointVisualizationCapability**
MVA	Send **conferenceResponse.broadcastMyLogicalChannel.grantedBroadcastMyLogicalChannel**
MVR	Send **conferenceResponse.broadcastMyLogicalChannel.deniedBroadcastMyLogicalChannel**

[End Correction]

Description:	A minor inconsistency has been discovered in section A.5.2.4.4 of H.246 Annex A. The H.245 equivalent continuous presence BAS codes were not included in H.245v3 so continuous presence processing cannot be translated through a H.320-H.323 gateway. To correct this, commands are added to H.245 and the following corrected translations amend H.246.

[Begin Correction]

A.5.2.4.4 Multipoint Control C&I

H.230 command/indication	**H.245 equivelent**
VIN	Send **terminalYouAreSeeing**
VCB/Cancel-VCB	Send **makeTerminalBroadcaster / CancelMakeTerminalBroadcaster**
VCS/Cancel-VCS	Send **sendThisSource / CancelSendThisSource**
VCR	Send **videoCommandReject**
VIN2	~~FFS~~Send **terminalYouAreSeeingInSubPictureNumber**
VIC	~~FFS~~Send **videoIndicateCompose**
VIM	~~FFS~~Send **videoIndicateMixingCapability**

[End Correction]

6.4.2 Reference to ATM Forum Document

Description:	To help clarify the usage of H.246 with respect to ATM, a reference to an

	ATM Forum document has been proposed. This reference shall appear in next H.246 publication from the ITU.

[Begin Correction]

---

1 Scope

...

Voice/Voiceband terminals on GSTN use the appropriate national standards for call control and G.711 or analogue signals for voice. Voice/Voiceband terminals on ISDN use the appropriate national variant of Q.931 for call control and G.711 for voice.

Interworking of H.323 over ATM with H.323 over non-ATM IP networks is possible through the use of an H.323-H.323 gateway. Transport of H.323 media streams over ATM is described in AF-SAA-0124.000.

[End Correction]

[Begin Correction]

---

2 Normative References

...

- ATM Forum Technical Committee, AF-SAA-0124.000, *Gateway for H.323 Media Transport Over ATM*, 1999

[End Correction]

6.5 Technical and Editorial Corrections to ITU-T Recommendation H.235 (2000)

6.5.1 Section 5 – Conventions

Description:	A need for clarification is found regarding the deeper meaning of "the value of the pad should be determined by the normal convention of the cipher algorithm". This issue persists since H.235v1. The following clarification provides the necessary background.

[Begin Correction]

---

5 Conventions

When deploying media encryption in conjunction with payload padding, the text sometimes says "the value of the pad should be determined by the normal convention of the cipher algorithm"; see e.g., sections 8.6.1, B.2.4 and Figure I.5. This is to mean that some cipher algorithms (e.g. DES) provide further implementation advice how the sender may choose the value of the padding byte(s). Examples could be random fill-in values, static values or other generated patterns. Whatever method is deployed does not impact interoperability, yet the security quality may well be different. This is considered as an implementation matter and is not specified any further in this recommendation.

[End Correction]

6.5.2 Section 7.0 - Connection Establishment Procedures

Description:	An error exists in H.235 Version 1 and in H.235 Version 2 regarding the description how to terminate secured connections that have insufficient security capabilities. The text below attempts to correct this error.

Editorial - Clause 7.0

[Begin Correction]

7.1 Introduction

In the cases in which there are no overlapping security capabilities, the called terminal may refuse the connection. The error returned should convey no information about any security mismatch; the calling terminal will have to determine the problem by some other means. In cases where the calling terminal receives a ~~CONNECT ACKNOWLEDGE~~ message without sufficient security capabilities, it should terminate the call.

[End Correction]

6.5.3 Section 8.1 - Security in H.245 Control Channel Operations

Description:	An editorial error has been found in H.235v1 and later giving imprecise description on how to secure the H.245 control channel.

[Begin Correction]

8.1 Secure H.245 channel operation

Assuming that the connection procedures in the previous clause (Connection establishment procedures) indicate a secure mode of operation, the negotiated handshake and authentication shall occur for the H.245 control channel before any other H.245 messages are exchanged. If negotiated, any exchange of certificates shall occur using any mechanism appropriate for the H-Series terminal(s). After completing the securing of the H.245 channel, the terminals use the H.245 protocol in the same manner that they would in an insecure mode.

[End Correction]

6.5.4 Section 8 & Annex D Section 7 - Key Management

Description:	H.235 Versions 1 and 2 lack a clear procedure how to pad the session key before encryption. This might cause interoperability problems. The change defines that the session key is padded with padding bytes before encryption; a length byte is not necessary as the ASN.1 PER decoder is able to deduce the amount to padding to be reduced. Due to document restructuring, this change appears in section 8.6.1 and also in B.3.4 of H.235v3.

[Begin Correction]

- The encrypted session key shall be carried in the H.235Key/**sharedSecret** within the **encryptionSync** field. The session key shall be carried in the **keyMaterial** field of the **KeySyncMaterial**. The **KeySyncMaterial** if not a multiple of the block size - shall be padded to a multiple of blocks before encryption. The value of the pad should be determined by the normal convention of the cipher algorithm. The (padded) **KeySyncMaterial** shall be~~is~~ encrypted using:
  - 56 bits of the shared secret, starting with the least significant bits from the Diffie-Hellman secret for OID "X" or OID "Y"
  - all the bits of the shared secret for OID "Z" starting with the least significant bits from the DH secret.

[End Correction]

Description:	H.235 Annex D.7 does not describe the procedures for secure fast start, parallel H.245 and early media. Lack of such a procedures is a potential source of interoperability problems. The added section provides the clarification; due to text restructuring, the changes appear in section 8.6.1.1.

[Begin Correction]

8.6.1.1 Fast start with parallel H.245 security

The fast start procedure may be used in conjunction with the parallel H.245 procedure that may deploy terminal capability negotiation and/or master/slave determination. A benefit of this combination of procedures is to offload the description complexity of security capabilities from the OpenLogicalChannel by having parallel H.245 define the Terminal Security Capabilities. For this, the OpenLogicalChannel security capabilities should remain empty, but the TerminalCapabilitySet in the parallel H.245 part shall capture the security capabilities.

For this particular case, it is assumed that the callee shall be the apriori master until the master/slave determination may yield a different result. The callee as the apriori master shall select terminal capabilities including the security capabilities and shall generate and distribute the session key. The callee shall be able to modify the caller's opened logical channel proposal when returning the agreed security capabilities.

If master/slave determination takes place, then the result of the master/slave determination shall take precedence over the apriori master assignment. The outcome of the master/slave determination shall not affect the initial fast start response.

Note:

Master/slave determination may be deferred to some later point in time beyond initial transmission of media.

For supporting "early media" the callee shall send its response fast start message including the Diffie-Hellman token(s) at the earliest possible time; for example, in the call signalling response immediately following the fast start SETUP message. This is illustrated in Figure 0.

The caller shall discard any encrypted media data as long as a corresponding session key from the response message is not available.

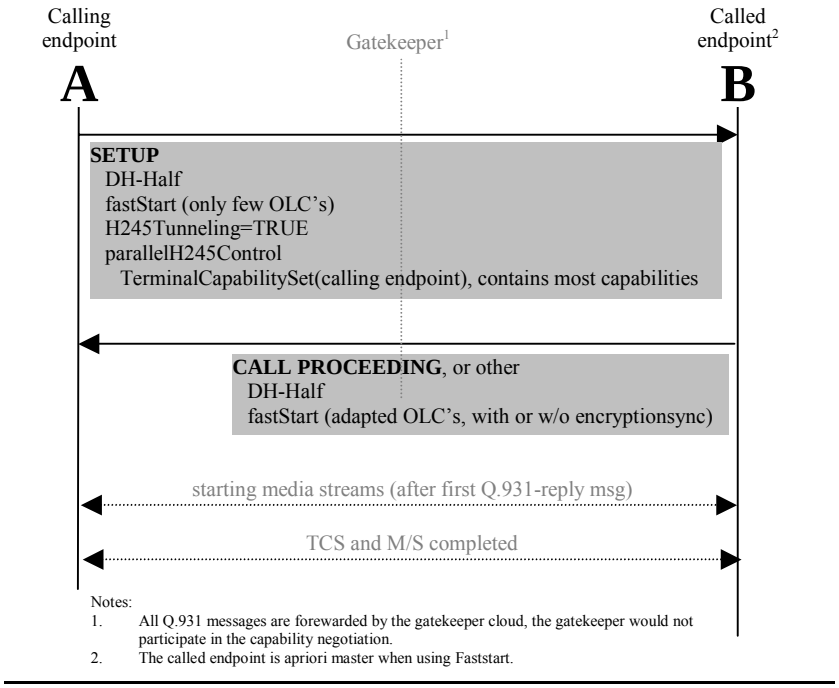


Figure 0/H.235: Fast start with parallel H.245 and security

[End Correction]

6.5.5 Section 8.8 – Diffie-Hellman Operation

Description:	H.235 Versions 1 and 2 in particular do not specify the Diffie-Hellman operation with all necessary variants and cases. It is missing description that defines "parallel DH exchanges" with multiple ClearTokens and missing OIDs for the DH-parameters. The following new section 8.8 provides the clarifying explanations. Text has been added to describe usage of other fields in such tokens to prevent interoperability problems.

8.8 Diffie-Hellman Operation

[Begin Correction]

This recommendation supports the Diffie-Hellman protocol for end-to-end key agreement. Depending on the situation, the negotiated Diffie-Hellman key may act as master key (Annex D.7) or as a dynamic session key (Annex F and H.530).

The Diffie-Hellman system is characterized by the system parameters g and p where p shall be~~is~~ a large prime and g shall be~~is~~ the generator of the multiplicative group modulo p. g^x mod p denotes the (public) Diffie-Hellman half-key of the callee while g^y mod p denotes the (public) Diffie-Hellman half-key of the callee.

H.235 conveys a Diffie-Hellman instance (g, p, g^x) encoded within a **ClearToken** where **dhkey** holds the **halfkey** g^x mod p (resp. g^y mod p) for some secret random x (resp. y), the prime p in **modsize** and the **generator** g. A special case is the triplet (0, 0, 0) or an empty **DHkey** that does not represent any DH-instance but shall be~~is~~ used in signaling that the voice encryption profile is not being used.

Often, the DH-system parameters p and g are fixed for a set of applications with well-defined values, yet end systems may also choose their own set of parameters. The callee should be aware of

the fact that non-standard DH-parameters may provide less security than the parameters look alike at first sight; e.g. the caller might have chosen a non-prime, or *g* generates just a smaller sub-group. While extensive parameter testing is infeasible in practice, it is up to the security policy of the callee whether to accept or reject such offers.

For the fixed DH system parameters, a shorthand characterization through an object identifier may yield more compact encoded messages than including literal values. A **ClearToken** that carries a DH-instance with fixed, standardized DH parameters, may reference the DH instance with a DH-OID in the **tokenOID** field; unless the **tokenOID** is used for other purposes (such as in Annex D.6.3.2 for the first **CryptoToken**). The sender may additionally include the literal DH values but need not do so.

In case several DH-instances are to be indicated each through a DH-OID, the DH-parameters in the first **CryptoToken** (which is being occupied by Annex D) shall be omitted by leaving **DHkey** absent and all DH-instances shall then be carried within separate **ClearTokens** where the **tokenOID** holds the DH-OID and **dhkey** may be left absent; any other fields within that **ClearToken** shall not be used. Note: this does not rule out the possibility to convey a DH instance in the first **CryptoToken** or other available **ClearTokens** by literally including the DH parameter values.

In case, a non-standard DH-instance is to be indicated, the DH-OID "DHdummy" shall be used and the non-standard DH-group parameters shall be explicitly provided in the **ClearToken**.

The callee may submit one or several **ClearTokens** each conveying a different Diffie-Hellman instance. The caller is encouraged to provide as many as possible DH instances as his/her security policy permits. This allows the callee to choose an appropriate instance for the response, thereby increasing the likelihood of finding a successful common parameter set.

The callee shall select and accept a single DH instance (if at all) that it chooses from the unordered set of DH instances provided by the caller in the SETUP message. In case the callee is able to select a DH instance that matches his/her own security needs, the callee shall not modify a proposed DH instance or return one that was not sent by the caller. The strength of the encryption algorithms available to both EPs during the call shall be the same strength as the chosen DH instance provides that is returned by the callee. The callee shall indicate the chosen DH instance in the response message.

In case the callee rejects any of the proposals for security reasons or due to lack of processing capabilities, the callee shall leave **dhkey** absent in the response message.

The callee shall include its DH token in the **Setup**-to-**Connect** response. The callee may include its DH token in the immediate response message following SETUP, or may included the DH token at some later stage, but at latest in the CONNECT message.

Note - There are several aspects to be taken into account as to when the callee may include the DH token(s) during the **Setup**-to-**Connect** responses: the response time, the processing load upon the callee, capability of early media and other aspects. These issues are considered implementation dependant.

For some reasons however, certain routing GKs may not deliver all **Setup**-to-**Connect** responses to the caller. Thus, one or more H.225.0 call signaling response messages including a possible DH token may be dropped and would not arrive at the caller. Then the caller would be unable to compute the DH master key and media session key(s). To prevent such cases, the callee should always include the same DH token in each **Setup**-to-**Connect** response message.

In cases where the DH-OID indicates a different DH-instance than is actually being conveyed within **modsize** and **generator**, the literal values conveyed within **modsize** and **generator** shall take precedence over the DH-OID in the token. For the response, the callee shall replace the conflicting DH-OID with the static DH-OID, e.g., "DH1024," that corresponds to the **modsize** and **generator** or "DHdummy" if there is no corresponding DH-OID.

[End Correction]

6.5.6 Section 11.2 - Media Stream Encryption Procedures

Description:	An editorial error has been found in H.235V2 giving inconsistent description on usage of the OID "S" (DES-MAC). The correct OID is "N".

[Begin Correction]

11.2 Media anti-spamming

…

an encryption algorithm (e.g. DES in MAC mode see ISO/IEC 9797). DES-MAC is indicated using the OID "~~S~~N" while triple-DES-MAC is indicated using OID "O".

[End Correction]

6.5.7 Annex B Section 3 – RTP/RTCP issues

Description:	Some clarifications on the ciphertext stealing mode and some editorial clarifications on the use of shall appear necessary in H.235 Versions 1 and 2. The change provides the clarification. Due to document reorganization, the change appears in section B.3.3 in H.235V3.

B.3 RTP/RTCP issues

[Begin Correction]

ECB and CBC modes always process the input stream a block at a time, and, while CFB and OFB can process the input in any number of octets, N (≤ B), it is recommended that N = B.

Two methods are available to handle packets whose payload is not a multiple of blocks:

1) Ciphertext Stealing for incomplete blocks for ECB and CBC; no~~Zero~~ padding for CFB and OFB.
2) Padding in the manner prescribed by [RTP, section 5.1].

   [RTP, section 5.1] describes a method of padding in which the payload shall be ~~is~~ padded to a multiple of blocks. T~~,~~ ~~t~~he last octet shall be set ~~with~~ to the number of padding octets (including the last), and the P bit set in the RTP header. The value of the pad should be determined by the normal convention of the cipher algorithm.

All H.235 implementations shall support both schemes. The scheme in use can be deduced as follows: if the P bit is set in the RTP header, then the packet is padded; if the packet is not a multiple of B and the P bit is not set, then Ciphertext Stealing applies, else the packet is a multiple of B, and padding does not apply.

[End Correction]

6.5.8 Annex B Section 3 - RTP/RTCP issues

Description:	H.235 Versions 1 and 2 do not absolutely precisely specify how to construct the IV from the sequence number and timestamp. The following change

	provides the clarification. Due to section restructuring, the change appears in section B.3.2.1 in H.235V3.

B.3 RTP/RTCP issues

[Begin Correction]

For the CBC case, an IV shall be constructed from the first B (where B is the block size) octets of: Seq# concatenated with Timestamp. This forms the pattern, SSTTTT, where SS is the 2-octet RTP Seq# and TTTT is the 4-octet RTP timestamp.~~(Seq# + Timestamp).~~ This pattern shall~~ould~~ be repeated until B ~~enough~~ octets have been generated, truncating as necessary. For example, 64- and 128-bit IVs would contain SSTTTTSS and SSTTTTSSTTTTSSTT, respectively.

[End Correction]

6.5.9 Annexes B and D - RTP/RTCP issues and Voice Encryption Security Profile

Description:	The textual description in the first paragraph and the third paragraph is inconsistent on which parts of the RTP packet and the A/V payload to actually perform payload encryption. The inconsistency persists since H.235 Version 1. The text below corrects this error.

B.3 RTP/RTCP issues

[Begin Correction]

The use of encryption on the RTP stream will follow the general methodology recommended in the document referenced in **[RTP]**. The encryption of the media shall occur in an independent, packet by packet basis[1]. The RTP header ~~(including the payload header)~~ shall not be encrypted. For audio codecs, the entire audio codec payload including any audio payload header(s) shall be encrypted. Synchronization of new keys and encrypted text is based upon dynamic payload type.

Care should be taken, when encrypting video streams. Certain video codecs could add specific video payload header which might weaken security when being encrypted. How exactly to encrypt video codec streams, is left for further study.

[End Correction]

D.7 Voice Encryption Security Profile

Description:	The textual description is inconsistent on which parts of the RTP packet and the audio payload to actually perform payload encryption. The inconsistency persists in H.235 Version 2. The text below corrects this error.

[Begin Correction]

The audio payload[2] is encrypted using the negotiated encryption algorithm ("X", "Y" or "Z") operating in CBC mode according to the procedures described in section 11 and annex B of H.235 and the ciphertext padding methods of Appendix I.1/H.235.

---

1 It should be noted that if RTP packet size is larger than MTU size, partial loss (of fragment) will cause the whole RTP packet to be indecipherable.

2 ~~– without the payload header~~

[End Correction]

6.5.10 Annex D Section 7.1 - Key Management

Description:	A spelling error exists in H.235 Version 2 regarding the description how to transport a session key. The text below attempts to correct this error.

Editorial - Clause D.7.1 Key Management

[Begin Correction]

- During FastStart the callee (source of the **Connect**) presents its DH token and the accepted FastStart structures. The session key is included in the **encryptionSync** field. The session key is itself encrypted with the DH shared secret in the same manner as the non-FastStart operation.

[End Correction]

6.5.11 Annex D Section 7.2 - Key Update and Synchronization

Description:	A spelling error exists in H.235 Version 2 regarding the description how to encode and synchronize a session key. The text below attempts to correct this error.

Editorial - Clause D.7.2 Key Update and Synchronization

[Begin Correction]

- **encryptedData**: set to the result of the encrypted **KeySynch~~h~~Material**.

[End Correction]

6.5.12 Annex D Sections 6 and 7 - Usage Illustration for Procedure I, Key update, and synchronization

Description:	Two editorial errors have been detected in the paragraph of H.235 Version 2 potentially leading to confusion and interoperability. The text below corrects

	this error.

D.6.3.4 Usage Illustration for Procedure I

[Begin Correction]

Consider the case in Figure D.1 where three passwords are pair-wise shared between EP1-GK1, between GK1-GK2 and between GK2-EP2. Three 12~~20~~-byte keys - *Key1*, *Key2* and *Key3* – are generated from these passwords based on the procedure described in H.235 Section 10.3.5~~2~~. For maximum security it is recommended to make each of the three random passwords/keys independent.

[End Correction]

D.7.2 Key update and synchronization

Description:	The last sentence in the note is not correct. The error persists in H.235 Version 2 potentially leading to confusion. The text below corrects this error.

[Begin Correction]

NOTE - Since the key update and synchronization relies on H.245 messages that are not piggy-backed during fast connect, this requires H.245 tunneling to be used for secured H.323 entities. ~~Thus, key update and synchronization can only be used in the signature security profile.~~

[End Correction]

6.5.13 Annexes D and E - Specific Conventions, Key management, and Call signaling

Description:	The textual description of H.235 Version 2 section D.2 needs to be clearly aligned with the procedural description in section D.10. The text below provides the clarification.

D.2 Specification Conventions

[Begin Correction]

This profile defines to "set the **generalID** in the **ClearToken** to the identifier of the recipient". This actually means, that for RAS messages destined for the gatekeeper this is the GK identifier; for RAS messages destined for the endpoint this is the ~~or~~ endpoint identifier~~1~~, for H.225.0 call signaling messages destined for the gatekeeper this is the GK identifier and for H.225.0 call signaling messages destined for the endpoint this is the called endpoint identifier, see also section D.10.

The **sendersID** shall be set to the identification string of the sender. This actually means, that for RAS messages destined for the gatekeeper this is the endpoint identifier; for RAS messages destined for the endpoint this is the gatekeeper identifier; for H.225.0 call signaling messages destined for the gatekeeper this is the

GK identifier and for H.225.0 call signaling messages destined for the endpoint this is the called endpoint identifier, see also section D.10.

~~[2] which one depends on the direction EP to GK or vice versa.~~

[End Correction]

D.7.1 Key management

Description:	The textual description of H.235 Version 2 section D.7.1 references incorrectly a non-existing H235Capability while actually H235SecurityCapability of H.245 shall be used. The text below corrects the error.

[Begin Correction]

- During the H.245 Cap exchange, endpoints present **H235<u>Security</u>Capability** entries for the codecs that they support. Each codec is associated with a separate H.235 <u>security</u> capability. These capabilities should indicate support for 56-bit RC2-compatible (OID – "X"), should indicate support for 56-bit DES (OID – "Y") and may indicate support for 168-bit Triple-DES (OID – "Z").

[End Correction]

D.9.2 H.225.0 Call signaling

Description:	Some H.225.0 v4 call signaling messages have been omitted in the table. Those missing messages shall be secured according to procedure I as well. The text in the table below corrects this error.

[Begin Correction]

H.225.0 Call Signaling message	H.235 signaling fields	authentication & integrity
Alerting-UUIE, CallProceeding-UUIE, Connect-UUIE, Setup-UUIE, Facility-UUIE, Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE<u>, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE</u>	CryptoTokens	procedure I

[End Correction]

E.16.2 H.225.0 Call signaling

Description:	Some H.225.0 v4 call signaling messages have been omitted in the table. Those missing messages shall be secured according to procedure II/III as well. The text in the table below corrects this error.

[Begin Correction]

H.225.0 Call Signaling message	H.235 signaling fields	authenti-cation-only	authenti-cation & integrity	non-repudiation
Alerting-UUIE, CallProceeding-UUIE, Connect-UUIE, Setup-UUIE, Facility-UUIE, Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	cryptoTokens	Procedure II/III	procedure II/III	procedure II/III

[End Correction]

6.5.14 Annex D Section 2 – Specification Conventions

Description:	H.235v2 Annex D text and description in several places are not entirely precise on the integrity protection of call signaling messages. Q.931 is not properly addressed. This has led to confusion. The following addition provides the clarification.

Section D.5 – Conventions

[Begin Correction]

This annex may apply message integrity protection that spans the entire message. For H.225.0 RAS the integrity protection covers the entire RAS message; for call signaling this covers the entire H.225.0 call signaling message including the Q.931 headers.

[End Correction]

6.5.15 Annex D Sections 5 and 12 - Normative References and Bibliography

D.5 Normative References

Description:	By the end of year 2001, ITU-T has declared the U.S. National Institute of Standards (NIST) as a recognized standardization organization. This now allows to normatively reference FIPS publications. H.235V2 deploys the Data Encryption Algorithm (DES), which is a NIST pubs. At the time of approval, H.235V2 could not actually reference DES normatively, as NIST was not yet a recognized SDO. Thus, it was decided to move the normative DES references to the bibliography. Still, the text recommends DES as mandatory. The change shown below undoes the informal DES references and makes them normative again.

[Begin Correction]

DES

[FIPS-46-2] US National ~~Bureau~~Institute of Standards, "Data Encryption Standard (DES)", Federal Information Processing Standard, (FIPS) Publication 46-2, December 1993, http://www.itl.nist.gov/div897/pubs/fip46-2.htm

[FIPS-74] US National ~~Bureau~~Institute of Standards, "Guidelines for Implementing and Using the Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 74, April 1981, http://www.itl.nist.gov/div897/pubs/fip74.htm.

[FIPS-81] US National ~~Bureau~~Institute of Standards, "DES Modes of Operation", Federal Information Processing Standard (FIPS) Publication 81, December 1980, http://www.itl.nist.gov/div897/pubs/fip81.htm.

[End Correction]

D.12 Bibliography

Description:	By the end of year 2001, ITU-T has declared the U.S. National Institute of Standards (NIST) as a recognized standardization organization. This now allows to normatively reference FIPS publications. H.235V2 deploys the Data Encryption Algorithm (DES), which is a NIST pubs. At the time of approval, H.235V2 could not actually reference DES normatively, as NIST was not yet a recognized SDO. Thus, it was decided to move the normative DES references to the bibliography. Still, the text recommends DES as mandatory. The change shown below undoes the informal DES references and makes them normative again.

[Begin Correction]

~~DES~~

~~[FIPS-46-2] US National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standard, (FIPS) Publication 46.2, December 1993, http://www.itl.nist.gov/div897/pubs/fip74.htm~~
~~[FIPS-74] US National Bureau of Standards, "Guidelines for Implementing and Using the Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 74, April 1981, http://www.itl.nist.gov/div897/pubs/fip74.htm~~
~~[FIPS-81] US National Bureau of Standards, "DES Modes of Operation", Federal Information Processing Standard (FIPS) Publication 81, December 1980, http://www.itl.nist.gov/div897/pubs/fip81.htm~~

[End Correction]

6.5.16 Annex D Section 6.1 – Overview

Description:	H.235v2 Annex D text and description in several places are not entirely precise on the integrity protection of call signaling messages. Call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

D.6.1 Overview

[Begin Correction]

For authentication, the user shall use a password-based scheme. The password-based scheme is highly recommended for authentication due to its simplicity and ease of implementation. Hashing all the fields in the H.225.0 RAS and call signaling messages is the recommended approach for integrity of the messages (also using the password scheme).

[End Correction]

6.5.17 Annex D Section 6.1.1 – Baseline Security Profile

Description:	H.235v2 Annex D text and description in several places are not entirely precise on the integrity protection of call signaling messages. Call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

D.6.1.1 Baseline security profile

[Begin Correction]

The baseline security profile is applicable in an environment where subscribed passwords/symmetric keys can be assigned to the secured H.323 entities (terminals,..) and network elements (GKs, proxies). It provides authentication and integrity or authentication-only for ~~RAS,~~ H.225.0 RAS and call signaling, and tunneled H.245 using password-based HMAC-SHA1-96 hash as specified by procedure I. H.225.0 call establishment using FastStart (GK-to-GK or terminal-to-terminal) includes integrated key management with Diffie-Hellman.

[End Correction]

6.5.18 Annex D Section 6.3.2 – Symmetric-Key Based Signalling Message Authentication Details (Procedure I)

Description:	H.235v2 Annex D text and description in several places are not entirely precise on the integrity protection of call signaling messages. Call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

D.6.3.2 – Symmetric-Key Based Signalling Message Authentication Details (Procedure I)

[Begin Correction]

- **tokenOID** set to

  "A" indicating that the authentication/ integrity computation includes all fields in the ~~RAS/~~H.225.0 RAS and call signaling messages.

- **hash** containing the authenticator computed using HMAC-SHA1-96. The authenticator can be computed over
  - all the ~~RAS/~~H.225.0 RAS and call signaling fields of the message if **tokenOID** in the **CryptoHashedToken** is set to "A" (indicating authentication and integrity).

tokenOID "A" is used for protection of tunneled H323-UU-PDUs including all H.245 message contents; the hash computation shall be done over the entire **H.225.0** call signaling ~~**PDU**~~ message with all fields according to the procedure described in section D.6.3.3.2.

[End Correction]

6.5.19 Annex D Section 6.3.3.2 – Authentication and Integrity

Description:	H.235v2 Annex D text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

D.6.3.3.2 – Authentication and Integrity

[Begin Correction]

2. ASN.1 encode the entire message; for RAS this shall include the entire H.225.0 RAS message; for call signaling this shall include the entire H.225.0 call signaling message;

[End Correction]

6.5.20 Annex D Section 6.3.4.2 – H.225.0 message authentication and integrity

Description:	H.235v2 Annex D text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

D.6.3.4.2 – H.225.0 message authentication and integrity

[Begin Correction]

The **tokenOID** within the **cryptoHashedToken** is set to "A" indicating that all the fields in the H.225.0 call signaling~~**Setup**~~ message are hashed. The **HASHED** within **token** in **cryptoHashedToken** has **algorithmOID** set to "U" indicating the use of HMAC-SHA1-96 and **params** set to NULL. EP1 then computes the authenticator based on the HMAC-SHA1 algorithm using the 12-byte key *Key1*. The authenticator is computed according to the hash method chosen (A) taking into account the entire H.225.0 call signaling message.

GK1 then computes a new authenticator for this H.225.0 call signaling~~**Setup**~~ message using key *Key2* and algorithm HMAC-SHA1-96 (**algorithmOID**="U"), inserts it in **hash** within **token** and passes the **Setup** message on to GK2.

[End Correction]

6.5.21 Annex D Section 6.3.4.3 – H.245 message authentication and integrity

Description:	H.235v2 Annex D text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

D.6.3.4.3 – H.245 message authentication and integrity

[Begin Correction]

EP1 generates a **CryptoToken** for the H.225.0 message, sets **tokenOID** to "A" indicating authentication and integrity, sets **timeStamp**, **random**, **sendersID**, **generalID** and **tokenOID** to "T" in the **ClearToken** of the **hashedVals**, set **algorithmOID** to "U" indicating the use of HMAC-SHA1-96 and **hash** to the computed hash authenticator over all the fields of the H.225.0 call signaling ~~**H323-UU-PDU**~~ message.

[End Correction]

6.5.22 Annex D Section 6.3.1 - Overview

Description:	H.235 Annex D does not specify the behavior in case the GK receives a secured message that holds an unknown OID. This might be due to a malicious attack or due to the application of unknown OIDs. The following text provides some clues how to cope with such situations.

D.6.3.1 Overview

[Begin Correction]

Gatekeepers detecting failed authentication and/or failed integrity validation in a RAS or Call signaling message received from a secured endpoint or peer gatekeeper respond with a corresponding reject message indicating security failure by setting the reject reason to **securityDenial**. Depending on the ability to recognize an attack and the most appropriate way implemented how to react upon, a gatekeeper receiving a secured **xRQ** with undefined object

identifiers (**tokenOID**, **algorithmOID**) may respond with an unsecured **xRJ** and reject reason set to **securityDenial** or may discard that message. The encountered security event should be logged. On the other hand, the endpoint shall discard the received unsecured message, time out and may retry once again by considering to choose different OIDs. Likewise, a gatekeeper receiving a secured H.225.0 SETUP message with undefined object identifiers (**tokenOID**, **algorithmOID**) may respond with an unsecured RELEASE COMPLETE and reject reason set to **securityDenied** or may discard that message. Similarly, the encountered security event should be logged.

[End Correction]

6.5.23 Annex D Section 6 - Symmetric-Key Based Signalling Message Authentication Details (Procedure I)

Description:	H.235 Annex D does not precisely state, when to include the DH key. This introduces some inconsistencies. The correction removes the uncertainty. Additionally, the note does not precisely define an unambiguous meaning of "representation of 0" and introduces two similar signaling means. The correction gives the method of choice with an unambiguous definition of 0.

D.6.3.2 Symmetric-Key Based Signalling Message Authentication Details (Procedure I)

[Begin Correction]

- **dhkey**, used to pass the Diffie-Hellman parameters as specified in H.235 during **Setup** ~~and~~ to **Connect**.
    - **halfkey** contains the random public key of one party
    - **modsize** contains the DH-prime (see Table D.4)
    - **generator** contains the DH-group (see Table D.4)

    NOTE - When the baseline security profile is used without the voice encryption security profile then no Diffie-Hellman parameters ~~need to~~should be sent and **dhkey** should be absent; ~~instead~~ **halfkey**, **modsize** and **generator** may be set to {'0'B,'0'B,'0'B} ~~may be set to the binary representation of 0 for simplicity~~.

[End Correction]

Description:	The note 5 in H.235 Annex D provides a wrong hint. This introduces some confusion. The correction removes the error.

[Begin Correction]

NOTE 5- The recipient is able to detect usage of procedure I by evaluating the ~~**algorithm**~~**tokenOID** within the hashed **EncodedGeneralToken** (detecting presence of "BU~~"~~).

[End Correction]

6.5.24 Annex D Section 6.3.4.1 - RAS Authentication and Integrity

Description:	H.235 V2 does not correctly illustrate the usage of the user key/password in section D.6.3.4.1 where a key of wrong size is mentioned. The following correction aligns the text with D.6.3.4 and 10.3.5.

D.6.3.4.1 RAS message authentication and integrity

[Begin Correction]

The **tokenOID** within the **cryptoHashedToken** is set to "A" indicating that all the fields in the **ARQ** message are hashed. The **HASHED** within **token** in **cryptoHashedToken** has **algorithmOID** set to "U" indicating the use of HMAC-SHA1-96 and **params** set to NULL. EP1 then computes the authenticator based on the HMAC-SHA1-96 using the 20~~12~~-byte key *Key1*. The authenticator is computed over the entire RAS message.

[End Correction]

6.5.25 Annex D Section 7 - Voice Encryption Security Profile

Description:	The language in H.235 section D.7 is not very precise and leads to confusion.

D.7 Voice Encryption Security Profile

[Begin Correction]

The audio payload ~~is~~ should be encrypted using the negotiated encryption algorithm ("X", "Y" or "Z") ~~operating in CBC mode~~ according to the procedures described in section 11 and annex B of H.235 and the ciphertext padding methods of Appendix I.1/H.235

[End Correction]

6.5.26 Annex D Sections 7.1 and 11 - Key Management

Description:	H.235V1 and higher describe how to transport a media session key as part of **keyMaterial** within H.245. As part of the encrypted **KeySyncMaterial** data structure, **generalID** conveys the identifier of the sender. However, since that **generalID** is encrypted too, the recipient is not able to actually verify the correctness without additional information. In order to let the recipient verify the **generalID** of the sender, a particular end-to-end **ClearToken** shall be used that conveys an unencrypted **sendersID** holding the endpoint identifier of the sender.

D.7.1 Key Management

[Begin Correction]

- During the **Setup**-to-**Connect** sequence a Diffie-Hellman (DH) exchange is performed – this seeds both endpoints with a shared secret. The **ClearToken** field of the **CryptoToken** fields shall contain a **dhkey**, used to pass the parameters as specified in H.235. **halfkey** contains the random public key of one party, **modsize** contains the DH-prime and **generator** contains the

DH-group. The DH parameters to be used are indicated in the table below. For more details, please refer to [RFC2412, appendix E2]. Note that since the H.225.0 messages are authenticated (as described earlier by Procedure I), the DH exchange is an authenticated one.

In either direction with a H.225.0 call signaling message carrying a Diffie-Hellman half-key, the caller or callee shall also include a separate end-to-end **ClearToken** with **sendersID** set to the endpoint identifier of the sender and **tokenOID** set to "E". Any intermediate H.323 signaling entity shall forward that particular end-to-end token unmodified.

[End Correction]

D.11 List of Object Identifiers

Description:	The following OID is defined for the end-to-end **ClearToken** as defined in D.7.1.

[Begin Correction]

"E"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 9} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 9}	End-to-end ClearToken carrying sendersID for verification at the recipient side.

[End Correction]

6.5.27 Annex D Section 7.1 - Key Management

Description:	The language in H.235v2 Annex D is not very precise. The correction provides the clarification. Note: Due to section reorganization, the change appears in H.235V3 in section 8.6.1.

D.7.1 Key management

[Begin Correction]

The initiator of SETUP, the caller presents both its DH token, and the supported FastStart structures. During the **Setup**-to-**Connect** sequence a Diffie-Hellman (DH) exchange shall be ~~is~~ performed – this seeds both endpoints with a shared secret that acts as a master key. The **ClearToken** field of the **CryptoToken** fields shall contain a **dhkey**, used to pass the parameters as specified in H.235. **halfkey** contains the random public key of one party, **modsize** contains the DH-prime and **generator** contains the DH-group. The DH parameters to be used are indicated in the table below. For more details, please refer to [RFC2412, appendix E2]. Note that since the H.225.0 messages are authenticated (as described by Annex D Procedure I), the DH exchange is an authenticated one.

[End Correction]

6.5.28 Annex D Section 7.1 - Key Management

Description:	H.235V1 and higher describe how to transport a media session key as part of **keyMaterial** within H.245. As part of the encrypted **KeySyncMaterial** data structure, **generalID** conveys the identifier of the sender. However, since

	that **generalID** is encrypted too, the recipient is not able to actually verify the correctness without additional information. In order to let the recipient verify the **generalID** of the sender, a particular end-to-end **ClearToken** shall be used that conveys an unencrypted **sendersID** holding the endpoint identifier of the sender. Due to document restructuring, this change appears in section 8.6.1 of H.235v3.

D.7.1 Key Management

[Begin Correction]

- During the **Setup**-to-**Connect** sequence a Diffie-Hellman (DH) exchange is performed – this seeds both endpoints with a shared secret. The **ClearToken** field of the **CryptoToken** fields shall contain a **dhkey**, used to pass the parameters as specified in H.235. **halfkey** contains the random public key of one party, **modsize** contains the DH-prime and **generator** contains the DH-group. The DH parameters to be used are indicated in the table below. For more details, please refer to [RFC2412, appendix E2]. Note that since the H.225.0 messages are authenticated (as described earlier by Procedure I), the DH exchange is an authenticated one.

  In either direction with a H.225.0 call signaling message carrying a Diffie-Hellman half-key, when identification information is available, the caller or callee - when being registered - shall also include a separate end-to-end **ClearToken** with **sendersID** set to the endpoint identifier of the sender and **tokenOID** set to "E". Any intermediate H.323 signaling entity shall forward that particular end-to-end token unmodified.

[End Correction]

6.5.29 Annex D Section 7.2 - Key update and synchronization

Description:	The language in H.235v2 Annex D is not very precise. The correction provides the clarification.

D.7.2 Key update and synchronization

[Begin Correction]

The key refresh rate *shall* be such that no more than 2^{32} blocks are encrypted using the same key. Implementations *should* refresh keys before 2^{30} blocks have been encrypted using the same key (see [H.235, section 11.1]). Both involved entities are free to change the media session key as often as considered necessary due to their security policy. For example, the master may distribute a new session key using **encryptionUpdate** of the **miscellaneousCommand** message. On the other hand the slave may~~can~~ request a new session key from the master to change by using the **encryptionUpdateRequest** of the **miscellaneousCommand** message.

[End Correction]

6.5.30 Annex D Section 7.1 - Key Management

Description:	H.235 Version 2 does not specify the Diffie-Hellman operation with all necessary variants and cases. It is missing description that defines "parallel

	DH exchanges" with multiple ClearTokens and missing OIDs for the DH-parameters. The following change shows how the DH-OIDs relate to the encryption algorithm OIDs.

D.7.1 Key Management

[Begin Correction]

Table D.4 provides the allocated OIDs for the various encryption algorithms and relates them with the allocated OIDs for the Diffie-Hellman group. Two types of DH instances are identified through an OID:

- "DHdummy": An instance of this DH group should be applied whenever exportable (512 bit) security is of concern or any non-standard DH group is being used. Note, that as no particular DH group is defined, the OID references any non-standard DH group. An instance of a 512-bit DH group shall be used to generate a master key for distribution of session key(s) for RC2-compatible ("X") or for DES-56 bit encryption algorithms ("Y").
- "DH1024": This DH group is to be applied when high (1024 bit) security is of concern. The OID references a standardized, fixed DH group. This DH group shall be used to generate a master key for distribution of session key(s) for Triple-DES ("Z") encryption algorithms.

It is recommended to apply the defined 1024-bit DH groups unless other security needs would make other Diffie-Hellman parameters preferential. Further, it is recommended to consider using the defined OIDs identifying the DH groups, see section 8.8. Nevertheless, implementations should be prepared to obtain the DH group parameters literally without explicit OID indication. In this case, implementations should ascertain that the correct DH group is being conveyed according to Table D.4.

Endpoints may use non-standard D-H group parameters. Using OID "DHdummy" should indicate such non-standard D-H groups. It is left to the decision of the callee whether to accept such D-H groups.

Note:

The choice of the D-H group does not eliminate the need to negotiate the actual media encryption algorithm. This shall be accomplished with the H.245 Terminal capability negotiation procedure.

During connection establishment (SETUP-to-CONNECT) usage of the encryption algorithm OIDs shall not be used to indicate a Diffie-Hellman instance.

Encryption Algorithm OID	DH-OID	D-H group description
"X" (RC2-compatible), "Y" (DES)	"DHdummy"	Mod-P, any suitable 512 bit prime
"Z" (triple-DES)	"DH1024"	Mod-P, 1024 bit prime Prime = $2^{1024} - 2^{960} - 1 + 2^{64} * \{ [2^{894} \text{ pi}] + 129093 \}$ = (179769313486231590770839156793787453197860296048756011706444 423684197180216158519368947833795864925541502180565485980503 646440548199239100050792877003355816639229553136239076508735 759914822574862575007425302077447712589550957937778424442426 617334727629299387668709205606050270810842907692932019128194 467627007)$_{10}$ Generator[1] = 2

Table D.4/H.235: Diffie-Hellman groups

[End Correction]

6.5.31 Annex D Section 7.2 – Key update and synchronization

Description:	H.235v2 may deploy voice encryption using an encryption algorithm in CBC mode. That same encryption algorithm is also being used for encrypting the session key. However, text says that for session key encryption in CBC mode, the IV is set to NULL, but lacks clear specification what value the IV shall actually take. This may lead to interoperability problems. The following additional text provides the clarification.

D.7.2 – Key update and synchronization

[Begin Correction]

- **paramS**: set to the initial value. **iv8** holds a random 64-bit block bit pattern that the initiator generates. This field shall~~is~~ not used for the CBC mode and shall be~~is~~ set to NULL, meaning that the CBC-IV for session key encryption shall be set to 0.

[End Correction]

6.5.32 Annex D Section 11 - List of Object Identifiers

Description:	H.235 Version 2 does not specify the Diffie-Hellman operation with all necessary variants and cases. It is missing description that defines "parallel DH exchanges" with multiple ClearTokens and missing OIDs for the DH-parameters. The following change defines the DH-OIDs.

D.11 List of Object Identifiers

[Begin Correction]

Object Identifier Reference	Object Identifier Value(s)	Description
"A"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 1}	Used in procedures I for the CryptoToken-tokenOID indicating that the hash includes all fields in the RAS/ H.225.0 message (authentication and integrity).
"T"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 5} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 5}	Used in procedure I for the ClearToken-tokenOID indicating that the ClearToken is being used for message authentication and integrity.
"U"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 6} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 6}	Used in procedure I for the Algorithm OID indicating use of HMAC-SHA1-96.
"DHdummy"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 40}	Non-standard or 512-bit DH-group explicitly provided
"DH1024"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 43}	1024-bit DH group
"X"	{iso(1) member-body(2) US(840) rsadsi(113549) encryptionAlgorithm(3) 2}	Voice encryption using RC2-compatible (56 bit) or RC2-compatible in CBC mode and 512-bit DH-group
"Y"	{iso(1), identified-organization(3), oiw(14), secsig(3), algorithm(2), desCBC(7)}	Voice encryption using DES (56 bit) in CBC mode and 512-bit DH-group
"Z"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) desEDE(17)}	Voice encryption using Triple-DES (168-bit) in outer-CBC mode and 1024-bit DH-group
"Z1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 3}	Voice encryption using Triple-DES (168-bit) in outer-EOFB mode and 1024-bit DH-group with 64-bit feedback

Table D.6/H.235: Object Identifiers used by Annex D

[End Correction]

6.5.33 Annex D Section 11 – List of Object Identifiers

Description:	H.235v2 Annex D text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

D.11 – List of Object Identifiers

[Begin Correction]

"A"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 1}	Used in procedures I for the CryptoToken-tokenOID indicating that the hash includes all fields in the ~~RAS/~~ H.225.0 RAS and call signaling messages (authentication and integrity).

[End Correction]

6.5.34 Annex E Section 2 – Conventions

Description:	H.235v2 Annex E text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following addition provides the clarification.

Section E.2 – Conventions

[Begin Correction]

This annex may apply message integrity protection that spans the entire message. For H.225.0 RAS the integrity protection covers the entire RAS message; for call signaling this covers the entire H.225.0 call signaling message including the Q.931 headers.

[End Correction]

6.5.35 Annex E Section 4 - Security Services

Description:	H.235 Annex E does not specify the behavior in case the GK receives a secured message that holds an unknown OID. This might be due to a malicious attack or due to the application of unknown OIDs. The following text provides some clues how to cope with such situations.

E.4 Security Services

[Begin Correction]

Gatekeepers detecting failed authentication and/or failed integrity validation in a RAS/call signaling message received from a terminal/peer gatekeeper respond with a corresponding reject message indicating security failure by setting the reject reason to **securityDenial**. Depending on the ability to recognize an attack and the most appropriate way implemented how to react upon, a~~A~~ gatekeeper receiving a secured **xRQ** with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured **xRJ** or may discard that message. The encountered security event should be logged. On the other hand, the endpoint shall discard the received unsecured message, time out and may retry once again by considering to choose different OIDs. Likewise, a gatekeeper receiving a secured H.225.0 SETUP message with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured RELEASE COMPLETE and reason set to **securityDenied** or may discard that message. Similarly, the encountered security event should be logged.

[End Correction]

6.5.36 Annex E Section 5 - Digital Signatures with Public/Private Key Pairs Details (Procedure II)

Description:	H.235 Annex E does not precisely state, when to include the DH key. This introduces some inconsistencies. The correction removes the uncertainty. Additionally, the note is does not precisely define an unambiguous meaning of "representation of 0" and introduces two similar signaling means. The correction gives the method of choice with a unambiguous definition of 0.

E.5 Digital Signatures with Public/Private Key Pairs Details (Procedure II)

[Begin Correction]

- **dhkey**, used to pass the Diffie-Hellman parameters as specified in H.235 during **Setup** ~~and~~ to **Connect**.
    - **halfkey** contains the random public key of one party
    - **modsize** contains the DH-prime (see Table D.4)
    - **generator** contains the DH-group (see Table D.4)

NOTE - When the signature security profile is used without the voice encryption security profile then no Diffie-Hellman parameters ~~need to~~should be sent and **dhkey** should be absent; ~~instead~~ **halfkey**, **modsize** and **generator** may be set to {'0'B,'0'B,'0'B}~~may be set to the binary representation of 0 for simplicity~~.

[End Correction]

6.5.37 Annex E Section 5 – Digital Signatures with Public/Private Key Pairs Details (Procedure II)

Description:	H.235v2 Annex E text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

E.5 – Digital Signatures with Public/Private Key Pairs Details (Procedure II)

[Begin Correction]

- **tokenOID** set to
    - "A" indicating that the authentication/ integrity computation includes all fields in the ~~RAS/~~H.225.0 RAS or call signaling message; see E.9)

- **signature** containing the signature computed using SHA1or MD5 RSA on all the fields (if **tokenOID** is "A", see section E.9) or certain critical fields (if **tokenOID** is "B", see section E.8) of the ~~RAS/~~H.225.0 RAS or call signaling message.

When **tokenOID** "A" is used for protection of tunneled H323-UU-PDUs including all H.245 message contents, then the signature computation shall be done over the entire H.225.0 call signaling~~**H323-UU-PDU**~~ message with all fields according to the procedure described in section E.9. In case, **tokenOID** "B" is used, authentication-only of the **CryptoToken** is achieved when applying the procedure III (see E.8).

[End Correction]

Description:	It was pointed out that one or more H.225.0 call signaling response messages including a possible DH token may be dropped by some GKs and would not arrive at the caller. The problem persists since H.235v1. The following

	clarification provides some more information and a procedure.

[Begin Correction]

E.5 Digital Signatures with Public/Private Key Pairs Details (Procedure II)

- **certificate** contains~~ing~~ the ~~sender's~~ digital certificate of the sender where **type** indicates the certificate type ("V" for MD5-RSA certificates or "W" for SHA1-RSA certificates) and **certificate** carries the actual certificate (see E.12).

[End Correction]

6.5.38 Annex E Section 7 - End-to-End authentication (Procedure III)

Description:	H.235 Annex E does not precisely state, when to include the DH key. This introduces some inconsistencies. The correction removes the uncertainty. Additionally, the note is does not precisely define an unambiguous meaning of "representation of 0" and introduces two similar signaling means. The correction gives the method of choice with an unambiguous definition of 0.

E.7 End-to-End authentication (Procedure III)

[Begin Correction]

- **dhkey**, used to pass the Diffie-Hellman parameters as specified in H.235 during **Setup** ~~and~~ to **Connect**.
    - **halfkey** contains the random public key of one party
    - **modsize** contains the DH-prime (see Table D.4)
    - **generator** contains the DH-group (see Table D.4)

    NOTE - When the signature security profile is used without the voice encryption security profile then no Diffie-Hellman parameters ~~need to~~should be sent and **dhkey** should be absent; ~~instead~~ **halfkey**, **modsize** and **generator** may be set to {'0'B,'0'B,'0'B}~~may be set to the binary representation of 0 for simplicity~~.

[End Correction]

6.5.39 Annex E Section 7 – End-to-End authentication (Procedure III)

Description:	H.235v2 Annex E text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

E.7 – End-to-End authentication (Procedure III)

[Begin Correction]

- **tokenOID** set to

"A" indicating that the hop-by-hop authentication/ integrity computation includes all fields in the ~~RAS/~~H.225.0 RAS or call signaling message (see E.9).

- **signature** containing the signature computed using SHA1-RSA or MD5-RSA on all the fields (if **tokenOID** is "A") or certain critical fields (if **tokenOID** is "B") of the ~~RAS/~~H.225.0 RAS or call signaling message.

[End Correction]

Description:	An editorial error has been identified in H.235 Annex E. The following change removes the error.

[Begin Correction]

E.7 End-to-End authentication (Procedure III)

- **token** ~~with the fields:~~

[End Correction]

6.5.40 Annex E Section 8 - Authentication-only

Description:	H.235 Annex E does not precisely state, when to include the DH key. This introduces some inconsistencies. The correction removes the uncertainty.

E.8 Authentication-only

[Begin Correction]

- **dhkey:** The Diffie-Hellman parameters. This field and sub-fields are ~~only~~ used during **Setup** ~~and~~ to **Connect** messages.

[End Correction]

6.5.41 Annex E Section 9 – Authentication and Integrity

Description:	H.235v2 Annex E text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

E.9 – Authentication and Integrity

[Begin Correction]

2. ASN.1 encode the entire message; for RAS this shall include the entire H.225.0 RAS message; for call signaling this shall include the entire H.225.0 call signaling message;

[End Correction]

6.5.42 Annex E Section 12 - Security Services

Description:	H.235 Annex E does not define how to match the identity within the certificate against other identifying information. Not doing this leads to security weaknesses; yet a recommended procedure is missing. The correction defines a procedure.

E.12 Security Services

[Begin Correction]

Whenever a digital certificate is conveyed in a message, the receiving entity (gatekeeper, endpoint) shall check the identity of the sender (gatekeeper, endpoint) against the identity of the certificate in order to prevent man-in-the-middle attacks.

For digitally signed messages sent from the gatekeeper to the endpoint different possibilities exist for an endpoint to check the gatekeeper identity:

– If the hostname is available for example in the common name attribute of the subject field or of the subjectAltName field in the certificate, the endpoint may check this hostname against the gatekeeper identifier. Additionally, the endpoint may use DNS to query the associated IP address and check it against the gatekeeper's IP address as presented in the gatekeeper's signed response message.

– For example, the gatekeeper identifier may be constructed by the IP address (represented as a 4 byte value in network byte order) concatenated with other identifying information of the gatekeeper identifier, truncated to the maximum length of senders ID field, which carries the gatekeeper’s identity. The endpoint may additionally check the IP address belonging to the hostname against the IP address presented in the IP header of the response of the gatekeeper.

  Note: This method would not work as expected when NAT (Network address translation) devices are in involved.

– If the hostname is not available in the certificate, the IP address - which would be part of the certificate (*iPAddress subjectAltName*) - shall be taken directly to perform the checks stated above.

Users should carefully examine the certificate presented by the gatekeeper to determine if it meets their expectations. If the endpoint has external information as to the expected identity of the gatekeeper, the hostname check may be omitted. For instance, an endpoint may be connecting to a gatekeeper whose address and hostname are dynamic but the endpoint knows the certificate that the gatekeeper will present. In such cases, it is important to narrow the scope of acceptable certificates as much as possible in order to prevent man in the middle attacks. In special cases, it may be appropriate for the endpoint to simply ignore the gatekeeper's identity, but it must be understood that this leaves the connection open to active attacks.

If the hostname does not match the identity in the certificate, user oriented endpoints shall either notify the user (endpoints may give the user the opportunity to continue with the connection in any case) or terminate the connection with a bad certificate error. Automated endpoints should log the error to an appropriate audit log (if available) and should terminate the connection (with a bad certificate error).

Automated endpoints may provide a configuration setting that disables this check, but shall provide a setting, which enables it.

Likewise, it is recommended that the gatekeeper perform an identity check for any digitally signed messages sent from the endpoint to the gatekeeper. How exactly the gatekeeper would implement such a checking is considered as a local matter and should be subject to implementation of the gatekeeper's security policy. As an example, one may imagine that the user name conveyed within the certificate may also be part of the H.323 identifier. Further on, the gatekeeper may crosscheck such identity information against locally administered/configured user data if available and may base a policy decision upon that.

If the gatekeeper has external information as to the expected identity of the endpoint, the hostname check may be omitted. For instance, a gatekeeper may be connecting to an endpoint whose address and hostname are dynamic but the gatekeeper knows the certificate that the endpoint will present. In such cases, it is important to narrow the scope of acceptable certificates as much as possible in order to prevent man in the middle attacks. In special cases, it may be appropriate for the gatekeeper to simply ignore the endpoint identity, but it must be understood that this leaves the connection open to active attack.

If the hostname does not match the identity in the certificate, the gatekeeper should log the error to an appropriate audit log (if available) and should terminate the connection (with a bad certificate error).

If a subjectAltName extension of type dNSName is present, that shall be used as the identity. Otherwise, the (most specific) Common Name field in the Subject field of the certificate shall be used. Although the use of the Common Name is existing practice, it is deprecated and Certification Authorities are encouraged to use the dNSName instead.

Matching shall be performed using the matching rules specified by [RFC3280]. If more than one identity of a given type is present in the certificate (e.g., more than one dNSName name), a match in any one of the set is considered acceptable. Names may contain the wildcard character * which is considered to match any single domain name component or component fragment. E.g., *.a.com matches foo.a.com but not bar.foo.a.com. f*.com matches foo.com but not bar.com.

[End Correction]

6.5.43 Annex E Section 12 – Handling of Certificates

Description:	H.235 Annex E is not clear which OIDs to use in case the digital certificate is being included literally. The following clarification aligns text with Procedure II and with Procedure III.

[Begin Correction]

E.12 Handling of certificates

- The certificate is included in the message exchange as described by Procedures II and III; in this case **certificate** holds the actual certificate and **type** holds OID "V" or OID "W".
- The recipient knows the certificate; possibly stored locally from an earlier exchange.
- Instead of including the certificate itself, the sender provides a URL where the certificate can be found. For this, **certificate** contains the URL and **type** is set to OID "P".

- The recipient obtains the certificate through some other means outside of this recommendation (e.g. LDAP directory lookup).

[End Correction]

6.5.44 Annex E Section 13 – Usage Illustration for Procedure II

Description:	H.235v2 Annex E text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

E.13 – Usage Illustration for Procedure II

[Begin Correction]

Below, we illustrate the procedure details for RAS, H.225.0 call signaling and H.245 message authentication, integrity and non-repudiation.

[End Correction]

6.5.45 Annex E Section 13.3 – H.225.0 message authentication, integrity & non-repudiation

Description:	H.235v2 Annex E text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

E.13.3 – H.225.0 message authentication, integrity & non-repudiation

[Begin Correction]

The procedure for H.225.0 messages is identical to that for RAS messages. The only difference is that the set of fields that need to be signed has to be identified for each H.225.0 call signaling message when the **tokenOID** is set to "B".

[End Correction]

6.5.46 Annex E Section 13.4 – H.225.0 message authentication, integrity & non-repudiation

Description:	H.235v2 Annex E text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

E.13.4 – H.225.0 message authentication, integrity & non-repudiation

[Begin Correction]

In either case (whether a H.225.0 message transmission is pending or an ad-hoc H.225.0 **facility** message is used), the proxy verifies the signature which is meant for it (in this case,

depicted by **tokenOID** of "A") upon receiving the message. Then, if a H.225.0 message transmission is pending for the proxy-GK1 leg, the H.245 message is tunneled within that message; otherwise, it is tunneled within an ad-hoc H.225.0 **facility** message. As in the case of transmission of any H.225.0 call signaling message, a new signature is computed for the H.225.0 call signaling message prior to its transmission from the proxy to GK1. The signature that was sent from EP1 to the proxy and that was not meant for the proxy is passed untouched by the proxy onto GK1.

[End Correction]

6.5.47 Annex E Section 18 – List of Object Identifiers

Description:	H.235v2 Annex E text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

E.18 – List of Object Identifiers

[Begin Correction]

"A"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 1}	Used in procedure II for the CryptoToken-tokenOID indicating that the signature includes all fields in the ~~RAS/~~H.225.0 RAS or call signaling message (authentication and integrity).

[End Correction]

Description:	H.235 Annex E is not clear which OIDs to use in case the digital certificate is being included literally. The following clarification aligns text with Procedure II and with Procedure III.

[Begin Correction]

E.18 List of Object Identifiers

"V"	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 4}	Used in procedure II or in procedure III as algorithm OID indicating use of MD5 RSA digital signature.
"W"	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 5}	Used in procedure II or in procedure III as algorithm OID indicating use of SHA1 RSA digital signature.

Table E.3/H.235: Object Identifiers used by Annex E

[End Correction]

6.5.48 Annex F Section 2 - Normative References

Description:	H.235 Annex F references an obsoleted IETF RFC. RFC 3280 has been issued as an improved version.

F.2 Normative References

[Begin Correction]

RFC 3280~~2459~~; Internet X.509 Public Key Infrastructure Certificate and Revocation List (CRL) Profile, R. Housley et al, *Internet Engineering Task Force*, April 2002~~1999~~.

[End Correction]

6.5.49 Annex F Section 4 Specification conventions

Description:	H.235 Annex F text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following addition provides the clarification.

Section F.4 Specification conventions

[Begin Correction]

This annex may apply message integrity protection that spans the entire message. For H.225.0 RAS the integrity protection covers the entire RAS message; for call signaling this covers the entire H.225.0 call signaling message including the Q.931 headers.

[End Correction]

6.5.50 Annex F Section 6 – Authentication and Integrity

Description:	H.235 Annex F text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

F.6 – Authentication and Integrity

[Begin Correction]

Gatekeepers detecting failed authentication and/or failed integrity validation in a RAS/call signaling message received from a terminal/peer gatekeeper will respond with a corresponding reject message indicating security failure. This is done by setting the reject reason to **securityDenial** or other appropriate security error code according to H.235 Annex B clause B.2.2. Depending on the ability to recognize an attack and the most appropriate way implemented how to react upon, aA gatekeeper receiving a secured **xRQ** with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured **xRJ** and reject reason set to **securityDenial** or may discard that message. The endpoint shall discard the received unsecured message, time out and may retry once again by considering to choose different OIDs. Likewise, a gatekeeper receiving a secured H.225.0 call signaling SETUP message with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured RELEASE COMPLETE and reject reason set to **securityDenied** or may discard that message whereas a gatekeeper receiving a secured H.225.0 FACILITY with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured FACILITY and reason set to **undefinedReason** or may discard that message.

Similarly, the encountered security event should be logged. As part of the returned response, the sender may provide a list of acceptable certificates in separate tokens, in order to facilitate selection of an appropriate one by the recipient.

[End Correction]

6.5.51 Annex F Section 6 - Authentication and Integrity

Description:	Replaces RFC 2459 reference with updated version RFC 3280.

F.6 Authentication and Integrity

[Begin Correction]

NOTE - When digital signatures are applied, a non-repudiation security service may be supported. This also depends on the settings of the key usage bits of the signing key in the certificate (see also RFC 3280~~2459~~).

[End Correction]

Description:	H.235 Annex F does not specify the behavior in case the GK receives a secured message that holds an unknown OID. This might be due to a malicious attack or due to the application of unknown OIDs. The following text provides some clues how to cope with such situations.

[Begin Correction]

Gatekeepers detecting failed authentication and/or failed integrity validation in a RAS/call signaling message received from a terminal/peer gatekeeper will respond with a corresponding reject message indicating security failure. This is done by setting the reject reason to **securityDenial** or other appropriate security error code according to H.235 Annex B clause B.2.2. Depending on the ability to recognize an attack and the most appropriate way implemented how to react upon, a~~A~~ gatekeeper receiving a secured **xRQ** with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured **xRJ** and reject reason set to **securityDenial** or may discard that message. The endpoint shall discard the received unsecured message, time out and may retry once again by considering to choose different OIDs. Likewise, a gatekeeper receiving a secured H.225.0 SETUP message with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured RELEASE COMPLETE and reject reason set to **securityDenied** or may discard that message whereas a gatekeeper receiving a secured H.225.0 FACILITY with undefined object identifiers (**tokenOID**, **algorithmOID**) should respond with an unsecured FACILITY and reason set to **undefinedReason** or may discard that message. Similarly, the encountered security event should be logged. As part of the returned response, the sender may provide a list of acceptable certificates in separate tokens, in order to facilitate selection of an appropriate one by the recipient.

[End Correction]

6.5.52 Annex F Section 7 - Procedure IV

Description:	H.235 Annex F does not describe GK behavior in case of multiple received digitally signed RRQs each carrying a DH token. The clarification defines the recommended behavior for the GK and the endpoint.

F.7 Procedure IV

[Begin Correction]

It is anticipated that a gatekeeper should receive only a single **RRQ** including a DH-token with a digital signature from a particular fixed endpoint. However, lost or delayed **RCF**/**RRJ** messages may lead to retransmission using another signed **RRQ**.

In case the corresponding registration response does not arrive timely at the endpoint, the endpoint may attempt another try. For this, the endpoint shall use the most recent DH token but use a new sequence number and a new timestamp.

For a particular fixed endpoint, the gatekeeper shall use the most recently received signed **RRQ** message and derive the shared secret from that DH-token, regardless of if the GK has already a shared secret available. Thus, the GK shall overwrite any existing shared secret with the newly derived secret. The GK shall respond with a signed **RCF** that holds the response DH-token. Preferably, the response DH token should be generated anew.

Notes:

The recommended and preferred method for key update is by using the FACILITY message as defined in section F.9. However, it is recognized, that key update may be achieved using another additive signed **RRQ** with a new DH-token.

A gatekeeper in possession of a shared secret shall respond to an HMAC-protected **RRQ** (according to Annex D) with an HMAC-protected response message.

[End Correction]

Description:	H.235 Annex F does not define how to match the identity within the certificate against other identifying information. Not doing this leads to security weaknesses; yet a recommended procedure is missing. The correction defines a procedure.

[Begin Correction]

Whenever a digital certificate is conveyed in a message, the receiving entity shall check the identity of the sender against the identity of the certificate according to procedure E.12 in order to prevent man-in-the-middle attacks.

[End Correction]

Description:	H.235 Annex E and F are not entirely clear about which OIDs to deploy to indicate either included digital certificate or URL towards a certificate. The following clarification aligns Annex E and F.

[Begin Correction]

F.7 Procedure IV

- **algorithmOID** in **tokenOID** shall be set to "W" indicating use of RSA-SHA1 signature.
- **signature** shall contain an ASN.1 encoded RSA signature (see section E.10 of H.235).
- **certificate** should contain the sender's user certificate if not available otherwise to the receiver; **type** shall hold OID "W" indicating an included RSA-SHA1 certificate or OID "P" (see H.235 Annex E.18) indicating that **certificate** holds an URL.

[End Correction]

6.5.53 Annex F Section 13 – List of Object Identifiers

Description:	H.235 Annex F text and description in several places are not entirely precise on the integrity protection of call signaling messages. H.225.0 call signaling is not properly addressed. This has led to confusion. The following modification provides the clarification.

F.13 – List of Object Identifiers

[Begin Correction]

"A1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 20}	Used as replacement for OID "A" in procedure II of Annex E for the CryptoToken-tokenOID indicating that the RSA signature/hash includes all fields in the ~~RAS/~~H.225.0 RAS or call signaling message (authentication and integrity).

[End Correction]

6.5.54 Appendix I Section 1 - Ciphertext padding methods

Description:	It has been detected that a descriptive figure is missing in H.235 that describes zero padding in CBC mode. The omission persists since H.235 Version 1.

Appendix I.1 Ciphertext padding methods

[Begin Correction]

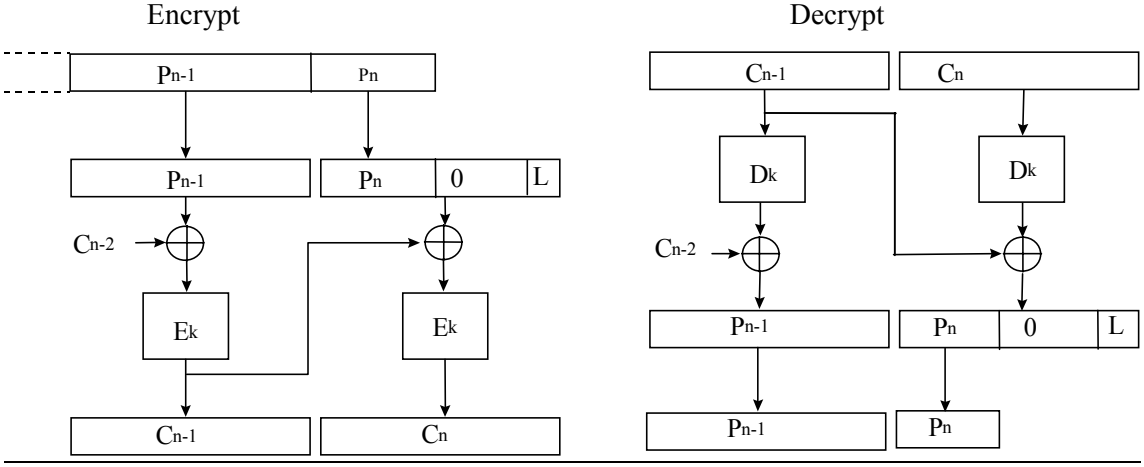


Figure I.2a/H.235 – Zero padding in CBC mode

[End Correction]

Description:	It is observed that ciphertext stealing in ECB or CBC modes requires at least one complete crypto block. This applies to H.235v1 and higher versions. The following clarification is added underneath of Figure I.1/H.235.

[Begin Correction]

I.1 Ciphertext padding methods

Note - Ciphertext stealing in ECB or CBC modes requires the payload to convey at least one complete block. Implementations deploying ciphertext stealing in ECB or CBC modes should ascertain that the payload conveys always at least one crypto block; e.g., by proper choice of the sampling/packetization rate or selection of the encryption algorithm.

[End Correction]

Description:	H.235 does not describe how to operate ciphertext stealing in CBC mode in case the payload is less than one complete block. This problem persists since H.235v1. The following clarification is added underneath of figure I.2/H.235.

[Begin Correction]

I.1 Ciphertext padding methods

In case the payload spans less than one single block, the initial value (IV) shall be used as the previous ciphertext block when ciphertext stealing mode is applied in CBC mode.

[End Correction]

6.5.55 Appendix I Section 4.6 - Back-end Service Support

Description:	A clash of overlapping OIDs values has been detected. OIDs “K”, “L” and “M” in the Appendix I currently have the same value assigned as OIDs “A”, “B” and “R” of Annex E. However, each of the mentioned OIDs shall have a unique value in order to unambiguously identify its purpose. Implementations deploying OIDs from Annex E and Appendix I would thus run into interoperability problems. It is proposed to re-allocate the OIDs in Appendix I with new and distinct values.

I.4.6 Back-end Service Support

[Begin Correction]

Object Identifier Reference	Object Identifier Value	Description
"K"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 31}	indicates a RADIUS challenge in the ClearToken
"L"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 32}	indicates a RADIUS response (conveyed in the challenge field) in the ClearToken
"M"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 33}	indicates BES default mode with a protected password in the ClearToken

Table I.1/H.235: Object Identifiers used by Appendix I.4.6

[End Correction]

6.6 Technical and Editorial Corrections to ITU-T Recommendation H.450 Series

6.6.1 Technical and Editorial Corrections to ITU-T Recommendation H.450.1 (1998)

6.6.1.1 Actions at a Destination Entity

Description:	Typographical errors have been discovered in section 6.6 of H.450.1 (1998). The text below outlines the necessary changes.

[Begin Correction]

1) Section 6.6, line 6

   Change:

   "**rejectUnrecognizedInvokePdu**"

   to

   "**rejectAnyUnrecognizedInvokePdu**"

2) Section 6.6, line 12

   Change:

   "**discardAnyUnrecognizedInvokePDU**"

   to

   "**discardAnyUnrecognizedInvokePdu**"

[End Correction]

6.6.1.2 Corrections to the ASN.1

Description:	H.225.0 (1999) introduces redundancy with H.450.1 in that both H.225.0 (1999) and H.450.1 have screening and presentation information. To remove the redundancy, it was decided that H.225.0 was the proper place for this information and the redundant elements shall be removed from H.450.1. Below shows the revision to the ASN.1 found in Table 6/H.450.1.

[Begin Correction]

```
Addressing-Data-Elements
            { itu-t recommendation h 450 1 version1(0) addressing-data-elements(9)}
      DEFINITIONS AUTOMATIC TAGS ::=
      BEGIN
      IMPORTS     AliasAddress, PartyNumber, PresentationIndicator, Screening Indicator FROM
H323-MESSAGES; -- see H.225.0

                                          ...
      -- PartyNumber defined in Recommendation H.225.0
      -- PublicPartyNumber defined in Recommendation H.225.0
      -- PrivatePartyNumber defined in Recommendation H.225.0
      -- NumberDigits defined in Recommendation H.225.0
      -- PublicTypeOfNumber defined in Recommendation H.225.0
      -- PrivateTypeOfNumber defined in Recommendation H.225.0
      -- PresentationIndicator defined in Recommendation H.225.0 (v3 and beyond)
      -- ScreeningIndicator defined in Recommendation H.225.0 (v3 and beyond)

      EndpointAddress              ::=   SEQUENCE{
            destinationAddress                SEQUENCE OF AliasAddress,
                  -- multiple alias addresses may be used to address the same H.323 endpoint
            remoteExtensionAddress            AliasAddress OPTIONAL,
            …,
            destinationAddressPresentationIndicator   PresentationIndicator OPTIONAL,
                  -- Note 1, 2
            destinationAddressScreeningIndicator            ScreeningIndicator OPTIONAL,
            remoteExtensionAddressPresentationIndicator     PresentationIndicator OPTIONAL,
                  -- Note 1, 2
            remoteExtensionAddressScreeningIndicator  ScreeningIndicator OPTIONAL
                                                }
                  -- Note 1: If this element is not available, presentation allowed shall be
assumed.
                  -- Note 2: If an H.450 APDU that carries this element EndpointAddress also
                  -- contains an element PresentationAllowedIndicator, then the setting of the
                  -- element PresentationAllowedIndicator shall take precedence in case of
                  -- conflicting presentation information.

                                          ...

   ScreeningIndicator                  ::=      ENUMERATED {

               userProvidedNotScreened (0),
                           number was provided by a remote user
                           , and has not been screened by a gatekeeper

               userProvidedVerifiedAndPassed (1),
                           number was provided by a user
                           equipment (or by a remote network), and has
                           been screened by a gatekeeper

               userProvidedVerifiedAndFailed (2),
                           not used, value reserved.

               networkProvided (3),
                           number was provided by a gatekeeper

               …                   }
```

[End Correction]

6.6.1.3 Clarifications to ROS APDUs

Description:	The ASN.1 specification of ROS APDUs has caused some uncertainty over the correct encoding of invoke identifiers. A correct encoding is essential for interoperability between different implementations. The text below attempts to clarify the uncertainty.

Add the following note below Table 4/H.450.1:

[Begin Correction]

Note:

In the *Invoke* APDU, the *invokeID* is an INTEGER constrained by a PER-visible constraint (InvokeIdSet = 0..65535) and is therefore encoded as a **constrained** INTEGER (16 bits, no length field). In the *ReturnResult* and *ReturnError* APDUs, however, the *invokeID* is

encoded as an **unconstrained** INTEGER (with explicit length field) because the applicable constraint ("must be that for an outstanding operation…") is not PER-visible. In the *Reject* APDU the *invokeID* is also encoded as an **unconstrained** INTEGER (with explicit length field) since no constraint applies.

[End Correction]

6.6.2 Technical and Editorial Corrections to ITU-T Recommendation H.450.2 (1998)

6.6.2.1 Editorial Corrections

Description:	Typographical errors have been discovered in sections 11.4.2, 11.5.2, 11.6.2, and 13.4 of H.450.2. The text below outlines the necessary changes.

[Begin Correction]

1) Editorial - Clause 11.4.2, line 4 c)

   Change:

   "The CTSetup.request primitive is used to request call establishment from TRTSE."

   to

   "The CTSetup.request primitive is used to request call establishment to TRTSE"

2) Editorial - Clause 11.4.2, line 5 d)

   Change:

   "The CTSetup.confirm primitive is used to indicate success of call establishment to TRTSE."

   to

   "The CTSetup.confirm primitive is used to indicate success of call establishment from TRTSE."

3) Editorial - Clause 11.5.2, line 6 e)

   Change:

   "The CTIdentify.indication primitive is used to request a call identification."

   to

   "The CTIdentify.indication primitive is used to indicate a call identification."

4) Editorial - Clause 11.5.2, line 11,12 j)

   Change:

   "The CTComplete.request primitive may be used by GKs to request sending of call transfer information to the transferred-to user."

   to

   "The CTComplete.request primitive may be used by GKs to request sending of call transfer information to the transferred-to endpoint."

5) Editorial - Clause 11.5.2, line 13,14 k)

   Change:

   "The CTComplete.indication primitive is used to indicate call transfer information to the transferred-to endpoint."

to

"The CTComplete.indication primitive is used to indicate call transfer information to the transferred-to user."

6) Editorial - Clause 11.6.2, line 2

Change:

"CT-T1 - Timer CT-T1 shall operate at the TRGSE during state CT-Await-Identify-Response. Its purpose is to protect against the absence of response to the CTIdentify.request."

to

"CT-T1 - Timer CT-T1 shall operate at the TRGSE during state CT-Await-Identify-Response. Its purpose is to protect against the absence of response to the CTIdentify.invoke."

7) Editorial – Clause 13.4, FIGURE 25 (sheet 2 of 3, 4th branch) of H.450.2

(i.e. FIGURE 22/H.450.2 (sheet 2 of 3, 4th branch) of H.450.2 (2/98) publication)

Change:

"T4 Timeout"

to

"CT-T4 Timeout"

In addition, the type of symbol was mistake. Time-Out event is an internal event.

[End Correction]

6.6.2.2 Clarification of CallIdentifier and ConferenceIdentifier

Description:	A clarification of the setting of H.225.0 elements **CallIdentifier** and **ConferenceIdentifier** values in conjunction with H.450.2 transferred calls has been added within a new clause 10.7 "Interactions with H.225.0 parameters". ***Special Note: This section appeared in the May 1999 Implementers Guide, but stated that the CallIdentifier should be the same for transferred calls. That definition contradicted H.323v2's definition of the CallIdentifier, so this section has been changed to align with H.323v2 and higher.***

[Begin Correction]

10.7 Interactions with H.225.0 parameters

The H.225.0 CallIdentifier value of the transferred call shall use a new value, rather than the value that was used in the primary call.

The H.225.0 ConferenceIdentifier of a transferred call may use a new value. However, the ConferenceIdentifier of an existing conference (multipoint conference) shall not be altered.

[End Correction]

6.6.2.3 Transfer without Consultation

Description:	An exceptional procedure for a transferred endpoint B actions has been added in clause 8.2.1 to allow call transfer without consultation to take place successfully even if the transferred-to endpoint C does either not support H.450.2 or not support H.450 at all. Furthermore, clause 6 was enhanced to allow a different Interpretation APDU setting.

[Begin Correction]

---

6 Messages and Information elements

...

When conveying the invoke APDU of operation callTransferSetup, the Interpretation APDU shall contain value clearCallIfAnyInvokePduNotRecognized in case of Transfer with Consultation. In case of Call Transfer without Consultation, the Interpretation APDU shall be set to value discardAnyUnrecognizedInvokePdu.

[End Correction]

[Begin Correction]

8.2.1 Transfer without Consultation with transferred-to endpoint C not supporting H.450.2

a) When receiving a CONNECT message from endpoint C (that does not include a response to the callTransferSetup Invoke APDU) while being in state CT-Await-Setup-Response, the transferred endpoint B should continue as if a callTransferSetup Return Result APDU would have been received. This allows endpoint B to successfully continue with the Call Transfer procedures (including appropriate internal call transfer state handling and clearing of the primary call to the transferring endpoint A). This exceptional procedure enables successful Call Transfer even if the transferred-to endpoint C does not support H.450 at all.

b) When a RELEASE COMPLETE message as a response to a SETUP message containing callTransferSetup Invoke APDU is received in endpoint B on the transferred call attempt, possibly containing callTransferSetup Return Error or Reject APDU, then endpoint B may retry call establishment to endpoint C using a normal basic call. Upon receiving the CONNECT message from endpoint C, endpoint B may continue with the procedures as described in a) above.

Note that this procedure may apply if endpoint C supports H.450.1 but no H.450.2 and if endpoint B has not selected the recommended Interpretation APDU value discardAnyUnrecognizedInvokePdu but has set the value to clearCallIfAnyInvokePduNotRecognized.

[End Correction]

6.6.3 Technical and Editorial Corrections to ITU-T Recommendation H.450.3 (1998)

6.6.3.1 Editorial Correction in H.450.3

Description:	Typographical errors have been discovered in H.450.3 clause 12 SDLs.

[Begin Correction]

Editorial – Clause 12 SDL FIGURES 21 (most right branch), 22 (most right branch), 23 (most right branch), 28 (sheet 1 of 4, second right branch) of H.450.3

(i.e. FIGURES 19,20,21 and 24 (sheet 1 of 4) of H.450.3 of H.450.3 (2/98) published).

The type of symbol was mistake. Time-Out event is an internal event.

Note: The text within the referred symbols remains unchanged.

change  to 

[End Correction]

6.6.3.2 Clarification of the CallIdentifier and ConferenceIdentifier

Description:	A clarification of the setting of H.225.0 elements **CallIdentifier** and **ConferenceIdentifier** values in conjunction with H.450.3 forwarded calls has been added within a new clause 9.9.3 "Interactions with H.225.0 parameters". ***Special Note: This section appeared in the May 1999 Implementers Guide, but stated that the CallIdentifier should be the same for diverted calls. That definition contradicted H.323v2's definition of the CallIdentifier, so this section has been changed to align with H.323v2 and higher.***

[Begin Correction]

9.9.3 Interactions with H.225.0 parameters

The H.225.0 **CallIdentifier** of a forwarded call shall use a new value, rather than the value that was used in the forwarding call.

The H.225.0 **ConferenceIdentifier** of a forwarded call may use a new value. However, the **ConferenceIdentifier** of an existing conference (multipoint conference) shall not be altered.

[End Correction]

6.6.3.3 Correction to the ASN.1

Description:	A typographical error has been discovered in the ASN.1 definitions presented in H.450.3, Chapter 11.

[Begin Correction]

H225InformationElement FROM H225-~~Generic~~generic-parameters-definition

[End Correction]

6.6.4 Technical and Editorial Corrections to ITU-T Recommendation H.450.4 (1999)

6.6.4.1 Change Relating to Interpretation APDU

Description:	In order to align H.450.4 with other H.450-series A modified description of the Call Hold Interpretation APDU (i-apdu) setting has been added in clause 6 of Recommendation H.450.4. This information will be contained in the revision 2 of H.450.4 Recommendation to be published by the ITU-T. The modified text is shown below.

[Begin Correction]

---

6 Messages and Information elements

...

When conveying the Invoke APDU of operations **remoteHold** and **remoteRetrieve**, the Interpretation APDU shall be omitted or shall contain the value **rejectAnyUnrecognizedInvokePdu**.

[End Correction]

6.6.4.2 Feature Interaction between H.450.4 and H.450.2

Description:	A modified description of the Call Hold interaction with Call Transfer has been added in clause 9.2.1 of Recommendation H.450.4. This information will be contained in the revision 2 of H.450.4 Recommendation to be published by the ITU-T. The modified text is shown below.

[Begin Correction]

9.2.1 Call Transfer (H.450.2)

If prior to Consultation, the first call has been put on hold, the served User endpoint shall decide whether or not to automatically retrieve the held User before Call Transfer is invoked.

- If the served User endpoint decides for the automatic retrieve option, a~~A~~ **retrieveNotific** Invoke APDU (in case of near end call hold) or a **remoteRetrieve** Invoke APDU (in case of remote-end call hold) may either be sent by the served user prior to the message containing the **callTransferInitiate** Invoke APDU or may be sent within the same message containing the **callTransferInitiate** Invoke APDU.

  If call transfer fails after retrieval from hold was successful (i.e. if callTransferInitiate Return Error or Reject APDU is received or if timer CT-T3 expires), the served user endpoint may automatically re-invoke SS-Hold.

  If remote-end call hold retrieval is unsuccessful, in order to proceed with call transfer the remoteRetrieve Return Error or remoteRetrieve Reject APDU should be disregarded.

- If the served User endpoint decides to not choose the automatic retrieve option, call hold applies to the primary call until call transfer has been completed successfully (i.e. until the primary call is cleared). If transfer fails, the primary call remains being held by User A.

[End Correction]

6.6.5 Technical and Editorial Corrections to ITU-T Recommendation H.450.5 (1999)

6.6.5.1 Clarification of the CallIdentifier

Description:	A clarification of the setting of H.225.0 element CallIdentifier in conjunction with H.450.5 parked calls has been added within clause 8.3 "Interactions with H.225.0 parameters". This information will be contained in the revision 2 of H.450.5 Recommendation to be published by the ITU-T. The modified text is shown below.

[Begin Correction]

8.3 Interaction with H.225.0 parameters

The H.225.0 **CallIdentifier** value within a parked call shall use a new value, rather~~be set to~~ the CallIdentifier value that was used in the primary call. For all other SETUP messages carrying SS-PARK or SS-PICKUP related APDUs as defined within this recommendation, new CallIdentifier values shall be used. Note that the CallIdentifier value of the parked/alerting call is preserved during the SS-PARK / SS-PICKUP procedure within the H.450 APDUs.

[End Correction]

6.6.6 Technical and Editorial Corrections to ITU-T Recommendation H.450.6 (1999)

There are no corrections for H.450.6.

6.6.7 Technical and Editorial Corrections to ITU-T Recommendation H.450.7 (1999)

6.6.7.1 Change Relating to Interpretation APDU

Description:	In order to align H.450.7 with other H.450-series, a modified description of the Message Waiting Indication Interpretation APDU (i-apdu) setting has been added in clause 7.1.1 of Recommendation H.450.7. This information will be contained in the revision 2 of H.450.7 Recommendation to be published by the ITU-T. The modified text is shown below.

[Begin Correction]

7.1.1 H.450.1 Supplementary Service APDU

...

When conveying the Invoke APDU of operations **mwiActivate**, **mwiDeactivate**, and **mwiInterrogate**, the interpretation APDU shall be omitted or shall contain the value **rejectAnyUnrecognizedInvokePdu**. ~~This is implicitly equivalent to specifying an interpretation APDU of rejectAnyUnrecognizedInvokePDU.~~

[End Correction]

6.6.8 Technical and Editorial Corrections to ITU-T Recommendation H.450.8 (2000)

6.6.8.1 Usage of CalledName and AlertingName

Description:	An editorial error has been found in the H.450.8 (2000) Recommendation in the usage of calledName and alteringName. The following text corrects the errors.

[Begin Correction]

7.2 Terminals or MCU as Originating Endpoint

…

A terminal or MCU in receipt of an H.225.0 Connect, Alerting, or Release Complete message containing a connectedName, ~~called~~alertingName, or busyName APDU should not present name information if the Name element indicates namePresentationRestricted.

8.2 Terminals or MCU as Terminating Endpoint

A terminal or MCU in receipt of the H.225.0 Setup message may include name information in the Connect, Alerting or Release Complete as described above in 6.2, 6.3 or 6.4. If presentation of the name to the calling party is desirable, the Name element in the alertingName, connectedName, or busyName operation should indicate namePresentationAllowed. If presentation of the name to the called party is to be restricted, the Name element in the ~~called~~alertingName, connectedName, or busyName operation should indicate namePresentationRestricted.

[End Correction]

6.6.9 Technical and Editorial Corrections to ITU-T Recommendation H.450.9 (2000)

There are no corrections for H.450.9.

6.6.10 Technical and Editorial Corrections to ITU-T Recommendation H.450.10 (2000)

There are no corrections for H.450.10.

6.6.11 Technical and Editorial Corrections to ITU-T Recommendation H.450.11 (2000)

There are no corrections for H.450.11.

6.6.12 Technical and Editorial Corrections to ITU-T Recommendation H.450.12 (2001)

6.6.12.1 Technical Correction

Description:	The receipt of a CmnInform APDU at User A's Endpoint is not described. Therefore add the text below at the end of section 7.1.1.1 ANF-CMN invocation.

[Begin Correction]

7.1.1.1 ANF-CMN invocation

…

Upon receipt of a CmnInform invoke APDU in any message, the Originating endpoint shall remain in the current state.

[End Correction]

6.6.12.2 Add definition of the states CMN-Wait-Response and CMN-Wait-Answer-Response

Description:	The states CMN-Wait-Response and CMN-Wait-Answer-Response are used only in the SDL diagrams but are not defined anywhere. To avoid confusion, a definition of their meaning is added in section 13.

[Begin Correction]

13. Specification and Description Language (SDL) Diagrams for ANF-CMN

...

In the following SDLs the states CMN-Wait-Response and CMN-Wait-Answer-Response are used to describe the behavior of the Endpoints using explicit primitive exchange.

The state CMN-Wait-Response is entered at the Endpoint after a primitive CMNRequest indication is received and the previous state was CMN-Idle.

The state CMN-Wait-Answer-Response is entered at the Endpoint after a primitive CMNRequest indication is received and the previous state was CMN-Wait-Answer.

[End Correction]

6.6.12.3 Redesign the SDL Diagrams, add two missing collision branches and delete an erroneous message symbol

Description:	Two collision branches are missing: add in section 13.1 Figure 8/H.450.12 the possible receipt of a CMNInform request from the application in state CMN-Wait-Answer and in Figure 9/H.450.12 the possible receipt of a CMNRequest request in state CMN-Wait-Response. In Figure 9/H.450.12 the receipt of a CMNInform Request in state CMN-Wait-Response shall be ignored and the message with CMNInform invoke APDU shall not be forwarded to endpoint B.

[Begin Correction]

Editorial - Replace the indicated diagrams by the following:

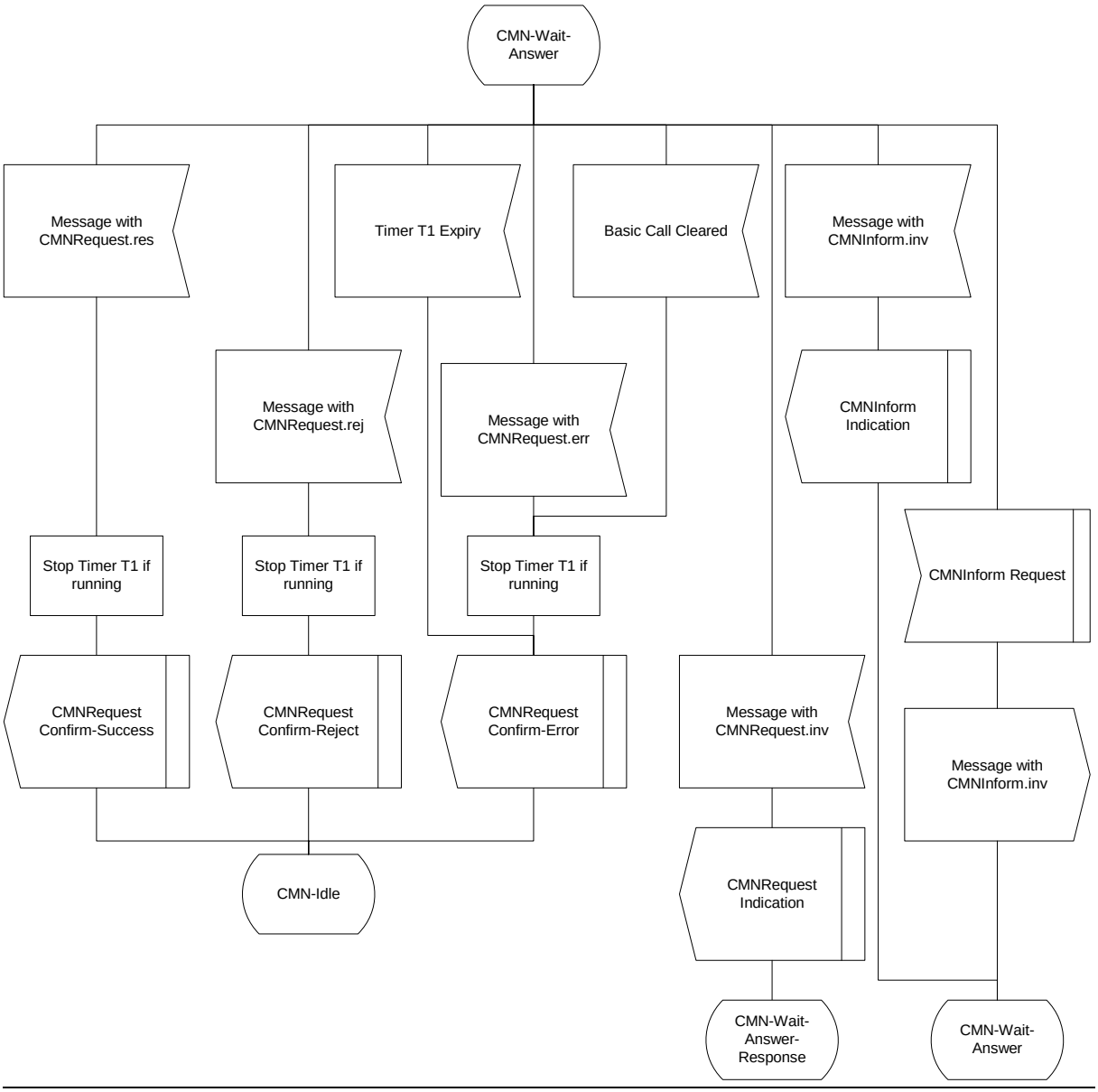


Figure 8/H.450.12 – SDL Representation of ANF-CMN at Endpoint A (Part 3)

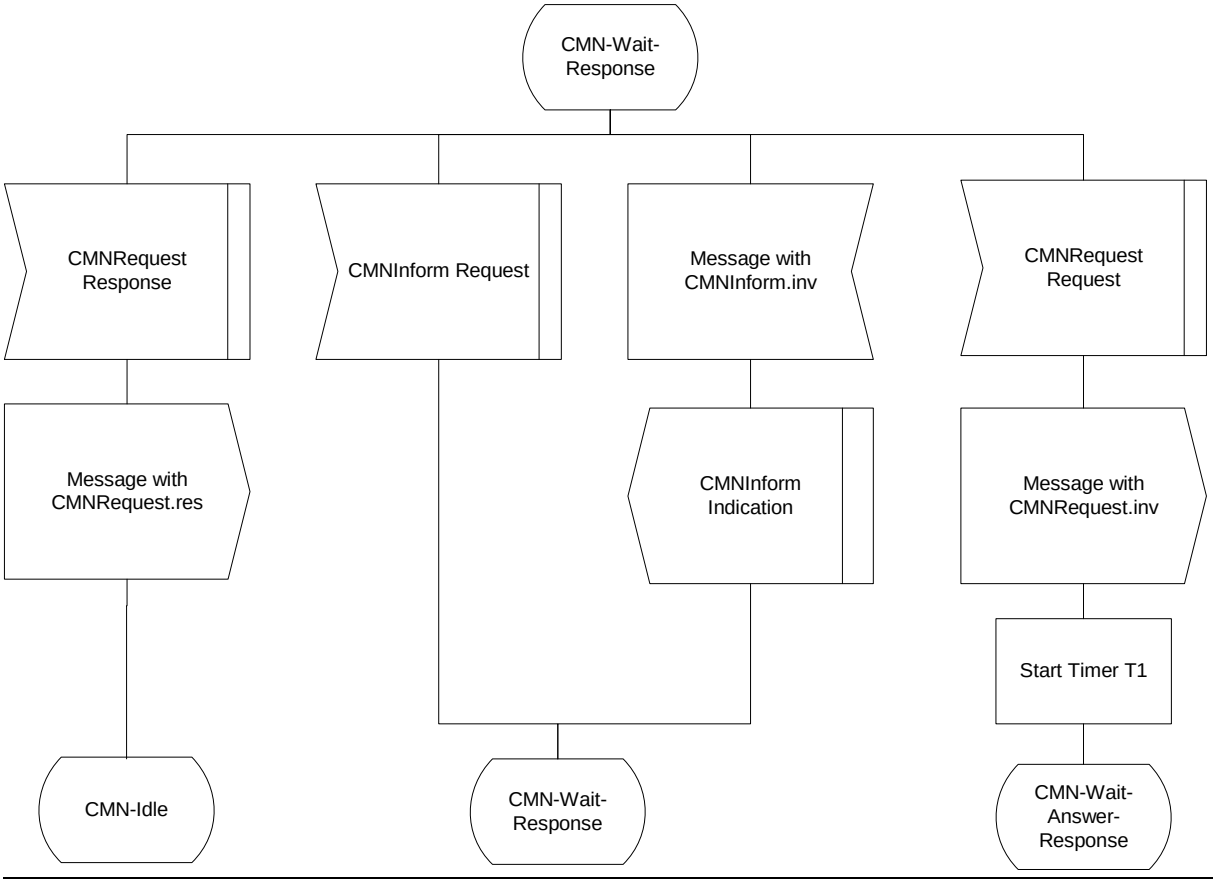


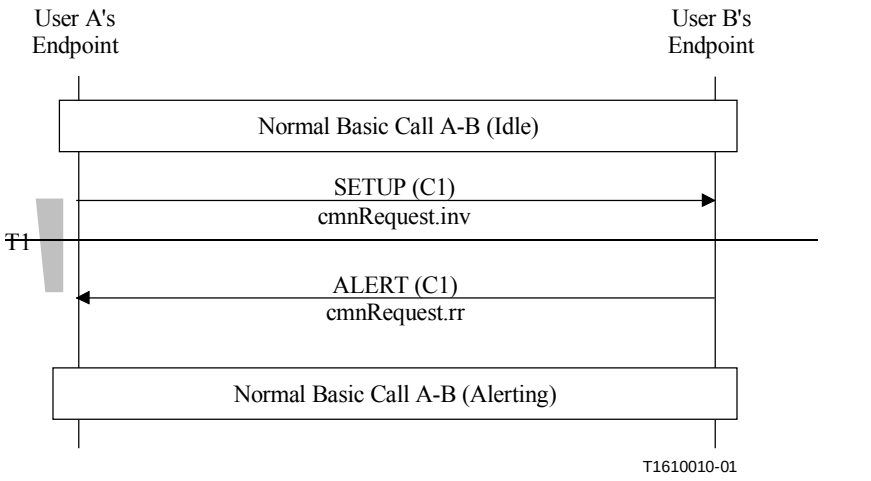
Figure 9/H.450.12 – SDL Representation of ANF-CMN at Endpoint A (Part 3)

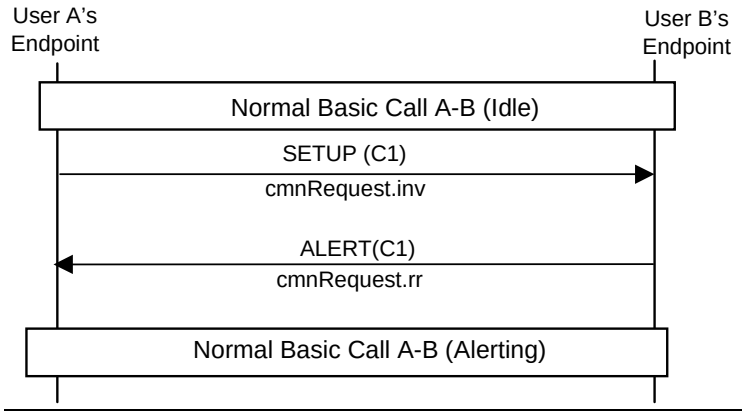
[End Correction]

6.6.12.4 Message Flow of ANF-CMN

Description:	Timer T1 is started if **cmnRequest** invoke is sent in FACILITY message, but not if it is sent in a SETUP message. However, the message flow diagram in Figure 2/H.450.12 erroneously contains timer T1. The erroneous diagram should be replaced by the corrected diagram as below.

[Begin Correction]





[End Correction]

6.7 Technical and Editorial Corrections to ITU-T Recommendation H.341 (1999)

6.7.1 Corrections to the RAS MIB in H.341

Description:	A few editorial errors have been identified in the RAS MIB in H.341. The following text describes the necessary corrections.

1) **RasAdmissionTableEntry** SEQUENCE, the field **RASAdmissionCallIdentifier** is inserted twice. The second entry shall be removed.

2) Each field in **CallSignalStatsEntry** SEQUENCE referred to the number of messages received ("In") and the number of messages transmitted ("Out"). These counters shall be combined. The new **CallSignalStatsEntry** SEQUENCE is shown below:

[Begin Correction]

```
CallSignalStatsEntry::= SEQUENCE {
                callSignalStatsCallConnectionsIn
                     Counter32,
                callSignalStatsCallConnectionsOut
                     Counter32,
                callSignalStatsAlertingMsgsIn
                     Counter32,
                callSignalStatsAlertingMsgsOut
```

```
                    Counter32,
                callSignalStatsCallProceedingsIn
                    Counter32,
                callSignalStatsCallProceedingsOut
                    Counter32,
                callSignalStatsSetupMsgsIn
                    Counter32,
                callSignalStatsSetupMsgsOut
                    Counter32,
                callSignalStatsSetupAckMsgsIn
                    Counter32,
                callSignalStatsSetupAckMsgsOut
                    Counter32,
                callSignalStatsProgressMsgsIn
                    Counter32,
                callSignalStatsProgressMsgsOut
                    Counter32,
                callSignalStatsReleaseCompleteMsgsIn
                    Counter32,
                callSignalStatsReleaseCompleteMsgsOut
                    Counter32,
                callSignalStatsStatusMsgsIn
                    Counter32,
                callSignalStatsStatusMsgsOut
                    Counter32,
                callSignalStatsStatusInquiryMsgsIn
                    Counter32,
                callSignalStatsStatusInquiryMsgsOut
                    Counter32,
                callSignalStatsFacilityMsgsIn
                    Counter32,
                callSignalStatsFacilityMsgsOut
                    Counter32,
                callSignalStatsInfoMsgsIn
                    Counter32,
                callSignalStatsInfoMsgsOut
                    Counter32,
                callSignalStatsNotifyMsgsIn
                    Counter32,
                callSignalStatsNotifyMsgsOut
                    Counter32,
                callSignalStatsAverageCallDuration
              Integer32
                }
```

[End Correction]

3) In **RasRegistrationTableEntry** SEQUENCE, **rasRegistrationEndpointType** is defined to be type "**Integer32**" and should be defined as type "**MmH323EndpointType**".

6.7.2 Support for Expanded Country Code Values in T.35

Description:	T.35 (1999) expanded the available country codes from one octet to two octets. In order to support the expanded country codes going forward, it is recommended that implementers make the following changes to these definitions in H.341.

[Begin Correction]

```
h323TermSystemt35CountryCode OBJECT-TYPE
            SYNTAX INTEGER (0..255)
            MAX-ACCESS read-only
            STATUS current
```

```
            DESCRIPTION
          "Country code, per T.35 Annex A."
 ::= { h323TermSystemEntry 5 }
 h323TermSystemt35CountryCodeExtention OBJECT-TYPE
            SYNTAX INTEGER (0..255)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
"Assigned nationally, unless the country code
is 255, in which case this value shall contain
the country code found in T.35 Annex B."
::= { h323TermSystemEntry 6 }
```

[End Correction]

6.8 Technical and Editorial Corrections to Annex G/H.225.0 (2002)

There are no corrections to this document.

6.9 Technical and Editorial Corrections to Annex C/H.246 (2000)

6.9.1 Additional Message Mappings

Description:	ISUP messages Release, Release Complete, Suspend and Resume are added to Table 1

[Begin Correction]

ISUP message	H.225.0 message
Release (REL)	RELEASE COMPLETE
Release Complete (RLC)	NA
Suspend (SUS)	NA
Resume (RES)	NA

[End Correction]

6.9.2 Changes for Call Diversion

Description:	Changes are made to Table 2 for call diversion information, original called number, redirection information, redirection number, redirection number restriction and subsequent number. Generic notification indicator is added.

[Begin Correction]

ISUP parameter	H.225.0 Information element
Call diversion information	~~NA~~ Notification indicator (non-H.450.3 endpoint) divertingLegInformation1 (H.450.3 endpoint) – see tables 29, 30, 31
Generic notification indicator	Notification indicator (non-H.450.3 endpoint) divertingLegInformation1 (H.450.3 endpoint) – see tables 29, 30
Original called number	~~NA~~ divertingLegInformation2 (H.450.3 endpoint)
Redirection information	~~NA~~ divertingLegInformation2 (H.450.3 endpoint)
Redirection number	~~NA~~divertingLegInformation1 (H.450.3 endpoint) – see table 31
Redirection number restriction	~~NA~~divertingLegInformation1 (H.450.3 endpoint) - see table 31
Subsequent number	~~NA~~Called party number

[End Correction]

6.9.3 Redirecting Number Replaced with Call Diversion and Redirection Number

Description:	In sections C.6.1.3, C.6.1.4, C.6.1.5 and C.6.1.6 redirecting number is removed, call diversion information and redirection number restriction are added.

[Begin Correction]

~~Redirecting number~~

~~NA~~

Call diversion information

See C.6.2.6

Redirection number restriction

See C.6.2.6

[End Correction]

6.9.4 Call Diversion with and without H.450.3

Description:	Section C.7.2.8.3 now describes the mapping of the redirecting number, redirection information and original called number in a diverted call that is presented at an H.450.3 capable end-point from the PSTN. It also describes the mapping of the redirection number sent in the backward direction from the H.323 network to the PSTN.

[Begin Correction]

C.7.2.8.3 Interworking at the exchange where a diverted call is presented to a H.323 network

~~For further study.~~

C.7.2.8.3.1 Gateways supporting H.450.3

If a PSTN to H.323 gateway receives an IAM message containing redirecting number and redirection information parameters it forwards a H.225 SETUP message that includes an H.450.3 divertingLegInformation2 invoke APDU. The gateway is to operate as a combined H.450.3 rerouting endpoint and H.450.3 calling endpoint. The original called number may also be present in the IAM message.

Table A/Annex C - Mapping ISUP redirecting parameters to H.450.3 APDU

IAM ->	SETUP ->
	divertingLegInformation2
Redirecting number	divertingNr
Redirection information Redirecting reason Redirection counter Original redirection reason	 diversionReason diversionCounter originalDiversionReason
Original called number	originalCalledNr

If the gateway receives an ALERTING, CONNECT or FACILITY message that contains a divertingLegInformation3 invoke APDU it sends an ISUP message to the calling party.

Table B/Annex C – Mapping of H.450.3 APDU fields to ISUP parameters

<- ACM, CPG, ANM	<- ALERTING, FACILITY, CONNECT
	divertingLegInformation3
Generic notification indicator *Call is diverting*	
Redirection number	redirectionNr
Redirection number restriction	presentationAllowedIndicator

C.7.2.8.3.2 Gateways not supporting H.450.3

If a gateway that does not support H.450.3 procedures receives an IAM message containing redirecting number and redirection information parameters it maps these parameters to a H.225.0 SETUP message that includes a redirecting number information element as shown in Table C. In the case of multiple diversions within the PSTN an original called number parameter may be present in the IAM message. In this case two redirecting number information elements are included in the SETUP message as shown in Table D: the first redirecting number information element is for the first diversion and the second redirecting number information element is for the last diversion.

Table C/Annex C - Mapping of ISUP redirecting parameters for a non-H.450.3 gateway – single diversion

IAM ->	SETUP ->
Redirecting number parameter Nature of address (1) Numbering plan (2) Address signal (3)	Redirecting number information element Type of number (1) Numbering plan (2) Reason for diversion (4) Number digits (3)
Redirection information parameter Redirecting reason (4)	
The numbers in parentheses show the mapping of individual fields	

Table D/Annex C - Mapping of ISUP redirecting parameters for a non-H.450.3 gateway – multiple diversions

IAM ->	SETUP ->
Redirecting number parameter Nature of address (1) Numbering plan (2) Address signal (3)	Redirecting number information element Type of number (6) Numbering plan (7) Reason for diversion (5) Number digits (8)

Redirection information parameter Redirecting reason (4) Original redirection reason (5)	
Original called number parameter Nature of address (6) Numbering plan (7) Address signal (8)	Redirecting number information element carried as H.460.5 information Type of number (1) Numbering plan (2) Reason for diversion (4) Number digits (3)
The numbers in parentheses show the mapping of individual fields	

6.9.5 New Release Complete / Cause Mappings

Description:	New Release Complete reasons were added to H.225.0 (1999), which need to be represented in Annex C/H.246. Below show the modifications to the relevant tables.

[Begin Correction]

Table 15/ANNEX C – Call clearing from the user

RELEASE COMPLETE→	**REL→**
Cause information element	Cause parameter
Cause value No. x	Cause value No. x (Notes 1 and 2)
ReleaseCompleteReason	Cause parameter
newConnectionNeeded	47 – Resource Unavailable
nonStandardReason	127 – Interworking, unspecified
replaceWithConferenceInvite	31 – Normal, unspecified

Table 52/ANNEX C – Call clearing during call establishment

←REL	**←RELEASE COMPLETE**
Cause parameter	Cause information element
Cause value No. x (Notes 1)	Cause value No. x
Cause parameter	ReleaseCompleteReason
47 – Resource Unavailable	newConnectionNeeded
127 – Interworking, unspecified	nonStandardReason
31 – Normal, unspecified	replaceWithConferenceInvite

[End Correction]

6.9.6 Single 64kbps Bearer FFS in Table 3

Description:	Technical corrections to Tables 3 and 6 of section C.6.1.1 are shown below. These corrections have to do with a single 64kbps bearer channel.

[Begin Correction]

Table 3/ANNEX C – Coding of the transmission medium requirement parameter (TMR)
One BC received

SETUP→		**IAM→**
Bearer capability information element		Transmission medium
Information transfer capability	Information transfer rate	requirement parameter
Speech	Value non-significant	*Speech*
3.1 kHz audio	Value non-significant	*3.1 kHz audio*
Restricted digital information	For further studies	For further studies
Unrestricted digital information ~~Or~~ ~~*Unrestricted digital information with tones/announcements*~~	*64 kbit/s unrestricted*	~~*3.1 kHz audio*~~ *FFS*
	2 × 64 kbit/s unrestricted	*2 × 64 kbit/s*
	384 kbit/s unrestricted	*384 kbit/s*
	1536 kbit/s unrestricted	*1536 kbit/s*
	1920 kbit/s unrestricted	*1920 kbit/s*
	Multirate: 6 x 64 kbit/s	*384 kbit/s*
	Multirate: 24 x 64 kbit/s	*1536 kbit/s*
	Multirate: 30 x 64 kbit/s	*1920 kbit/s*
NOTE: For a call originated from an H.323 endpoint, the Rate Multiplier shall be used to indicate the bandwidth to be used for this call. If a gateway is involved, then this value shall reflect the number of external connections to be set up. The bandwidth needed for the call is the bandwidth needed on the SCN side, and may or may not match the bandwidth allowed on the packet-based network by the ACF H.225.0 RAS messages.		

…

Table 6/ANNEX C – Coding of the user service information parameter (USI)

SETUP→	**IAM→**
Content	User service information parameter
BC	BC (Note 1)
NOTE 1 – The BC should be the same as that received in the SETUP with the exception of when the BC is 1x64k ~~it should be replaced with 3.1kHz Audio~~. 1x64k BC is for further study.	

[End Correction]

6.9.7 Handling the Suspend Message

Description:	Technical corrections were applied to C.6.1.11 as described below.

[Begin Correction]

C.6.1.11 Receipt of the Suspend message (SUS) network initiated

The actions taken on the ISUP side upon receipt of the Suspend message (SUS) are described in 2.4.1/Q.764 [1].

There is no support for Suspend message (SUS) network initiated on the H.225 side, so the actions taken should be the actions as described in Q.764 for the controlling exchange.

[End Correction]

6.9.8 Handling the Resume Message

Description:	Technical corrections were applied to C.6.1.12 as described below.

[Begin Correction]

C.6.1.12 Receipt of the Resume message (RES) network initiated

The actions taken on the ISUP side upon receipt of the Resume message (RES) are described in 2.4.1/Q.764 [1].

There is no support for Resume message (RES) network initiated on the H.225.0 side, so the actions taken should be the actions as described in Q.764 for the controlling exchange.

[End Correction]

6.9.9 Editorial Corrections to Table 28

Description:	Editorial corrections were applied to Table 28 in C.6.2.3.

[Begin Correction]

Table 28 / ANNEX C Connected Party Number	
←CONNECT	**←ANM/CON**
Connected ~~Party~~ Number	**Connected ~~Party~~ Number** Or (note) **Generic Number** **(-additional Connected ~~Party~~ number)**
ConnectedAddress	**Connected ~~Party~~ Number**
Note: If an additional Connected ~~Party~~ number is included in the Generic Number then the additional Connected ~~party~~ number should be sent in the Connected ~~Party~~ number.	

[End Correction]

6.9.10 Technical Correction Relating to Sending ACM

Description:	Section C.7.1.3 contains a technical error in the assignment of the values of M, K and I. The corrected text is shown below.

[Begin Correction]

C.7.1.3 Sending of the Address Complete Message (ACM)

...

Backward call indicators

...

If bit I is ~~1~~ 0 then:

bit K ISDN user part indicator

1 *ISDN user part used all the way*

~~If bit I is 0 then:~~

bit M ISDN access indicator

~~0~~1 *terminating access ~~non~~ ISDN*

[End Correction]

6.9.11 Clarification of Cut-Through Behavior

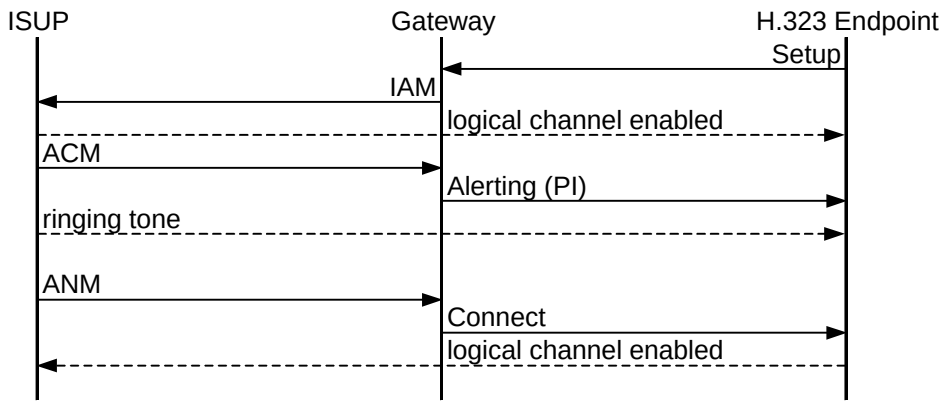
Description:	The following additional text is necessary to clarify cut-through behavior in Annex C.

[Begin Correction]

C.6 Outgoing call - Interworking from H.225.0 to ISUP

In traditional telephone networks, cut-through occurs very early in the call (before the called party answers) to provide tones or announcements, and to eliminate clipping on answer while the voice channel is being connected end-to-end. Section 8.1.7.4/H.323 describes the behavior for early cut-through (that is, cut-through before the H.225 Connect message).

For calls from the packet network to the circuit network, the best behavior would be to cut through in the backward direction on IAM, and on the forward direction on answer (to avoid fraud):



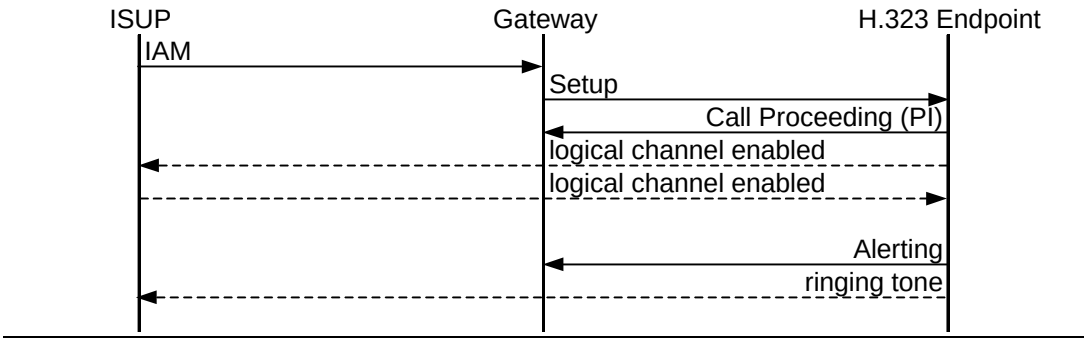
The notation “Alerting (PI)” indicates the presence of the progress indicator as described in section 8.1.7.4/H.323.

[End Correction]

[Begin Correction]

C.7 Incoming call - Interworking from ISUP to H.225

In general, operation with an SS7 network where the call is from the circuit network to the packet network would be best if media is cut-through in both directions on the IAM (that is, cut-through occurs on the first response to a Setup in the H.323 network) as shown in the following diagram:



The notation “Call Proceeding (PI)” indicates the presence of the progress indicator as described in section 8.1.7.4H.323.

[End Correction]

6.9.12 Removal of Tones and Announcements from Bearer Capability

Description:	Information Transfer Capability field in Bearer Capability IE in H.225.0 does not contain encoding for “Unrestricted Digital Information With Tones/Announcements” but only for “Unrestricted Digital Information” or “Restricted Digital Information”. The reference to “Tones/Announcement” should be deleted from the mapping of the Bearer Capability IE in Setup Transmission Medium Requirement Parameter in IAM. Table 3/Annex C, Table 7/Annex C, Table 9/Annex C, and Table 12/Annex C should be changed as below.

[Begin Correction]

Editorial - For changes to Table 3/Annex C, please refer to section 6.9.6 in this document.

[End Correction]

[Begin Correction]

Table 7/ANNEX C – Receipt of ACM with a cause parameter

←PROGRESS	←ACM
Cause information element (Note 1) Progress indicator No. 8 (Note 2)	Cause parameter Optional backward call indicators parameter In-band information ind. *In-band info…*
NOTE 1 – If the cause value received in the Address Complete Message (ACM) is unknown in H.225.0, the unspecified cause value of the class is sent. NOTE 2 – The progress indicator No. 8 *(in-band information or an appropriate pattern is now available)* is only sent if the BC received in the SETUP message is coded *speech~~,~~ or 3.1 kHz audio ~~or unrestricted digital information with tones/announcements~~*. NOTE 3 – If a bearer is available then end interwork should apply the far end tone/announcement.	

[End Correction]

[Begin Correction]

Table 9/ANNEX C – Sending criteria of the progress indicator information elements created by the ~~originating exchange~~interworking function

←H.225.0 Message sent (See Table 8)	**←ACM**
Progress indicator information element	Content
No. 1 (*Call is not end-to-end ISDN: further progress information may be available in-band*)	Backward call indicators parameter ISDN User Part indicator 0 *ISDN User Part not used all the way*
No. 2 (*Destination address is non-ISDN*)	Backward call indicators parameter ISDN User Part indicator 1 *ISDN User Part used all the way* ISDN access indicator 0 *Terminating access non-ISDN*
No. 8 (Note) (*In-band information or appropriate pattern now available*)	Optional backward call indicators parameter In-band information indicator 1 *In-band info...*
NOTE – The progress indicator No. 8 (*in-band information or an appropriate pattern is now available*) is only sent if the BC received in the SETUP message is coded *speech~~,~~ or 3.1 kHz audio ~~or 1 x 64kHz unrestricted digital information~~*.	

[End Correction]

[Begin Correction]

Table 12/ANNEX C – Sending criteria of the progress indicator information elements created by the ~~originating exchange~~interworking function

←H.225.0 Message sent (See Table 11)	**←CPG**
Progress indicator information element	Content (Note 2)
No. 1 (*Call is not end-to-end ISDN: further progress information may be available in-band*)	Backward call indicators parameter ISDN User Part indicator 0 *ISDN User Part not used all the way*
No. 2 (*Destination address is non-ISDN*)	Backward call indicators parameter ISDN User Part indicator 1 *ISDN User Part used all the way* ISDN access indicator 0 *Terminating access non-ISDN*
No. 4 (*Call has returned to the ISDN*)	Backward call indicators parameter ISDN User Part indicator 1 *ISDN User Part used all the way* ISDN access indicator 1 *Terminating access ISDN* whereas the last indication received was "0", *Terminating access non-ISDN*
No. 8 (Note 1) (*In-band information or appropriate pattern now available*)	Event information parameter Event indicator 000 0011 *In-band info ...*
No. 8 (Note 1) (*In-band information or appropriate pattern now available*)	Optional backward call indicators parameter In-band information indicator 1 *In-band info ...*
NOTE 1 – The progress indicator No. 8 (*in-band information or an appropriate pattern is now available*) is only sent if the BC received in the SETUP message is coded *speech*~~,~~ or *3.1 kHz audio* ~~or 1 x 64kHz *unrestricted digital information*~~. NOTE 2 – The mapping of the contents in the CPG message is only relevant if the information received in the message is different compared to earlier received information, e.g. in the ACM message or a CPG message received prior to this message.	

[End Correction]

6.9.13 Sending of Progress Indicator

Description:	There is an error in some descriptions regarding the generation of progress indicator in various H.225 messages as a result of receiving ISUP messages. This is corrected in the following updates.

[Begin Correction]

Table 8/ANNEX C – Message sent to the H.225.0 upon receipt of ACM

←Message sent to the H.225	←ACM
	Backward call indicators parameter Called party's status indicator
CALL PROCEEDING when not been sent before (Note 1), otherwise: – PROGRESS if a progress indicator information element is to be sent (Note 2) – No message if no progress indicator information element is to be sent (Note 2, 4)	00 *No indication*
ALERTING	01 *Subscriber free (Note 3)*
NOTE 1 – The receipt from the network of an Address Complete Message (ACM) without the *subscriber free* indication is interpreted by the network as a sending complete indication, in the case where the network couldn't determine it before. NOTE 2 – The sending of a progress indicator information element is described below. NOTE 3~~– If the ACM does not contain a progress indicator the Interworking function should set Progress Indicator to 1 or 8.~~ ~~NOTE 4 –~~ The FACILITY message may be used anyway by the interworking function to transfer H.225.0 internal information e.g. the fastStart parameter. For the coding of the FACILITY message see Table 14/H.225.0 [7].	

…

Progress indicator

Progress indicator information elements possibly present in the access transport parameter of the Address Complete Message (ACM) are transferred into the message sent to the calling user. If the calling user is an H.323 end system it need not interpret this information element.

In addition, progress indicator information elements are created by the Interworking function according to the coding of the Address Complete Message (ACM). Table 9 shows the sending criteria of each value.

Every message sent to the access (ALERTING, CALL PROCEEDING or PROGRESS) may contain two progress indicator information elements. When more than two progress indicator information elements are to be sent, the supplementary progress indicator information elements are sent in a PROGRESS message.

[End Correction]

[Begin Correction]

Table 10/ANNEX C – Receipt of CPG with a cause parameter

←PROGRESS	←CPG
Cause information element (Note 1)	Cause parameter
Progress indicator No. 8 (Note 2)	Event information parameter Event indicator *In-band info…* Or Optional backward call indicators parameter In-band information ind. *In-band info…*
NOTE 1 – If the cause value received in the Call Progress Message (CPG) is unknown in H.225, the unspecified cause value of the class is sent. NOTE 2 – The progress indicator No. 8 (*in-band information or an appropriate pattern is now available*) is only sent if the BC received in the SETUP message is coded *speech*~~,~~ or *3.1 kHz audio* ~~or 1 x 64kHz *unrestricted digital information*~~. NOTE 3 – ~~If the CPG does not contain a progress indicator the Interworking function should set Progress Indicator to 1 or 8.~~ ~~NOTE 4 –~~ If the bearer is established the interwork function should initiate far end tone/announcement.	

[End Correction]

6.9.14 Editorial Corrections

Description:	There are several inaccuracies in the document. The following changes correct them.

Editorial - The Numbering plan indicator of the Called party number should be encoded as per the changed bit pattern below.

[Begin Correction]

C.6.1.1 Sending of the Initial Address Message (IAM)

…

Called party number

…

– Numbering plan indicator:

0~~0~~1 *ISDN (telephony) numbering plan (Recommendation E.164)*

[End Correction]

Editorial - Table 13/Annex C should be corrected to indicate that the interworking function and not the originating exchange creates the progress indicator information elements.

[Begin Correction]

Table 13/ANNEX C – Sending criteria of the progress indicator information elements created by the ~~originating exchange~~interworking function

←CONNECT	←ANM
Progress indicator information element	Content
No. 1 (*Call is not end-to-end ISDN: further progress information may be available in-band*)	Backward call indicators parameter ISDN User Part indicator 0 *ISDN User Part not used all the way*
No. 2 (*Destination address is non-ISDN*)	Backward call indicators parameter ISDN User Part indicator 1 *ISDN User Part used all the way* ISDN access indicator 0 *terminating access non-ISDN*
No. 4 (*Call has returned to the ISDN*)	Backward call indicators parameter ISDN User Part indicator 1 *ISDN User Part used all the way* ISDN access indicator 1 *terminating access ISDN* whereas the last indication received was "0" *terminating access non-ISDN*

[End Correction]

Editorial - The word “subaddress” should be inserted in the paragraph as below.

[Begin Correction]

C.7.1.1 Sending of the SETUP message

…

Calling party subaddress

In the case of GK routed call the interworking function should send the Calling Party Subaddress as received from the ISUP in the Access Transport Parameter.

[End Correction]

6.9.15 Progress Indicator Usage in Setup

Description:	Since Progress Indicator IE is allowed in an H.225.0 Setup message, the text that indicates that it is forbidden should be removed.

[Begin Correction]

C.6.1.1.2 Optional Parameters

…

Access transport

~~Progress indicator is forbidden in a SETUP message.~~

The High layer compatibility and Low layer compatibility is FFS.

Called Party subaddress and Calling Party subaddress may be mapped to the IAM Access Transport parameter.

[End Correction]

6.10 Technical and Editorial Corrections to Annex E/H.323

6.10.1 Editorial Corrections to Improve Readability

Description:	H.323 Annex E contains a number of ambiguous statements, which have created confusion among vendors attempting to implement the Annex. This section details editorial changes to the document, which should add clarity to the text.

[Begin Correction]

E.1.1.6 Sender sequence number policy

Assigned per host-address ~~+~~ and source-port, sending ~~applications~~ Annex E layers shall start with some random value, incrementing by 1 for every PDU sent. If the sequence number reaches 224 (16 777 216) it shall wrap around to 0.

[End Correction]

[Begin Correction]

E.1.1.7 Receiver sequence number policy

When receiving a UDP packet, the ~~application~~ Annex E layer shall check the host-address~~+~~, source-port~~+~~, and sequence number to recognize duplicate messages. The ~~application~~ Annex E layer may re-order messages according to sequence numbers and recognize packet-loss when finding gaps in sequence numbers.

[End Correction]

[Begin Correction]

E.1.1.8 Retransmissions

...

When there is a known ~~request/reply~~ roundtrip message interval value from a previous transmission, timer T-R1 should be set to ~~the~~ that roundtrip message interval value +10%.

[End Correction]

[Begin Correction]

E.1.1.10 Forward error correction

Annex E messages may be sent more than once to enable forward error correction. If the arrival of a message is crucial, the ~~application~~ Annex E layer may choose to send the same message twice (without incrementing the sequence number). If both messages arrive, the second one will be treated as normal message duplication.

[End Correction]

[Begin Correction]

E.1.4.2.2.4 Restart Message

...

If a restart does not affect on-going calls, then it is invisible to the ~~application~~Annex E layer, and therefore shall not be signalled.

[End Correction]

[Begin Correction]

E.1.2.2 Serial model

In the serial-model, when a PDU is sent, ~~the application (or rather~~ the Annex E ~~stack)~~layer waits until a positive reply is returned for the same Session-Identifier. This behaviour is used for protocols that cannot sustain out-of-order message arrival and require real-time operations while sending small amounts of information. An example of such a protocol is Q.931.

When using this model, the Ack-flag shall always be set for static-typed messages. Unless otherwise specified, Annex E implementations shall use the default retransmission timers (**T-R1** and **T-R2**) and counter (**N-R1**).

[End Correction]

[Begin Correction]

E.2.2.1 UDP-based procedure

...

~~Applications~~ The Annex E layer should retransmit a lost packet if it does not get a reply after some time. The precise retransmission procedure is detailed in E.1.1.8.

[End Correction]

[Begin Correction]

E.2.2.2 Mixed TCP and UDP procedure

...

This means that backwards compatibility when calling H.323 version 1 (1996) or 2 (1998) entities is transparent, as the v1/v2 H.323 application will not be aware of the UDP packet.

[End Correction]

6.10.2 Usage of ports in H.225.0 signaling over Annex-E

Description:	The following issues are found with the current description: • The current text implies that an entity cannot receive messages on any port other than the well-known port 2517. • The current text does not specify the port to which Annex E acknowledgements should be sent. According to the sender sequence number policy, the sequence numbers are assigned per host-address and source-port. Annex E acknowledgements need to be sent to the same source-port from which the PDU was sent. If the endpoint does not follow this rule then the receiver of the acknowledgements would have no way to correlate them to the original messages, when the endpoint sends from multiple ports. • The current text implicitly allows an incoming endpoint to send Annex E acknowledgements/ messages from a different port than the advertised/well-known Annex E port. If this is allowed, then the originator on receiving messages will not be able to correlate messages with the original call. • An entity may use any port as source-port for a new call origination. For the reasons listed above, an endpoint shall use the same port throughout the duration of a call in order to ensure that messages are properly correlated.

[Begin Correction]

E.2.3.2 Well-known port

UDP port 2517 shall be used for the well-known port. A single H.323 entity on a physical device shall use a single, distinct UDP port as the advertised port for receiving messages. However, it may utilize a distinct port on each interface if the physical device has multiple network interfaces. ~~Entities may transmit from any random port.~~

The calling entity shall send all Annex E messages for a call to the called entity's advertised destination port. The called entity shall send all Annex E messages related to said call to the IP address and port from which the initial Annex E message for the call was received. The called entity shall send all Annex E messages using the same port on which it received the initial H.225.0 PDU from the caller.

The calling entity may transmit messages from any random port, but shall use the same port throughout the duration of the call.

[End Correction]

6.10.3 Sequencing of Annex E messages

Description:	A new subsection stating the receiver sequence policy needs to be added for H.225.0 over Annex E due to the following considerations: • The current text in the section E.1.2.2 "Serial model" discusses the role of "Session-Identifier" in treating positive acknowledgements. The clarification in this proposal, explicitly mentions the need to use "Session-Identifier" in the serial model, by the sender as well. • The current text in the section E.1.1.7 "Receiver sequence number policy" discusses message re-ordering by the application. This new section in the specifics clarifies in this regard, for the serial model, and disallows the re-ordering for H.225.0 over AnnexE. The re-ordering is not needed because H.225 for Annex E follows serial model. • The current text in the section E.1.1.7 "Receiver sequence number policy" allows an entity to recognize gaps in the sequence number as a message loss. For the H.225.0 over Annex E, the send sequence numbers are assigned based on host-address and source-port. Hence, there may be gaps when seen by the receiver based on host-address, source-port and sequence-number.

[Begin Correction]

E.2.3.9 Receiver sequence number policy for H.225.0 over Annex E

When receiving a H.225.0 message over Annex E, an entity shall check the host-address, source-port and sequence-number to recognize duplicate messages. The transmitting entity follows serial model for the same Session-Identifier and assigns sequence numbers per host-address and source-port. Since, for a single H.323 call, it is not possible for messages to get out of order, the Annex E layer shall not attempt to reorder messages according to sequence numbers. Gaps in the sequence numbers are possible and an entity shall not recognize it as a packet loss.

[End Correction]

6.10.4 Usage of Restart messages

Description:	The current text in the section E.1.4.2.2.4 "Restart Message" specifies use of Restart message to signal loss of all active calls and to reset sequence number space. The sending of Restart message is optional. If an endpoint does not signal restart to the remote endpoint, then a scenario may develop where the receiving endpoint may mistakenly discard messages. The explicit resetting of the number space is useful when a restarted endpoint randomly

	selects the same sequence number as the one it used for the last message before restart. If this happens, then the receiving endpoint might think that the message is a duplicate one. The Restart message helps the sender to signal restart and allows the receiver to adjust its duplicate detection mechanism. With this new proposal an endpoint will be able to send restart message to reset sequence number range without affecting existing calls. The restarted endpoint shall send Restart message with or before the first H.225.0 message. A new field is being proposed in the Restart message, which will let the receiver know if the calls are disconnected or not. The receiver shall tear down calls, or start recovery procedures on receiving Restart message. The use of Restart message is optional but highly recommended.

[Begin Correction]

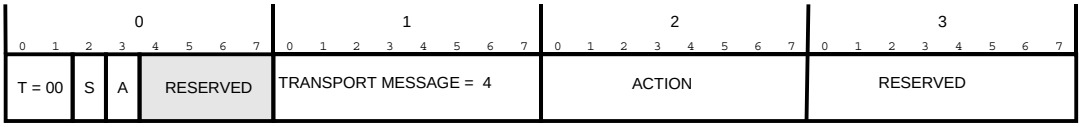
E.1.4.2.2.4 Restart Message

The following structure shall be used to encode ANNEX E Restart payloads. The transport-message octet shall be set to 3. ~~Restart payloads are used to signal to the remote peer that it has restarted, and that all active calls have been disconnected. Any message arriving from the previous sequence-number range shall be considered stale and ignored. All outstanding calls that were related to the state of the system before the restart will be dropped.~~Restart payloads are used to signal to the remote peer that the sender has restarted. Restart payload should be sent as a part of the first message to the remote entity. The receiver shall reset its receiver sequence number range on receiving the Restart payload. It shall consider any message arriving from the previous sequence number range as stale and shall ignore it.

The receiver shall tear down existing calls or start recovery procedures depending on the "action" field in the Restart payload.

If a restart does not affect on-going calls, then it is invisible to the application, and therefore shall not be signalled.

Image/Table: Restart Message Structure



Field	Content of fields	Bits
action	The action desired by the receiver of the Restart payload	8

Action value	meaning
0	Unspecified
1	Tear down calls
2	Start Recovery procedures
3..	Reserved for future use

[End Correction]

6.10.5 Timer and Retry Counter For Failure Detection

Description:	The timer value and number of retries specified in Annex E/H.323 are meant for detection of failure of the peer entity but not sufficient if both entities implement robustness procedures as described by Annex R/H.323. Specfically, the values of T-IMA1 and N-IMA1 at six seconds and at six respectively are considered too high for a successful call re-establishment after the failure of a signalling entity.

[Begin Correction]

Image/Table: I-Am-Alive timers

Item	Value	Comments
T-IMA1	6 seconds	I-Am-Alive transmission interval*
N-IMA1	6	Number of consecutive I-AM-ALIVE messages not responded to after which the remote peer is declared dead.*

* These timers should follow the recommended values in Annex R if Annex R is also being used between the two entities.

[End Correction]

6.11 Technical and Editorial Corrections to ITU-T Recommendation H.283 (1999)

6.11.1 Support for Expanded Country Code Values in T.35

Description:	T.35 (1999) expanded the available country codes from one octet to two octets. In order to support the expanded country codes going forward, it is recommended that implementers take note of the following usage guidelines for fields in H.283.

[Begin Correction]

...

```
H221NonStandard ::= SEQUENCE
{
    t35CountryCode     INTEGER(0..255),    -- country, as per T.35 Annex A
    t35Extension       INTEGER(0..255),    -- assigned nationally, unless
    the
                                                                    --
    t35CountryCode is binary 1111 1111,
                                        -- in which case this field shall
                                                                    -- contain the
    country code found
                                                                    -- in T.35
    Annex B
    manufacturerCode   INTEGER(0..65535)  -- assigned nationally
}
```

...

[End Correction]

6.12 Technical and Editorial Corrections to ITU-T Recommendation Annex-R/H.323

6.12.1 Partial Method A as a Robustness Method

Description:	There is a need to define a new robustness method that is followed by endpoints that implement robustness procedures but do not have a backup of their own. However they will participate in call recovery for calls for which the signalling neighbour implements either Method A or B.

[Begin Correction]

R.5.1 Method A: State recovery from neighbours

In Method A, each entity is aware of the signalling transport addresses for backup entities for each upstream and downstream signalling neighbour. When entities become aware of the failure of their upstream or downstream signalling neighbour, they attempt to connect to one of the backup entities. The backup entity recovers minimal call state from its signalling neighbour using Status and StatusInquiry messages (enhanced with additional fields). Note that in some cases it may be necessary for the neighbour to query its neighbour for call state if it has not kept all the necessary information locally (e.g. a routing gatekeeper may not have cached open logical channel information).

The recovered call state is sufficient to continue the call (forward call signalling and call control signalling and know of open logical channels) but not sufficient to allow the recovered entity to participate in billing and some other services.

R.5.1.1 Partial Method A

There is also a case where an H.323 entity does not itself have a backup entity but it still implements robustness procedure so it can help preserve calls, if its signalling neighbour that does have a backup entity fails.

The H.323 entity that participates in the recovery of stable calls with the backup entity of its signalling neighbour, but does not itself have any backup, is said to implement Partial Method A.

…

R.6.4 Transport address and re-established connections

Both of these solutions (with the possible exception of some fault-tolerant platform solutions) must deal with recovery of the signalling channel using a backup transport address. These must be exchanged when call signalling is established, using the backupCallSignalAddresses fields in Setup and Connect. An entity sends the call signalling address of its backup in both Setup and Connect. An entity receives the call signalling address of the backup entity from its origination-side neighbour when it receives Setup and from its termination-side neighbour when it receives Connect.

An entity that implements Partial Method A shall send an empty **backupCallSignalAddresses** to indicate that it does participate in robustness procedure but it does not itself have a backup.

All entities shall add their own call signal address as the first entry in the **backupCallSignalAddresses** list including the port number they are listening on. This is required for the signalling neighbour (or its backup) to re-establish connection with the entity.

[End Correction]

6.12.2 KeepAlive Messages

Description:	Annex R specifies usage of keepAlive messages as a way to detect the failure of its signalling neighbour. These messages are expected to be sent on each connection. There is an optimization opportunity here where these keepAlive messages need to be sent on a per-pair of signalling entities and not on per connection basis

[Begin Correction]

R.6.3 Detecting failure – KeepAlive

Without a keepalive mechanism, entity failure or failure of the signalling connection will be known only when the connection is used. Annex E provides a keepAlive mechanism to detect the failure even with little traffic. TCP's keepAlive mechanism has too long a timeout to be of use and so with TCP failure might not be detected for an extended time under conditions of low traffic sent to the failed entity. Our small-scale solution depends on failure being detected by both signalling neighbours (connections are always established from the neighbour toward the recovered entity) and so we need keepAlive messages at the H.323-level that can be used with TCP connections. KeepAlive messages are available to be optionally used in H.245. We would specify that Status/Status Inquiry be used periodically over TCP connections to provide this keepAlive mechanism. Although this issue is common, we will see that it is only a significant problem for the Method A, the state recovery from neighbour method.

The entity closer to the called party (destination side of connection or side that uses call reference flag = 1 in CRV used on connection – See ITU-T Q.931 for definition of the call reference flag) shall send StatusInquiry periodically (this is the direction of least traffic during established calls). The period should vary randomly from a configurable maximum value to one half that value in order to avoid congestion. Two seconds is the recommended default maximum, in order to allow detection of failure before other messages timeout. The maximum value shall be included in the StatusInquiry as timeToLive, so that the recipient can also monitor failure without an additional StatusInquiry/Status exchange in the opposite direction. The recipient system needs only to maintain a timer using the indicated maximum value as a timeout.

When multiplexed channels are used, it is not necessary to send StatusInquiry/Status for each call signaled on the channel. A StatusInquiry or Status message with a CRV IE of 0 (zero) and with the field callIdentifier of 0 (zero) applies to all calls using the channel.

KeepAlive messages, especially at the H.323-level, can add significant signalling overhead. But note that only Method A with TCP connections uses these KeepAlives and Method A is for the small-scale case where the number of connections per entity is low. To minimize the overhead, the use of TCP should be avoided. StatusInquiry/Status keepAlives are **not** needed in our large-scale solution.

In order to further minimize the impact of exchanging keepAlives, if there are several calls between the same two entities, StatusInquiry/Status messages need be sent on any one of the connections between the two entities. In order to associate each active call with the correct set of entities, an endpoint GUID shall be included by the originating entity in the Setup message and another by the destination entity in the Connect message. These GUIDs shall be unique to each entity and, in case any entity has more than one signalling interface, shall be generated per interface. If there are multiple H.323 instances on the entity, each instance shall generate a unique GUID. KeepAlive timers shall be maintained on each unique GUID pair. Upon the expiry of the keepAlive timer, any entity may send a StatusInquiry message with a CRV IE of 0 (zero) and with the field callIdentifier of 0 (zero) using any available connection. The signalling neighbour shall respond with a keepAlive Status message.

[End Correction]

6.12.3 Entity Failure Detection Using Annex E/H.323 Methods

Description:	Annex E defines its own keepAlive mechanism. It uses I-Am-Alive message every six seconds and uses six retries of the message to decide failure of signalling neighbor. Robustness using Annex E signalling needs quicker detection of failure and hence these values should be made configurable in Annex R implementations.

[Begin Correction]

R.6.3 Detecting failure – KeepAlive

…

Detecting failures for Annex E connections will use the existing I-Am-Alive messaging. This procedure defines keepAlive messages between the signaling entities based on a timer. This timer uses a value defined by T-IMA1 timer, by default set to 6 seconds. However, in the case where the two entities also implement Annex R, this timer shall be configurable in accordance with recommended values as above. The I-Am-Alive messaging also uses the a counter defined by the N-IMA1, that defines the number of consecutive retries of I-Am-Alive messages before which the signalling neighbour is assumed to have failed. For Annex R enabled entities, this counter is recommended to have a maximum value of two (2).

[End Correction]

6.12.4 Re-assignment of CRV Values for Recovered Calls

Description:	If the backup entity is itself engaged in calls with the signalling neighbour of the failed entity, there is potential of CRV duplication between its own calls and the recovered calls that it is expected to take over. In that case re-assignment of CRV is necessary. The following addition specifies the re-assignment.

[Begin Correction]

R.10.1 Recovery procedures with conflicting CRV values

It is possible that at the time of failure, the active entity and its backup peer are both simultaneously in calls to the same signaling neighbour. In this case there is the remote possibility that both of these calls use the same CRV values with the signalling neighbour and backup peer is not able to continue the call from the failed entity keeping the same CRV. Assignment of a new CRV and communicating it to the signalling neighbour is required.

If the failed entity implements Method A, the signalling neighbour re-establishes a call signalling connection with the backup entity of the failed entity. Then the signalling neighbour shall send StatusInquiry and Status messages to the backup entity. But before sending the StatusInquiry and Status messages, the entity shall check if it is the one that originated (caller side of) the call and if it already has prior calls to the backup entity. If the signalling neighbour is on the caller side of the call and it has prior calls to the recovered entity as shown in Figure R.3, then the signalling neighbour shall assign a new unique CRV value for this call to the recovered entity and use it (in CRV IE) in all subsequent Q.931 and RAS messages. The recovered entity shall assign a unique

CRV value for this call and use it in its communication with the gatekeeper. If the signalling neighbour is on the called side of the call and it has prior calls from the recovered entity as shown in Figure R.4, then the entity shall assign a new unique CRV value in StatusInquiry message with CRV flag = 1 because it is destination side of the call. The recovered entity shall adopt this new CRV for this call. All the subsequent Q.931 messages for this call shall use this new CRV value. The recovered entity, if required, shall assign a unique CRV value for this call to be used in RAS messages.

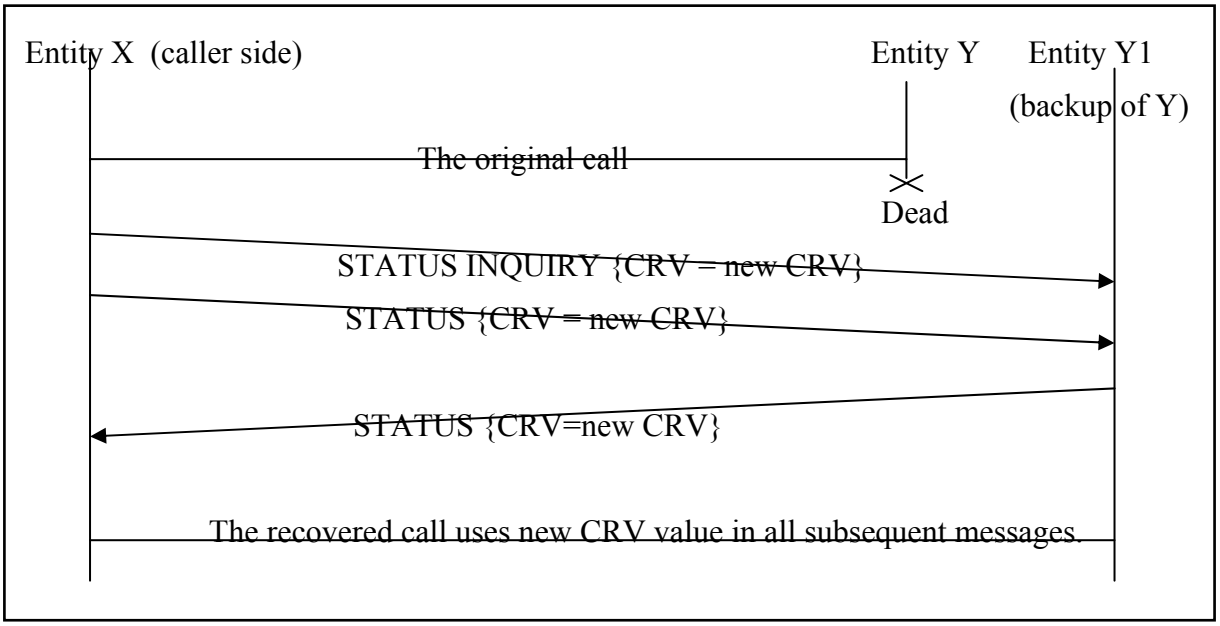


Figure R.3/H.323: Failed entity is of Method A and called side

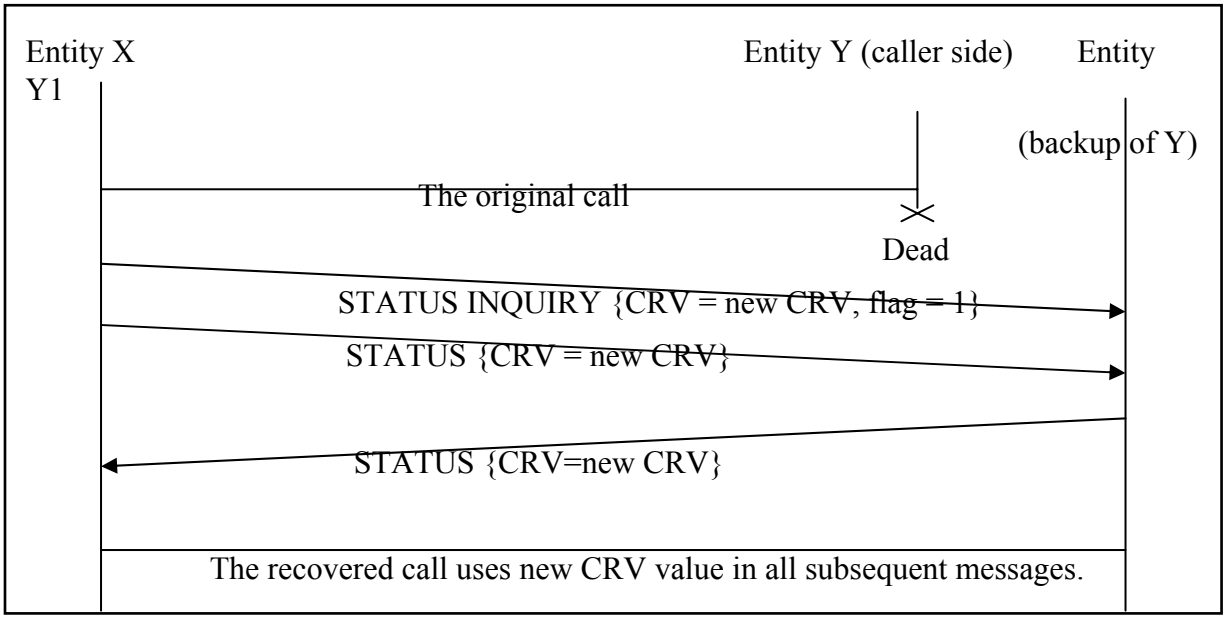


Figure R.4/H.323: Failed entity is of Method A and caller side

If the failed entity implements Method B, the signalling neighbour or the backup entity of the failed can re-establish the call signalling connection. Whoever re-establishes the call signalling connection, before sending any Q.931 messages, the entity shall check if it is the one that originated (caller side of) the call and if it already has prior calls to the recovered entity. If the entity that is re-establishing the connection is on caller side of the call and it has prior calls to the signalling neighbour as shown in Figure R.4**,** then the entity shall assign a new CRV value for this call and use it in all subsequent Q.931 and RAS messages. The signalling neighbour shall assign a unique CRV value for this call and use it in RAS messages in its communication with the gatekeeper. If the entity that is re-establishing the connection is on called side of the call and it has prior calls from the signalling neighbour entity as shown in Figure R.6, then the entity shall assign a new unique CRV value and use it in Q.931 message with CRV flag = 1 because it is destination side of the call. The signalling neighbour entity shall adopt this new CRV for this call. All the subsequent Q.931 messages for this call shall use this new CRV value. The signalling neighbour entity, if required, shall assign a unique CRV value for this call to be used in RAS messages.

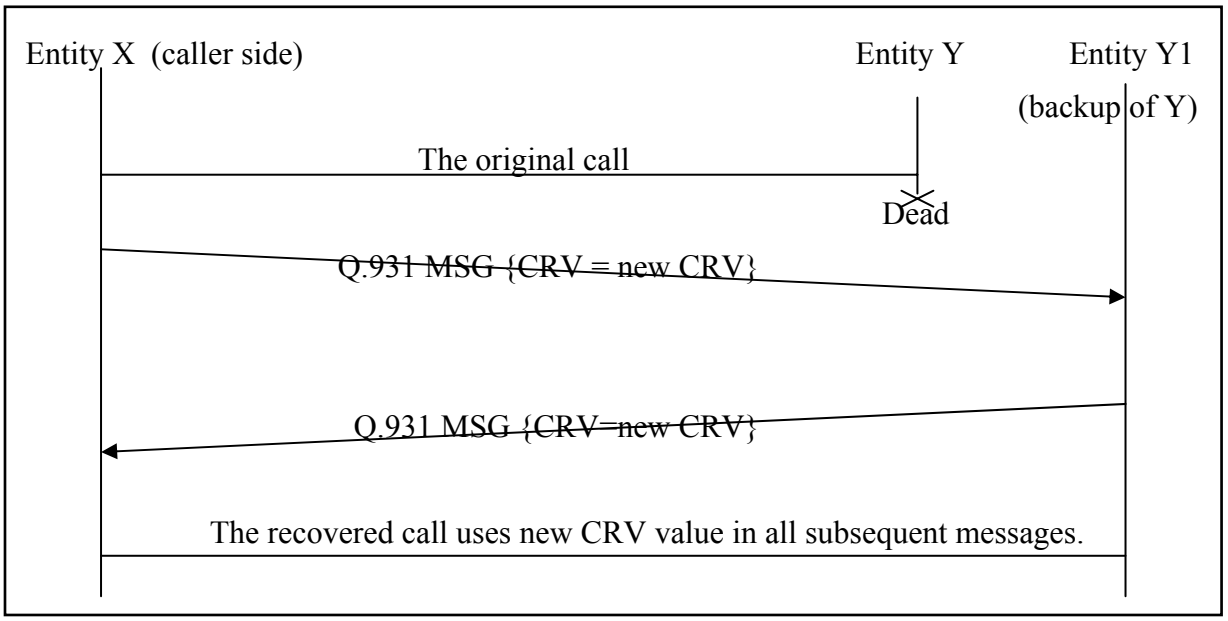
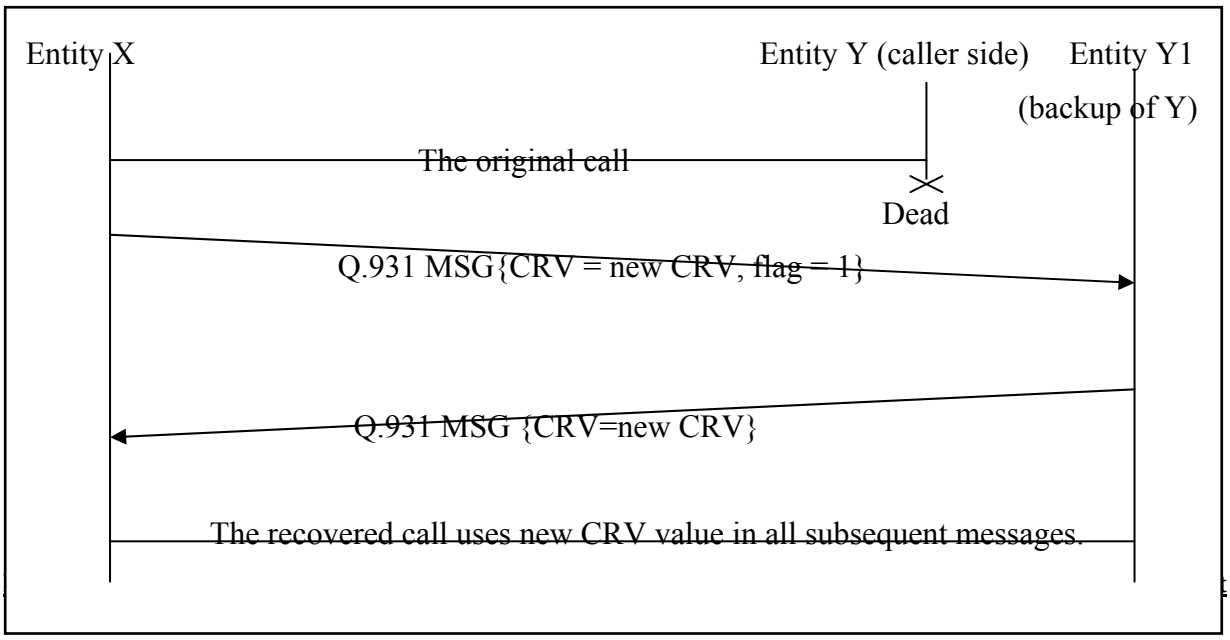


Figure R.5/H.323: Failed entity is of Method B and called side and Survived entity initiates re-establishment



[End Correction]

6.12.5 Robustness Data Definition

Description:	Based on the suggested changes and to make the signalling more transparent, the robustness generic data definition needs to be revised. The following is the new definition. The table defining the usage of fields per message is also not required with this restructuring.

[Begin Correction]

R.11 GenericData usage

The data fields necessary to implement this annex's features are carried in GenericData fields of various messages as defined below. `RobustnessData` shall be encoded and the resulting binary data carried as a raw instance of GenericData in the specified messages.

```
RobustnessData ::= SEQUENCE
{
    backupCallSignalAddresses   SEQUENCE OF TransportAddress,
                                -- empty when not required
    h245Address                 TransportAddress OPTIONAL,
    fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
    timeToLive                  TimeToLive OPTIONAL
    hasSharedRepository         NULL OPTIONAL,
    includeFastStart            NULL OPTIONAL,
    ...
}

BackupCallSignalAddresses ::= SEQUENCE OF CHOICE {
    tcp                 TransportAddress,
    alternateTransport AlternateTransportAddresses
    ...
}


RobustnessData ::= SEQUENCE
{
    versionID           INTEGER (1..256),
    robustnessData      CHOICE {
        rrqData             Rrq-RD,
        rcfData             Rcf-RD,
        setupData           Setup-RD,
        connectData         Connect-RD,
        statusData          Status-RD,
        statusInquiryData   StatusInquiry-RD,
        ...
    },
    ...
}

Rrq-RD ::= SEQUENCE
{
    backupCallSignalAddresses   BackupCallSignalAddresses,
    hasSharedRepository         NULL OPTIONAL,
    ...
}

Rcf-RD ::= SEQUENCE
{
    hasSharedRepository     NULL OPTIONAL,
```

```
    ...,
    irrFrequency            INTEGER (1..65535) OPTIONAL     -- in seconds;
                                                            -- not present
                                                            -- if GK does not
                                                            -- want IRRs for
                                                            -- recovered calls
}

Setup-RD ::= SEQUENCE
{
    backupCallSignalAddresses   BackupCallSignalAddresses,
    hasSharedRepository         NULL OPTIONAL,
    endpointGuid                GloballyUniqueIdentifier OPTIONAL,
    ...
}

Connect-RD ::= SEQUENCE
{
    backupCallSignalAddresses   BackupCallSignalAddresses,
    hasSharedRepository         NULL OPTIONAL,
    endpointGuid                GloballyUniqueIdentifier OPTIONAL,
    ...
}

Status-RD ::= SEQUENCE
{
    h245Address    TransportAddress OPTIONAL,
    fastStart      SEQUENCE OF OCTET STRING OPTIONAL,
    resetH245      NULL OPTIONAL,
    ...
}

StatusInquiry-RD ::= SEQUENCE
{
    h245Address         TransportAddress OPTIONAL,
    timeToLive          TimeToLive OPTIONAL,
    includeFastStart    NULL OPTIONAL,
    ...
}
```

…

R.11.1 GenericData usage in H.225.0 messages

~~Entities supporting robustness shall use GenericData related fields as follows (see Table R.1):~~

~~Table R.1/H.323 – Usage of GenericData fields for robustness data~~

~~Message~~	~~include RobustnessData in GenericData~~	~~Required fields~~						~~robustness FeatureDescr in desiredFeatures of featureSet~~
		~~hasShared Repository~~	~~backupCallSig Addresses~~	~~robustness FastStart~~	~~include FastStart~~	~~robustness TimeToLive~~	~~robustness H245Addr~~	
~~RRQ~~	~~M~~	~~M~~						~~M~~
~~RCF~~	~~M~~	~~M~~						~~M~~
~~ARQ~~								~~M~~
~~ACF~~								~~M~~
~~Setup~~	~~M~~	~~M~~	~~M~~					~~M#~~

~~Connect~~	~~M~~	~~M~~	~~M~~					~~M~~
~~Status+~~	~~M~~			~~M~~			~~M~~	
~~StatusInquiry+~~	~~M~~				~~M~~	~~M~~	~~M~~	
~~M – mandatory – all others forbidden.~~ ~~+ – when used for robustness procedures.~~ ~~# – desiredFeatures is not inside featureSet in Setup.~~								

~~All entities supporting robustness procedures shall support Status with the added RobustnessData field to enhance interoperability between the method A and B.~~

RRQ, RCF, ARQ, ACF, Setup, Connect, Status, and StatusInquiry shall include RobustnessData in GenericData as per the data definitions for the respective messages.

All messages (RRQ, RCF, ARQ, ACF, Setup, and Connect) excluding the Status and StatusInquiry shall include the robustness FeatureDescr in desiredFeatures of featureSet. Note that the desiredFeatures is not inside featureSet in Setup.

The version of this data (versionID field in RobustnessData) shall be set to 1.

[End Correction]

6.12.6 Indication of Non-existent Call in STATUS

Description:	The following text should be inserted to clarify that how the receiving entity should respond to a STATUS INQUIRY message for a call that it does not recognize.

[Begin Correction]

R.7.3 The robustness procedure

After a failure the H.323 Entity shall re-establish the Call Signalling connection and shall send both STATUS INQUIRY and STATUS messages to the other H.323 Entity. The other H.323 Entity shall respond with a STATUS messages, thus reaching a state where both sides are aware of the Call State of the other side. If the receiving entity is unaware of the call, it shall respond with a STATUS message with CallState IE set to NULL. The Call Signalling connection should be established to one of the entries in **backupCallSignalAddresses** in the order of preference defined by the order of elements in **backupCallSignalAddresses** structure.

[End Correction]

6.12.7 Terminal capabilities re-negotiation

Description:	The manner in which recovering entities transmit their own capabilities learn about the capabilities of the remote endpoint is left unspecified. The following text should be added to provide some clarification in this regard. Corrections are suggested for procedures for both Method A (Section R.7.3) and for Method B (Section R.8.4) entities.

[Begin Correction]

R.7.3 The robustness procedure

…

Both the recovering H.323 entity and its signalling neighbour shall implicitly reset their H.245 state machines for the call as the recovering entity is not aware of any remote terminal capabilities or the knowledge of the result of MSD negotiations. Moverover, the recovering entity's capabilities may differ from the failed entity. Before any H.245 messages are sent, both entities need to exchange TCS messages and determine Master/Slave.

[End Correction]

[Begin Correction]

R.8.4 H.245 connection re-establishment

After the Call Signalling channel has been re-established and the robustness procedure has reached a stable state, if H.245 tunnelling was in use, the entities can continue tunnelling H.245 messages using the new Call Signalling channel.

If a separate H.245 connection was being used it may have also failed alone or along with the Call Signalling channel. If the entity has detected failure on an H.245 channel, it shall drop its connection without closing it (not sending EndSessionCommand, which would indicate to the other end that the call was over). It shall then attempt to establish a new connection by sending its h245Address in a Facility message to its signalling neighbour. An entity receiving Facility with an h245Address for a call for which it already has an H.245 channel (possibly failed but not detected) shall close that existing channel and open the new one. Neither entity shall perform H.245 initialization procedures (master slave determination and terminal capability exchange) for the new channel.

The recovering entity may have different set of capabilities than the failed entity. In this case and especially when H.245 procedures were started between the signalling neighbours, the entities should restart their H.245 state machines and begin anew. This is done by using the **resetH245** flag in the STATUS robustness-data. After transmission of this flag, the entities should follow it up by exchanging TCS and MSD messages.

[End Correction]

6.13 Technical and Editorial Corrections to ITU-T Recommendation H.530 (2002)

6.13.1 Protection against replay attacks

Description:	A security weakness has been identified within H.530. The weakness relates to the observation that the V-GK cannot verify the received AuthenticationConfirmation message as fresh; thus replay or masquerade attacks become feasible. The weakness can be removed by introducing additional security parameters in the key management message.

8.2 – Secure location updating procedures

[Begin Correction]

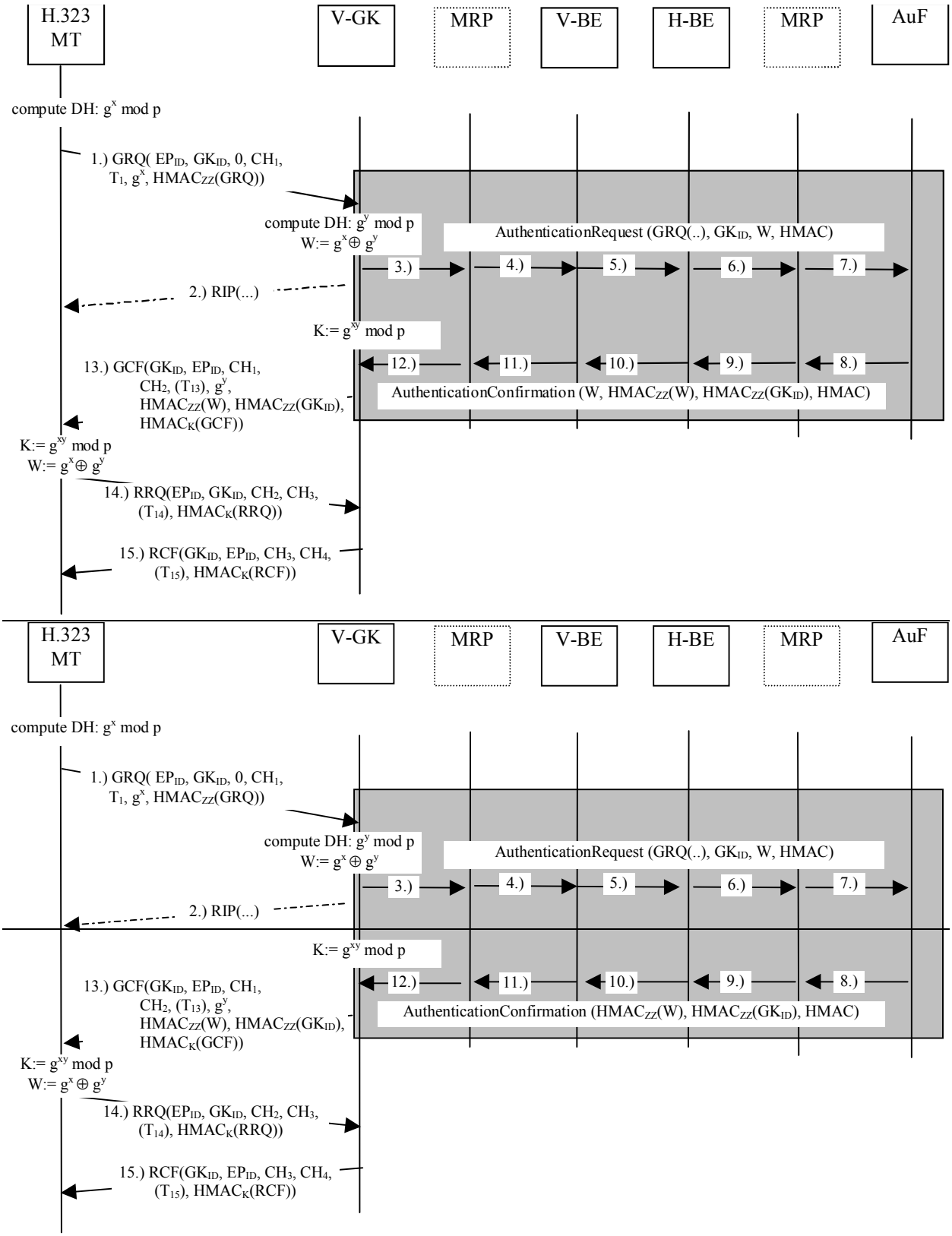


Figure 2/H.530: Authentication and key management during GK discovery phase

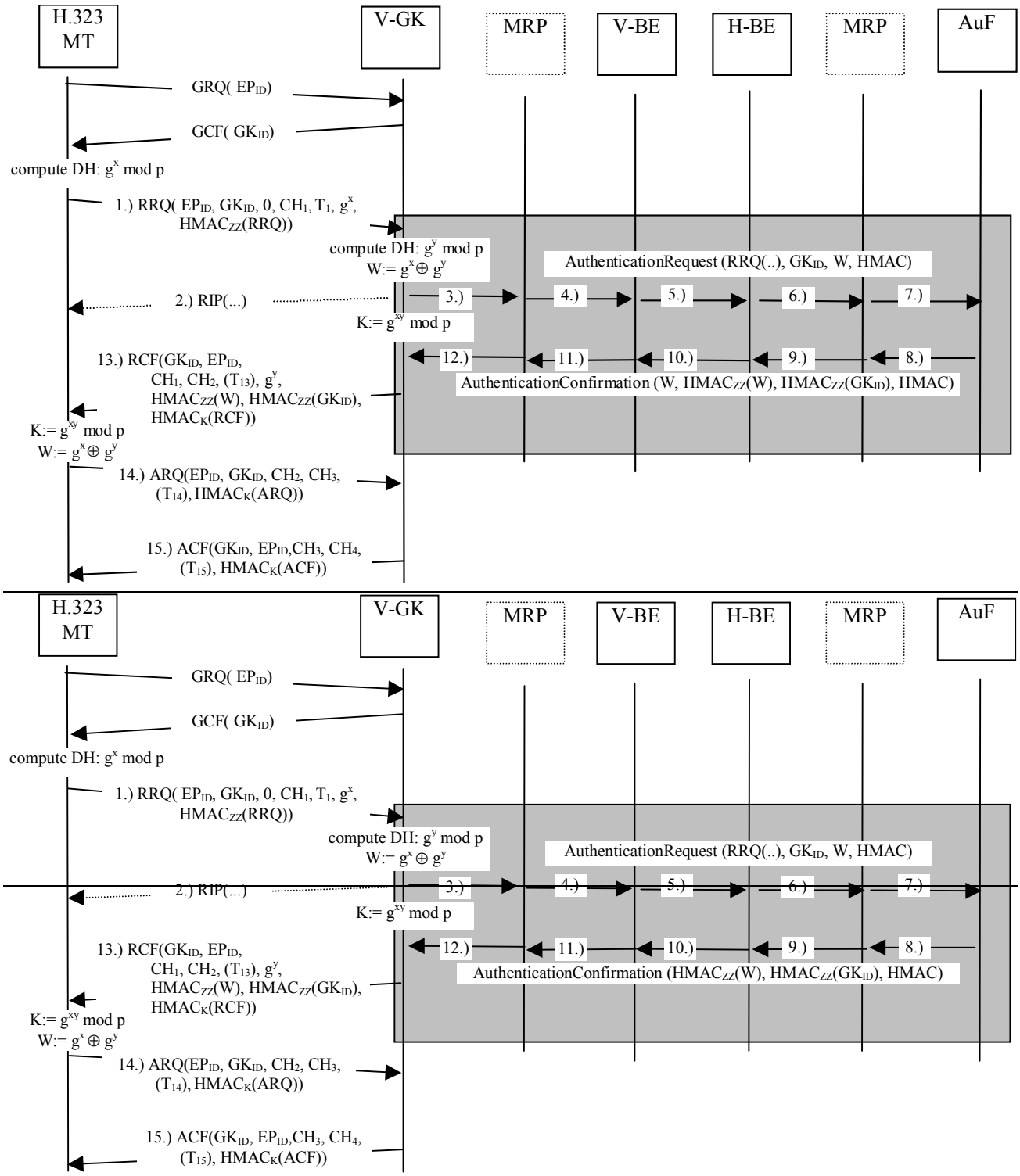


Figure 3/H.530: Authentication and key management during registration phase

[End Correction]

8.2.1 - MT to V-GK

[Begin Correction]

The V-GK receives an **AuthenticationConfirmation/AuthenticationRejection** with the result of the authentication and authorization check by the AuF and conveyed credentials,

see message 12.). The V-GK shall verify that the conveyed mobility **ClearToken** holds the same value *W* as was sent in message 3). A mismatch indicates a replay attack; in this case the V-GK shall consider the MT authentication by the AuF as failed and respond with **GRJ**/**RRJ** indicating the **reason** according to H.235 B.2.2 [4].

The V-GK may supervise reception of **AuthenticationConformation**/**AuthenticationRejection** messages using a timer. The timer duration should be chosen long enough by taking the network transit and the AuF processing into account. If the timer expires and the corresponding reply from the AuF has not arrived, the V-GK shall send an unprotected **RCF**.

The V-GK shall generate a new challenge CH_2 and build **RCF**. The **RCF** shall convey the previous challenge CH_1 within **password**, a new challenge CH_2 within **challenge** within the **ClearToken** inside the **CryptoToken** of **RCF**. That **ClearToken** shall also convey the computed Diffie-Hellman half-key of the V-GK in the **halfkey** field of the **dhkey** field within the **ClearToken** of that message. The applied prime number shall be included in **modsize** while the DH-generator shall be included in **generator** of that **ClearToken**.

Further, the V-GK shall forward the credentials from the AuF to the MT. The credentials encompass the mobility **ClearToken** shown as **WT().** This mobility **ClearToken** conveys on one hand the authenticated compound value *W* in the **halfkey** field of the **dhkey** field and on the other hand the authenticated V-GK ID; the value *W* should not be part of forwarded **WT()**. The **tokenOID** shall be set to "G2" and any other parameters in that mobility **ClearToken** shall be unused.

[End Correction]

8.2.2 - V-GK to MRP

[Begin Correction]

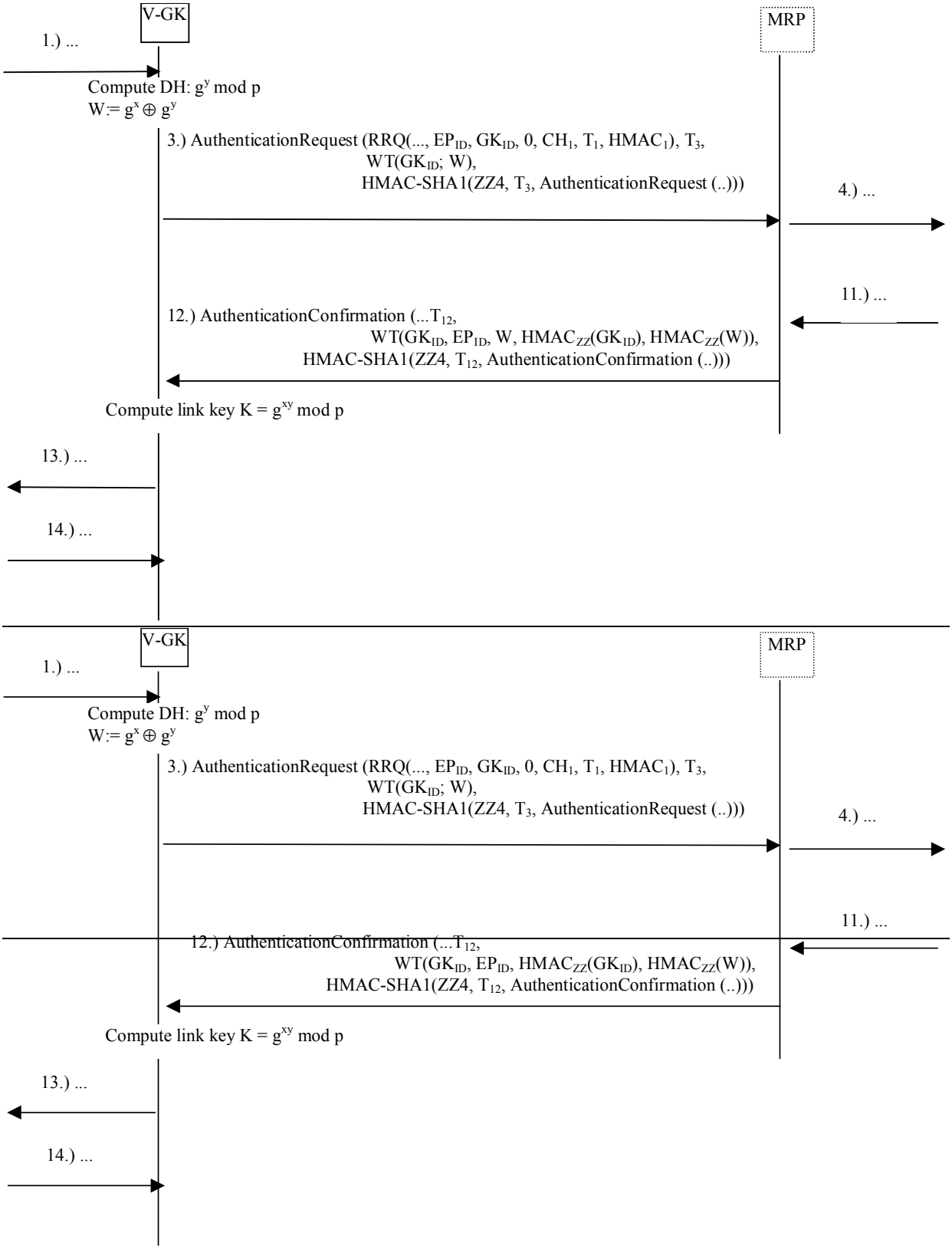


Figure 5/H.530: Transmission of authentication information between V-GK and MRP

[End Correction]

8.2.3 - MRP to V-BE

[Begin Correction]

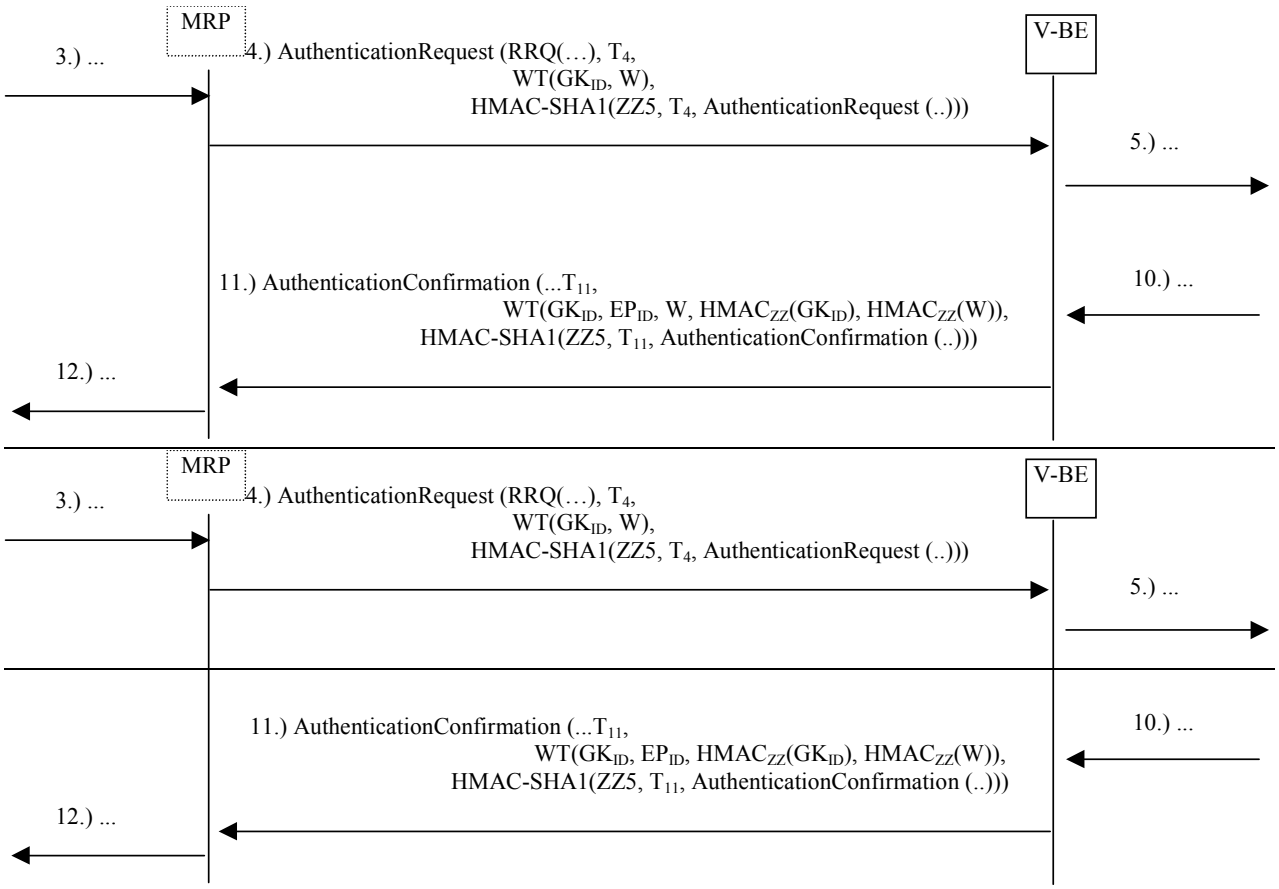


Figure 6/H.530: Transmission of authentication information between MRP and V-BE

[End Correction]

8.2.4 - V-BE to H-BE

[Begin Correction]

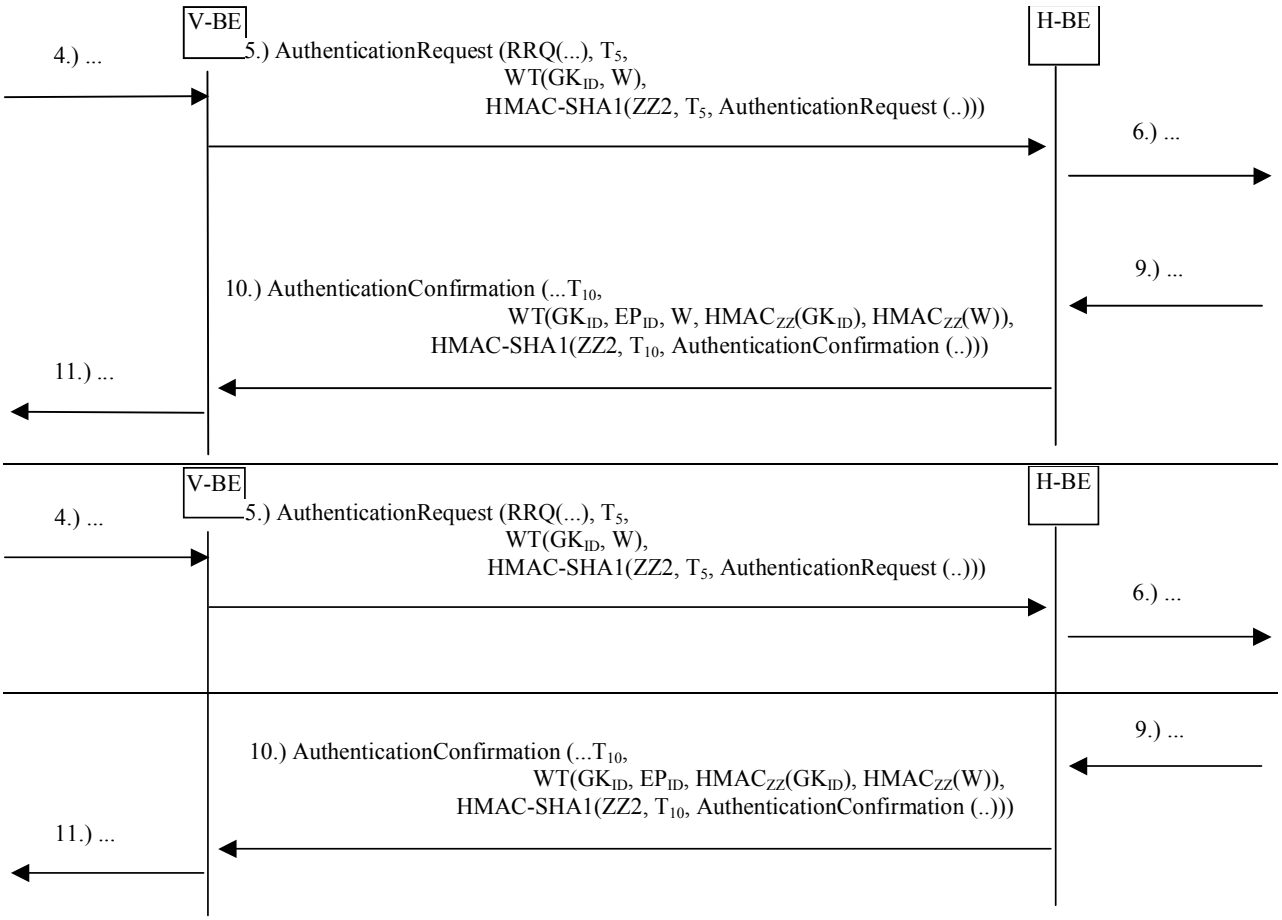


Figure 7/H.530: Transmission of authentication information between BEs

[End Correction]

8.2.5 - H-BE to MRP

[Begin Correction]

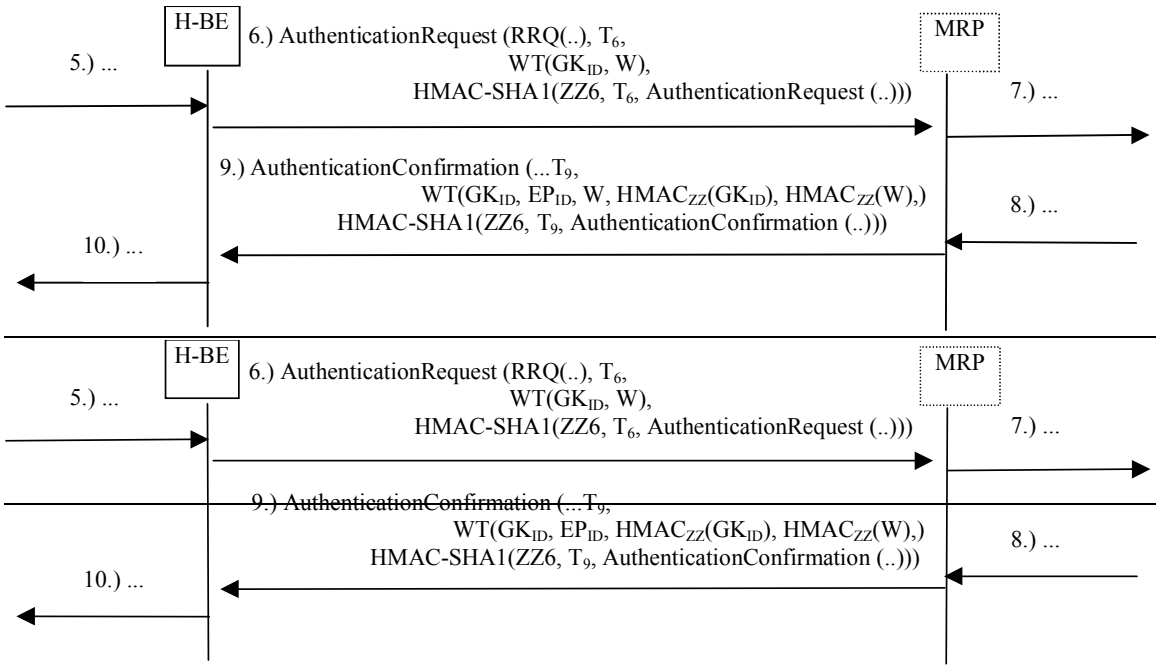


Figure 8/H.530: Transmission of authentication information between H-BE and MRP

[End Correction]

8.2.6 - MRP to AuF

[Begin Correction]

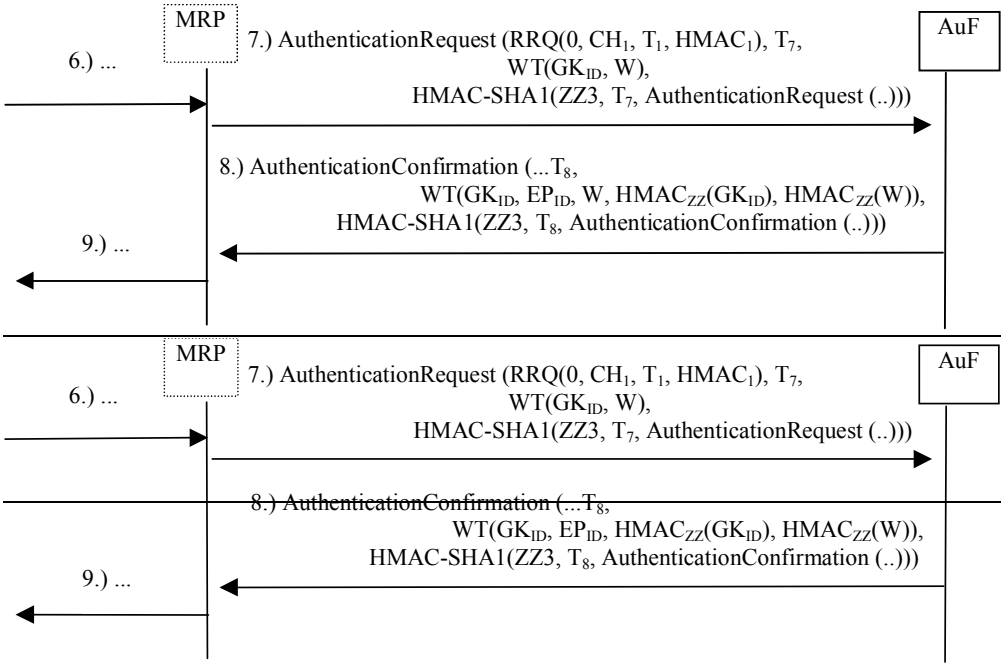


Figure 9/H.530: Transmission of authentication information between MRP and AuF

Otherwise, the AuF shall also compute the credentials of the authenticated compound value *W* using HMAC-SHA1-96 key hash function and *ZZ* as the shared key. The authenticated

compound value *W* shall be included in a separate mobility **ClearToken**, where the result is stored in the **halfkey** field of the **dhkey** field within that mobility **ClearToken**. Further, the AuF shall compute an authenticated GK_{ID} as another credential using HMAC-SHA1-96 key hash function and *ZZ* as the shared key. The result shall be included within **generator** in that **ClearToken**. The AuF shall also include *W* in the **modsize** field of **dhkey**; this allows the V-GK to recognize **AuthenticationConfirmation**/ **AuthenticationRejection** as fresh. The **generalID** shall convey the GK_{ID}, while the **sendersID** shall convey the EP_{ID} in that **ClearToken**; this shall allow the V-GK to associate an **AuthenticationConfirmation**/ **AuthenticationRejection** with the corresponding **AuthenticationRequest** message. The **tokenOID** of that **ClearToken** shall be set to "G2" and any other parameters in that mobility **ClearToken** shall be unused. The mobility **ClearToken** is shown as **WT()**.

[End Correction]

8.5 - Application of the Symmetric Security Protocol in the Home Domain

[Begin Correction]

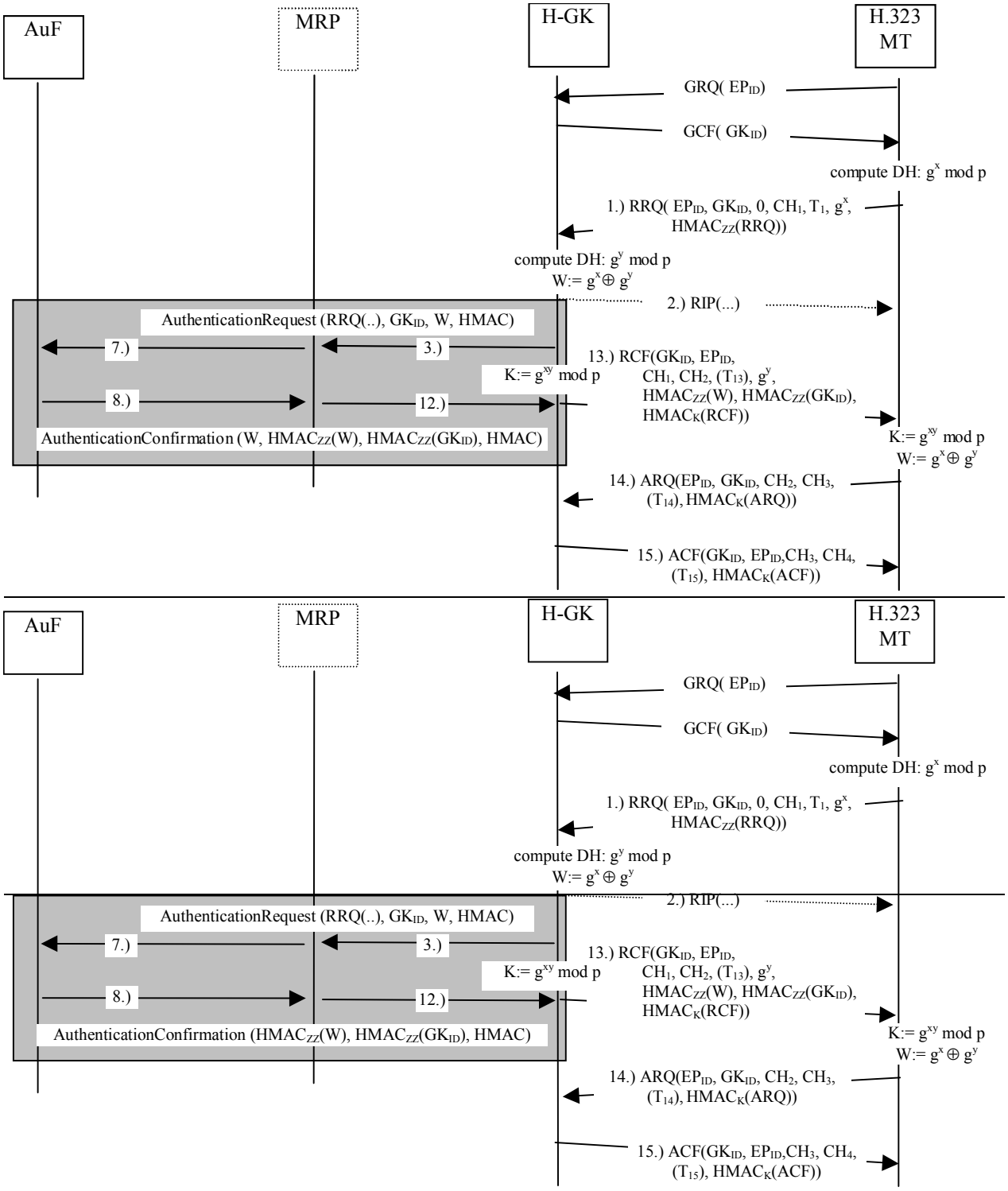


Figure 11/H.530: MT authentication in the home domain during registration phase

[End Correction]

6.14 Technical and Editorial Corrections to ITU-T Recommendation H.460.6 (2002)

6.14.1 Close All Channels

Description:	The intent of the Close All Media Channels request described in section

	4.1.2 is to close all open media channels and cancel all available sessions, as described in section 4.5. To this end, text in sections 4.1.2 and 4.5.2 should be changed as follows.

[Begin Correction]

4.1.2 Close All Channels

This parameter may be used by a party to request that the receiving endpoint close all open media channels and cancel all available sessions. Support for this parameter is optional, and shall be negotiated during EFC feature negotiation.

…

4.5.2 Requesting Close-All-Channels

An endpoint or a third party may request that the other endpoint close all open media channels and cancel all available sessions by sending a **genericData** element with the EFC featureID and parameter 2 present in any convenient call signalling message (e.g., FACILITY). The receiving endpoint is expected to silently close all open channels without any response (e.g., without issuing any **Null-OLC**s.)

[End Correction]

6.14.2 Signaling of EFC Support in supportedFeatures

Description:	It is held that signalling of EFC in **supportedFeatures** by the originating party is unnecessary. The text in section 4.2 should be corrected as below.

[Begin Correction]

4.2 Invocation of Extended Fast Start

An originating party shall indicate its desire to use EFC when it issues a SETUP message. The SETUP shall contain a request for EFC support in the **desiredFeatures** element, or a requirement for EFC support in the **neededFeatures** element. ~~The **supportedFeatures** element shall indicate support for EFC as well.~~ The EFC feature is symmetric, hence requestor support for the feature may be inferred from a request for EFC, and the **supportedFeatures** element need not be included to indicate support for EFC. In addition, the SETUP message shall include a **genericData** element specifying EFC Proposal (parameter 1) and a **fastStart** element containing one or more proposals. That is, EFC procedures shall include the standard Fast Connect procedures.

[End Correction]

6.14.3 Prevention of Race Condition in Master/Slave Determination

Description:	There is a possible race condition that may occur, depending on the order in which an endpoint processes **fastStart** elements versus tunnelled H.245 master/slave negotiation messages embedded in the same H.225.0 message. Thus, it is suggested that the following paragraph be added to the end of section 4.2.1.

[Begin Correction]

4.2.1 Master/Slave Determination

Parties supporting Extended Fast Connect should use the H.245 tunnel to carry out master/slave negotiation. For the initial Fast Connect exchange, the caller (sender of the SETUP with proposals) shall be considered the slave, and the called party (acceptor of proposals) shall act as the master. Although this convention will suffice for simple A-to-B calls, it can lead to complications in more complex call scenarios.

Different implementations may process **fastStart** elements and tunnelled H.245 messages in different orders. EFC proposals or acceptances shall not be included in any H.225.0 message that carries an H.245 **MasterSlaveDeterminationAck** message that conveys a change in master/slave status. Doing so could lead to temporary confusion about which party is master and how to respond to the EFC elements.

[End Correction]

6.14.4 Remote Endpoint Type and Version after Re-routing

Description:	An endpoint may not be aware of the H.323 protocol version number supported by the remote endpoint, especially if the call gets re-routed one or more times. In some cases it might be helpful for the endpoint to have this information. The following additions should be made to H.460.6 document.

[Begin Correction]

5.5 EFC Third-party Pause and Rerouting

EFC supports third-party pause and rerouting, as described in H.323 Annex F for SETs, when used by a routing gatekeeper. The third party (the gatekeeper in the example in Figure 5) may idle the caller's transmit and/or receive channels via **Null-OLC**s, then supply the caller's proposal **fastStart** to a new party (e.g., in a SETUP). The acceptance **fastStart** will appear to the caller as a redirection or reconfiguration, asillustrated in Figure 5.

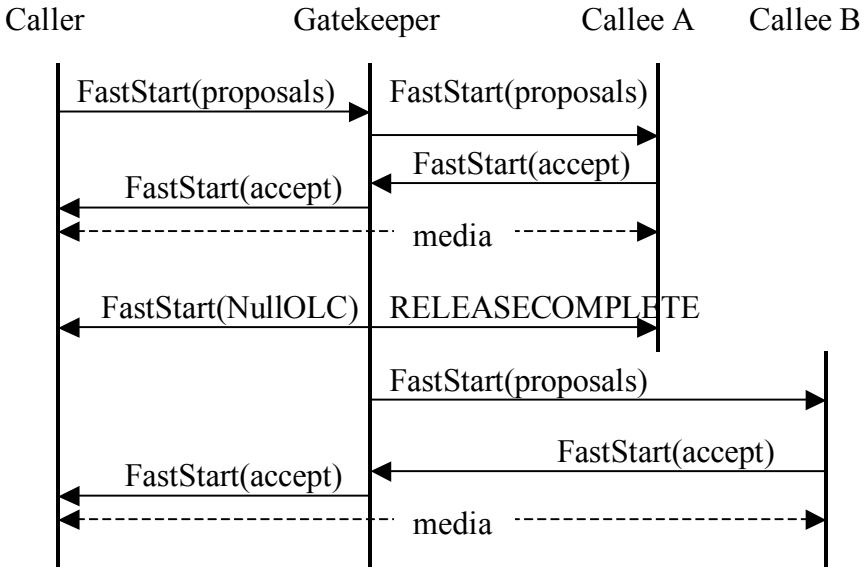


Figure 5/H.460.6 - Third-Party Redirection

In the above diagram, the Gatekeeper, or the entity that re-routes a call should send a Facility message containing the **destinationInfo** field upon completion of the re-routing to the entity that gets re-routed, i.e., Caller. An endpoint should examine this message for the H.225.0 version information at any point that a Facility message is received containing this field.

After coming out of the "paused" state an endpoint should examine the version-id fields in TCS messages to determine the H.245 version supported by the remote endpoint.

In addition, an endpoint interested in knowing the version of the remote endpoint should send a Status Inquiry message and wait for the receipt of the Status message to determine the version of the H.225.0 in use when it exits the paused state when the above Facility message is not received within a reasonable amount of time. The length of this time is left to the implementation.

[End Correction]

7 Implementation Clarifications

7.1 Token Usage in H.323 Systems

There has been some confusion on the usage of individual **CryptoH323Tokens** as passed in RAS messages. There are two main categories of **CryptoH323Tokens**; those used for H.235 procedures and those used in an application specific manner. The use of these tokens should be according to the following rules:

- All H.235 defined (e.g. **cryptoEPPwdHash**, **cryptoGKPwdHash**, **cryptoEPPwdEncr**, **cryptoGKPwdEncr**, **cryptoGKCert**, and **cryptoFastStart**). shall be utilized with the procedures and algorithms as described in H.235.
- Application specific or proprietary use of tokens shall utilize the **nestedcryptoToken** for their exchanges.
- Any **nestedcryptoToken** used should have a **tokenOID** (object identifier) which unambiguously identifies it.

7.2 H.235 Random Value Usage in H.323 Systems

The random value that is passed in xRQ/xCF sequence between endpoints and Gatekeepers may be updated by the Gatekeeper. As described in section 4.2 of H.235 this random value may be refreshed in any xCF message to be utilized by a subsequent xRQ messages from the endpoint. Due to the fact that RAS messages may be lost (including xCF/xRJ) the updated random value may also be lost. The recovery from this situation may be the reinitializing of the security context but is left to local implementation.

Implementations that require the use of multiple outstanding RAS requests will be limited by the updating of the random values used in any authentication. If the updating of this value occurs on every response to a request, parallel requests are not possible. One possible solution, is to have a logical "window" during which a random value remains constant. This issue is a local implementation matter.

7.3 Gateway Resource Availability Messages

The Resources Available Indication (RAI) is a notification from a gateway to a gatekeeper of its current call capacity for each H-series protocol and data rate for that protocol. The gatekeeper responds with a Resources Available Confirmation (RAC) upon receiving a RAI to acknowledge its

reception. A Gatekeeper should ignore any RAI notifications (e.g. send no RAC) upon receiving a RAI which contains bogus information (i.e. a bad endpointIdentifier).

7.4 OpenLogicalChannel in fastStart

In the H.225.0 ASN.1, **fastStart** is defined as SEQUENCE OF OCTET STRING OPTIONAL. The text definition states "This uses the **OpenLogicalChannel** structure defined in H.245…" Each OCTET STRING in **fastStart** is to contain the **OpenLogicalChannel** structure, not an entire request message.

7.5 Clarification in Q.931 (1993)

Table 4-3/Q.931 (1993) (Information Element Identifier Coding) shows that the Progress Indicator IE identifier is 0x1e, but Figure 4-29/Q.931 (octet layout of Progress Indicator IE) shows the identifier as 0x1f. Note that the identifier should be 0x1e.

7.6 Graceful Closure of TCP Connections

When a TCP connection is closed, the graceful closure procedure documented in section 3.5 of RFC 793 should always be used.

7.7 Race Condition on Simultaneous Close of Channels

Section 8.5 of H.323 describes the procedures that an endpoint follows to terminate a call. It should be noted that as prescribed in Step 6, both endpoints shall issue a Release Complete simultaneously. Endpoints should be prepared for this potential race condition.

7.8 Acceptance of Fast Connect

When an endpoint accepts the Fast Connect procedure, it may select from the proposed channels as specified in section 8.1.7.1/H.323. The Recommendation clearly specifies what fields shall be modified by the endpoint to accept both the forward and the reverse channels. An endpoint shall not modify any fields other than those specified in 8.1.7.1/H.323 when returning the proposed channels.

Newer versions of H.245 may introduce new fields into the **OpenLogicalChannel** sequence or one of the structures contained therein, as well as new procedures. An older endpoint is obviously not required to decode such new fields or to return such new fields when accepting any proposal. Implementers should consider the consequences of transmitting a newer H.245 OLC to an older endpoint. For the purposes of Fast Connect, the calling endpoint shall assume that the called endpoint's version of H.245 is the minimum version of H.245 necessary to be complaint with an H.323 device that advertises the version of H.225.0 transmitted in the messages from the called endpoint (refer to the "Summary" section of H.323).

7.9 Semantic Differences between Lightweight RRQs and IRQ/IRR Messages

The lightweight RRQ and the IRR message serve two different functions with an H.323 system. While both are a means of allowing the Gatekeeper to discover that an endpoint is alive, they also each serve separate, unique functions.

The lightweight RRQ is intended to prevent a registration with a Gatekeeper from expiring. The message is generated by the endpoint and does not require the Gatekeeper to poll each endpoint on a regular interval. This message is also a means of allowing the Gatekeeper to provide updated registration information, such as a new list of Alternate Gatekeepers, after the initial registration.

Version 1 of H.323 did not have the concept of a lightweight RRQ, so the IRQ/IRR exchange is the only mechanism available to determine endpoint status of Version 1 devices. However, the

lightweight RRQ may be a better choice for determining endpoint status for Version 2 and higher devices.

The IRQ/IRR exchange allows the Gatekeeper to poll the endpoint periodically to discover if the endpoint is still alive. However, an IRR is also intended to convey details about current active calls. This can be used by the Gatekeeper to discover calls that have terminated, which may happen if the endpoint fails to properly send a DRQ message for a call. The IRR message also provides specific details about active calls.

7.10 Specifying the Payload Format for a Channel

Implementers should be conscientious of the fact that there are possibly multiple payload formats defined for media formats. For example, two payload formats are defined for H.263—one is defined for the Recommendation H.263 (1996) and one for Recommendation H.263 (1998). Other payload formats may be defined for existing codecs or revisions of those codecs. For interoperability, it is strongly advised that implementers provide the **mediaPacketization** element of the **h2250LogicalChannelParameters** sequence in the **OpenLogicalChannel** message so that there is no ambiguity at to which payload format is being used.

7.11 Version Dependencies in Annexes

It was noted that the Annexes to H.323 often fail to indicate the minimum version of H.323 and H.245 required for the Annex. This table is an attempt to clarify the version relationships:

H.323 Annex	*Minimum H.323 Version*	*Minimum H.245 Version*
Annex Dv1 (1998)	1998 (Version 2)	1998 (Version 4)
Annex Dv2 (2000)	2000 (Version 4)	2000 (Version 7)
Annex E	1998 (Version 2)	N/A
Annex F	1998 (Version 2)	N/A
Annex G	1998 (Version 2)	1998 (Version 4)
Annex J	1998 (Version 2)	N/A
Annex M.1	2000 (Version 4)	N/A
Annex M.2	2000 (Version 4)	N/A
Annex P	2000 (Version 4)	2003 (Version 9)
Annex R	2000 (Version 4)	N/A

7.12 Routing through Signaling Entities and Detecting Loops

In some call scenarios, a call may be routed though a signaling entity multiple times. For example, a call from Endpoint 1 (EP1) may be routed through Gatekeeper 1 (GK1) and Gatekeeper 2 (GK2) to Endpoint 2 (EP2) as shown in the Figure 1.

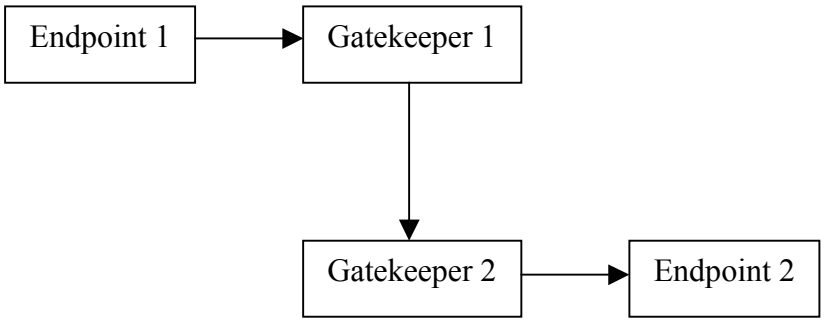


Figure 1 - Call placed through multiple gatekeepers

If EP2 redirects the call to a third endpoint, such as Endpoint 3 (EP3), signaling entities such as GK1 and GK2 should be prepared to handle such call rerouting. For this example, assume that EP2 returned a Facility message with a **reason** of **callForwarded** upon receiving a Setup message. Rather than propagate that response back to EP1, GK2 may choose to handle the call forward operation. GK2 would send a Release Complete to EP2 and begin rerouting the call. Suppose that GK2 sends an LRQ message to GK1 for EP3 and that GK1 replies with its address so that that calls routed to EP3 are routed through it. GK2 would then send a Setup message for this call to GK1 as shown in Figure 2.

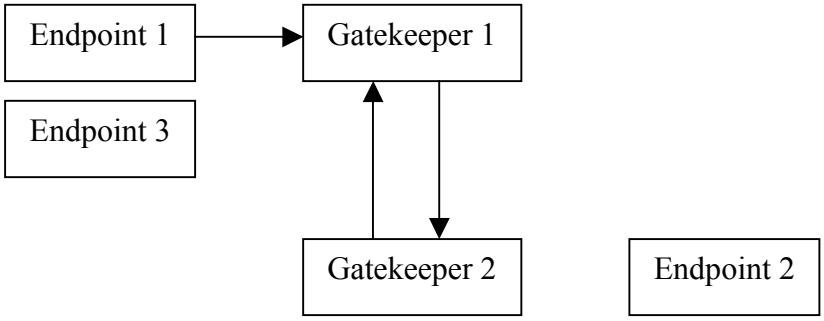


Figure 2 - Gatekeeper 2 re-routes call back to Gatekeeper 1

When GK1 receives the Setup message from GK2, it may inadvertently mistake the call as "bogus", since the Call Identifier will match an already existing call within the Gatekeeper. Implementers should consider this type of call scenario and be prepared to receive incoming calls that contain Call Identifiers for calls that are already being routed through the routing entity. The routing entity should examine not only the Call Identifier, but also the destination address of the call (the call signaling address, aliases, or Called Party Number of the destination). In this case, the call is routed through GK1 with a destination address of EP2 is rerouted by GK2 to GK1, but with a destination address of EP3. In this way, the GK1 will properly handle call routing and rerouting, as well as prevent loops in the call signaling path.

In this example, there was a dependency on the H.323v2 Call Identifier. Unfortunately, H.323 version 1 systems did not have Call Identifiers. For this reason, these loop detection and rerouting procedures are not possible. Nonetheless, it is advisable for routing entities to make an effort to prevent loops properly. For example, if the entities in Figure 2 were version 1 devices, the GK1 may examine the source address, destination address, and Conference Identifier (CID) of the call. The first time the call is presented to the Gatekeeper, the destination address is EP2, just as before. However, when GK re-routes the call back to GK1, the destination address is EP3. In this way, GK1 may allow proper rerouting of the call to EP3.

The logic for Version 1 devices seems similar to that for Version 2 and higher devices, but there are issues when EP2 and EP3 are MCUs, for example. Suppose that EP2 is an MCU that is directing all calls to EP3. The first time a call is redirected to GK1, GK1 may realize that this is, indeed, a

call redirection as described above. However, when the second call is redirected, GK1 has no means of distinguishing between the first redirected call and the second: the source address *may* be the same, the destination address is the same as the previously rerouted call (EP3), and the Conference ID is the same. So in this case, GK1 may have no choice but to assume that a loop has occurred and release the offending call. Although this is unfortunate, H.323v2 and higher systems do not suffer from this problem. What is important, though, is that loop detection is possible—even with version 1 systems.

7.13 Packetization for G.729, G.729a, G.711, and G.723.1

The delay associated with codec processing and packetization should be kept as short as possible. To accomplish this objective when G.729 or G.729A is used, two frames per packet should be considered as the maximum packet size. Similarly, G.711 may be used with packet sizes of 10 ms (80 frames) or 20 ms (160 frames) to achieve this objective. Finally, when G.723.1 is used, only one frame should be included in each packet. The 30 ms frame size of G.723.1 results in speech collection and coding delay of at least 60 ms, contributing to difficulty of interactive communications.

8 Allocated Object Identifiers and Port Numbers

Information in this section is provided for informational purposes and convenience. This section does not supercede nor replace proper references in H.225.0, H.225, H.235, or other Recommendations.

8.1 Allocated Object Identifiers

The following object identifiers have been allocated for protocols associated with H.323. Any future object IDs that are allocated should be indexed here to prevent duplication.

Note that object IDs below that are allocated below the arc { itu-t(0) recommendation(0) } are show with an abbreviated prefix of "0 0" below.

{ 0 0 h(8) 2250 version(0) *[v]* }	H225.0 version numbers
Assigned values of *v*: 1-4	
{ 0 0 h(8) 2250 annex(1) g(7) version(0) *[v]* }	H225.0 Annex G version numbers
Assigned values of *v*: 1-2	
{ 0 0 h(8) 2250 annex(1) g(7) usage(1) *[u]* }	H225.0 Annex G usage tags
Assigned values of *u*: none	
{ 0 0 h(8) 245 version(0) *[v]* }	H245 version numbers
Assigned values of *v*: Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) video(0) *[c]* }	Generic video capabilities
Assigned values of *c*: Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) audio(1) *[c]* }	Generic audio capabilities
Assigned values of *c*: Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) data(2) *[c]* }	Generic data capabilities

Assigned values of *c*: Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) control(3) *[c]* }	Generic control capabilities
Assigned values of *c*: Please refer to Table D.1/H.245	
{ 0 0 h(8) 245 generic-capabilities(1) multiplex(4) *[c]* }	Generic multiplex capabilities
Assigned values of *c*: Please refer to Table D.1/H.245	
{ 0 0 h(8) 283 generic-capabilities(1) 0 }	H.283 Capability
{iso (1) identified-organization (3) icd-ecma (0012) private-isdn-signalling-domain (9)}	Identifies QSIG as the tunneled protocol within an H.225.0 Call Signalling Channel

8.2 Allocated Port Numbers

The following IP port numbers have been allocated for various components of H.323:

1300 TLS secured call signalling

1718 Multicast RAS Signalling

1719 Unicast RAS Signalling

1720 TCP call signalling

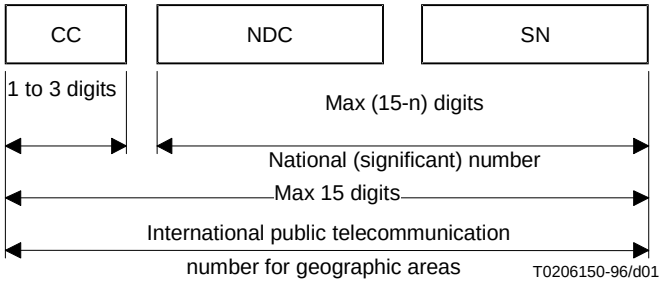
2099 Annex G/H.225.0 Signalling

2517 Annex E/H.323 Signalling

9 Use of E.164 and ISO/IEC 11571 Numbering Plans

9.1 E.164 Numbering plan

ITU-T Recommendation defines E.164 numbers the following way for geographic areas:



CC Country Code for geographic areas
NDC National Destination Code (optional)
SN Subscriber Number
n Number of digits in the country code

NOTE – National and international prefixes are not part of the international public telecommunication number for geographic areas.

Figure – International public telecommunication number structure for geographic areas

Similar descriptions are also defined for non-geographic areas. Recommendation E.164 further defines country codes (CC) for all the countries and regions of the world.

An international E.164 number always starts with a country code and its total length is always 15 digits or less. More importantly, it does not include any prefixes that are part of a dialing plan (for example, "011" for an international call placed in North America, or "1" for a long-distance call), nor does it include "#" or "*". The number "49 30 345 67 00" is an E.164 number with CC=49 for Germany. A national number is the international number stripped of the country code, "30 345 67 00" in this case. The subscriber number is the national number stripped of the national destination code, "345 67 00" in this case.

An E.164 number has global significance: any E.164 number can be reached from any location in the world. A "dialed digit sequence", however, only has significance within a specific domain. Within a typical private numbering plan in an enterprise, for example, a prefix, such as "9", may indicate that a call goes "outside", at which point the local telephone company's dialing plan takes over. Each telephone company or private network is free to choose its own dialing plan. It is also free to change it as it pleases—and frequently does so (adding new area codes, for example).

In a typical geographically determined network where users input telephone numbers manually and where users do not travel too much, having different dialing plans everywhere is usually a problem. However, when a user travels, the user must determine the other network's numbering plan in order to place calls. When computer systems perform the dialing automatically, the user is usually required to customize the dialing software for every region or network.

Because of these issues with varying dialing plans and automated dialing, it is essential to be able to refer to an absolute "telephone number" instead of "what you have to dial to reach it from a specific location." Proper usage of E.164 numbers can resolve these issues. Many systems use E.164 numbers instead of dialed digits: for example, a PBX may gather the dialed digits from a user on a telephone and then initiate a call to the local phone company using an E.164 number in the Called Party Number information element in Q.931. When completing the Called Party Number IE, specifying the numbering plan as "ISDN/telephony numbering plan (Recommendation E.164)" indicates an E.164 number. Specifying the type of number as "unknown" and the specifying the numbering plan as "unknown" indicates dialed digits.

The following are a set of definitions from E.164:

number

A string of decimal digits that uniquely indicates the public network termination point. The number contains the information necessary to route the call to this termination point.

A number can be in a format determined nationally or in an international format. The international format is known as the International Public Telecommunication Number which includes the country code and subsequent digits, but not the international prefix.

numbering plan

A numbering plan specifies the format and structure of the numbers used within that plan. It typically consists of decimal digits segmented into groups in order to identify specific elements used for identification, routing and charging capabilities, e.g. within E.164 to identify countries, national destinations, and subscribers.

A numbering plan does not include prefixes, suffixes, and additional information required to complete a call.

The national numbering plan is the national implementation of the E.164 numbering plan.

dialing plan

A string or combination of decimal digits, symbols, and additional information that define the method by which the numbering plan is used. A dialing plan includes the use of prefixes, suffixes, and additional information, supplemental to the numbering plan, required to complete the call.

address

A string or combination of decimal digits, symbols, and additional information which identifies the specific termination point(s) of a connection in a public network(s) or, where applicable, in interconnected private network(s).

prefix

A prefix is an indicator consisting of one or more digits, that allows the selection of different types of number formats, networks and/or service.

international prefix

A digit or combination of digits used to indicate that the number following is an International Public Telecommunication Number.

country code (CC) for geographic areas

The combination of one, two or three digits identifying a specific country, countries in an integrated numbering plan, or a specific geographic area.

national (significant) number [N(S)N]

That portion of the number that follows the country code for geographic areas. The national (significant) number consists of the National Destination Code (NDC) followed by the Subscriber Number (SN). The function and format of the N(S)N is nationally determined.

national destination code (NDC)

A nationally optional code field, within the E.164 number plan, which combined with the Subscriber's Number (SN) will constitute the national (significant) number of the international public telecommunication number for geographic areas. The NDC will have a network and/or trunk code selection function.

The NDC can be a decimal digit or a combination of decimal digits (not including any prefix) identifying a numbering area within a country (or group of countries included in one integrated numbering plan or a specific geographic area) and/or network/services.

national (trunk) prefix

A digit or combination of digits used by a calling subscriber, making a call to a subscriber in his own country but outside his own numbering area. It provides access to the automatic outgoing trunk equipment.

subscriber number (SN)

The number identifying a subscriber in a network or numbering area.

9.2 Private Network Number

Private Network Numbers are used in private or virtual private telephony networks, e.g., a corporate network of PBXs and virtual private lines.

ISO/IEC 11571 defines Private Network Number (PNP) as having up to three regional levels.

A PNP Number shall comprise a sequence of x decimal digits (0,1,2,3,4,5,6,7,8,9) with the possibility that different PNP Numbers within the same PNP can have different values of x. The maximum value of x shall be the same as for the public ISDN numbering plan, see ITU-T Recommendation E.164.

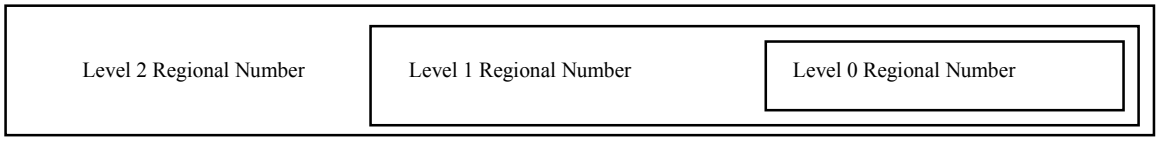


Figure – H.323 - Structure of a PNP Number with three levels of regions

A level n Regional Number (RN) shall have significance only within the level n region to which it applies. When that number is used outside that level n region, it shall be in the form of an RN of level greater than n. Only a Complete Number shall have significance throughout the entire PNP.

A typical example in North America would be a 4-digit "extension" as the Level 0 Regional Number: a 3-digit "location code" combined with the 4 digit "extension" would form the Level 1 Regional Number. The Level 2 Regional Number would be nil.

A prefix could also be used to signal which regional number is used, and would not be part of the regional number per se, but only part of the dialing plan. Again, a typical example would be the use of digit "6" to access a Level 1 Regional Number, and no digit for a Level 0 Regional Number.

The following are a set of definitions from ISO/IEC 11571:

Private Numbering Plan (PNP)

The numbering plan explicitly relating to a particular private numbering domain, defined by the PISN Administrator of that domain.

PNP Number

A number belonging to a PNP.

Region

The entire domain or a sub-domain of a PNP. A region does not necessarily correspond to a geographical area of a PISN.

Region Code (RC)

The leading digits of a PNP Number which identify a region. The RC may be omitted to yield a shortened form of a PNP Number for use internally to that region.

Regional Number (RN)

A particular form of a PNP Number which is unambiguous in the region concerned.

Complete Number

A number which is unambiguous in the entire PNP, i.e. which corresponds to the highest regional level employed in that PISN.

10 ASN.1 Usage, Guidelines, and Conventions

10.1 NULL, BOOLEAN, and NULL/BOOLEAN OPTIONAL

Throughout the ASN.1 used in H.323-series documents, the reader will see the types NULL and BOOLEAN used, along with the modifier OPTIONAL in some cases. People have questioned when NULL should be used or when BOOLEAN should be used and what the semantic differences are.

The BOOLEAN type allows a TRUE or FALSE value to be conveyed in the protocol. When used in conjunction with OPTIONAL, it actually allows three values to be conveyed through the protocol: TRUE, FALSE, and *absent*. The question is what does *absent* mean? In some instances, the absence of a BOOLEAN OPTIONAL means should be interpreted as FALSE, while in other

cases, it should be interpreted as "I don't care" or "I don't know"—but not always. For example, the **additiveRegistration** field in the RRQ of H.225.0 Version 4 is defined as a BOOLEAN OPTIONAL. When present, it clearly indicates that the endpoint supports the feature or does not support the feature. However, absence of this field shall also be interpreted as FALSE. The reason is that an older endpoint would not know anything about the field and would obviously not be able to include it. Moreover, they certainly do not support the feature. Another example is the **originator** field in the **perCallInfo** sequence. When present, the meaning is quite clear: the caller is the originator or the terminator of the call. However, if the field is not present, it may mean that the endpoint does not know or cannot supply this information for some reason.

The NULL type is often used to select one of several CHOICE options. NULL carries no particular value, as it merely indicates presence. In selecting the conference goal in a Setup message, for example, the goal CHOICEes are simply NULL types to allow the endpoint to indicate a selection. Another common use of NULL is with the OPTIONAL modifier. A NULL OPTIONAL type allows an endpoint to indicate support for a feature, for example. It is similar in semantics to a BOOLEAN in that the presence of a NULL field indicates TRUE and absence of the NULL field indicates a FALSE. As an example, the **fastConnectRefused** field in the Alerting message is a NULL OPTIONAL. Absence of the field is interpreted as FALSE—Fast Connect is not (yet) refused. Presence of the field, though, clearly indicates refusal of Fast Connect. So why was BOOLEAN not used as the type for this field? It would not have made the encoding any clearer, because the field is past the extension marker (ellipsis). A version 1 and 2 device, for example, would not know to send this field, so there would be three values to consider if BOOLEAN were used: TRUE, FALSE, and *absent*.

Ideally, a field will convey no more values than makes sense. In most cases, these types indicate only two possible values: TRUE/present or FALSE/absent. However, there may be cases where three values are intended and the reader should refer to the appropriate Recommendation to determine if, indeed, there is significance in tri-state fields.

10.2 ASN.1 Usage in H.450-Series Recommendations

This section summarizes the use of ASN.1 in the current H.450.x recommendations. This information is provided for implementers of the H.450.x protocols, as well as authors of new H.450.x Recommendations.

10.2.1 ASN.1 version and encoding rules

The ASN.1 code in H.450.x is based on the 1994 version of X.680-683, including the amendments on "*Rules of extensibility*".

The *basic aligned variant* of *packed encoding rules* (PER) is used as specified in X.691 (1995).

10.2.2 Tagging

All modules defined in Recommendations H.450.x use the *tag default* AUTOMATIC TAGS.

The ROS APDUs (see below) are defined in H.450.1 as *tagged types* within the CHOICE type ROS. No other type defined in H.450.x is a *tagged type*, i.e. all *sets, sequences* and *choices* (except ROS) are automatically tagged.

10.2.3 Basic ASN.1 Types

The following types occur in ASN.1 definitions of H.450.x:

BMPString, NumericString	NULL
BOOLEAN	OBJECT IDENTIFIER

CHOICE	OCTET STRING
CLASS (see below)	*Open type (see below)*
ENUMERATED	SEQUENCE
GeneralizedTime	SEQUENCE OF
INTEGER	SET OF

No use is currently foreseen for the following basic types (needs consideration on a case-by-case basis):

CHARACTER STRING	ObjectDescriptor
EMBEDDED PDV	REAL
EXTERNAL	UTCTime
GeneralString, GraphicString, PrintableString, TeletexString (T61String), UniversalString, VideotexString, VisibleString (ISO646String)	

Use of the following basic types in future recommendations H.450.x should not be precluded (needs consideration on a case-by-case basis):

BIT STRING	Selection Type (out of a CHOICE)
IA5String	SET
INSTANCE OF	TYPE-IDENTIFIER (see X.681)

Note: Some of these types are already used by other recommendations in the H.323 universe, e.g. BIT STRING and TYPE-IDENTIFIER in H.235.

10.2.4 Value sets, subtyping and constraints used in H.450.x:

H.450.x recommendations use *size constraints* (strings, set-of and sequence-of) and *value range* constraints (integers). In H.450.1 *inner subtyping* ("WITH COMPONENTS") is used occasionally.

The use of *value sets*, *single values*, *contained subtypes* and *permitted alphabets* should be possible if needed by future services. The *type constraint* (for restricting an *open type*) may be useful, too.

Explicit set arithmetic (UNION, INTERSECTION, EXCEPT, ALL EXCEPT) is currently not used on subtype specifications.

10.2.5 Object classes, parameterization, general constraints, and ROS

H.450.1 defines a *remote operations service* (ROS) based on X.880. ROS uses *object classes* (X.681), *parameterization* (X.683) and *constraints* (X.682) for its generic part.

Two object classes OPERATION and ERROR are defined and then used to define four PDU types (*Invoke, ReturnResult, ReturnError* and *Reject*) as sequences containing individual parts of these classes. The first three PDU types contain an optional *open type* component which is tied by a *table constraint* ("at (@)" notation) to the code value identifying the particular operation or error.

For each supplementary service the actual operations and errors are then defined as *object instances* of the generic classes OPERATION and ERROR in the corresponding Rec. H.450.x. Each operation and error is identified uniquely (within the context of the H.450.x series) by a code value

(type INTEGER). A list of currently assigned operation and error values is contained in section 10.8 below.

Each supplementary service defines an *object set* containing all operations defined for that service.

10.2.6 Extensibility and non-standard information

Wherever meaningful, an *extension marker* (ellipsis "...") is included in the definitions.

All operations, and some errors, include placeholders for non-standard (e.g. manufacturer-specific) information. This non-standard information can either be of type *NonStandardParameter* (imported from H.225.0) or of type *Extension*, which is defined in H.450.1 and consists of an *object identifier* followed by an *open type*. The definition of the Extension type uses an *object class* (EXTENSION) with *parameterization* and *constraints* similar to the ROS definition.

Usually there is space for more than one addition of non-standard information in an operation. Additions of both types (NonStandardParameter and Extension) can be mixed in any order.

10.2.7 List of Operation and Error Codes

Table 10.1: ASN.1 Operation values used in H.450 series

Value number	Value name	Defined in standard:
0	callingName	H.450.8
1	~~called~~alertingName	H.450.8
2	connectedName	H.450.8
3	busyName	H.450.8
7	callTransferIdentity	H.450.2
8	callTransferAbandon	H.450.2
9	callTransferInitiate	H.450.2
10	callTransferSetup	H.450.2
11	callTransferActive	H.450.2
12	callTransferComplete	H.450.2
13	callTransferUpdate	H.450.2
14	subaddressTransfer	H.450.2
15	activateDiversionQ	H.450.3
16	deactivateDiversionQ	H.450.3
17	interrogateDiversionQ	H.450.3
18	checkRestriction	H.450.3
19	callRerouting	H.450.3
20	divertingLegInformation1	H.450.3
21	divertingLegInformation2	H.450.3
22	divertingLegInformation3	H.450.3
23	cfnrDivertedLegFailed	H.450.3
27	ccnrRequest	Draft H.450.9

28	ccCancel	Draft H.450.9
29	ccExecPossible	Draft H.450.9
31	ccRingout	Draft H.450.9
32	ccSuspend	Draft H.450.9
33	ccResume	Draft H.450.9
40	ccbsRequest	Draft H.450.9
80	mwiActivate	H.450.7
81	mwiDeactivate	H.450.7
82	mwiInterrogate	H.450.7
100	divertingLegInformation4	H.450.3
101	holdNotific	H.450.4
102	retrieveNotific	H.450.4
103	remoteHold	H.450.4
104	remoteRetrieve	H.450.4
105	callWaiting	H.450.6
106	cpRequest	H.450.5
107	cpSetup	H.450.5
108	groupIndicationOn	H.450.5
109	groupIndicationOff	H.450.5
110	pickrequ	H.450.5
111	pickup	H.450.5
112	pickExe	H.450.5
113	cpNotify	H.450.5
114	cpickupNotify	H.450.5

Table 10.2: ASN.1 Error Values used in H.450 series

Value number	**Value name**	**Defined in standard:**
0	userNotSubscribed	H.450.1
1	rejectedByNetwork	H.450.1
2	rejectedByUser	H.450.1
3	notAvailable	H.450.1
5	insufficiantInformation	H.450.1
6	invalidServedUserNumber	H.450.1
7	invalidCallState	H.450.1
8	basicServiceNotProvided	H.450.1
9	notIncomingCall	H.450.1

10	supplementaryServiceInteractionNotAllowed	H.450.1
11	resourceUnavailable	H.450.1
12	invalidDivertedNumber	H.450.3
14	specialServiceNumber	H.450.3
15	diversionToServedUserNumber	H.450.3
24	numberOfDiversionsExceeded	H.450.3
25	callFailure	H.450.1
31	notActivated	H.450.7
43	proceduralError	H.450.1
1000	temporarilyUnavailable	H.450.3
1004	invalidReroutingNumber	H.450.2
1005	unrecognizedCallIdentity	H.450.2
1006	establishmentFailure	H.450.2
1007	notAuthorized	H.450.3
1008	unspecified	H.450.2, H.450.3
1010	shortTermRejection	Draft H.450.9
1011	longTermRejection	Draft H.450.9
1012	remoteUserBusyAgain	Draft H.450.9
1013	failureToMatch	Draft H.450.9
1018	invalidMsgCentreId	H.450.7
2000	callPickupIdUnvalid	H.450.5
2001	callAlreadyPickedUp	H.450.5
2002	undefined	H.450.4, H.450.5, H.450.7, H.450.9

Annex: H.323 Recommendation Series Defect Report Form

DATE:	
CONTACT INFORMATION **NAME:** **COMPANY:** **ADDRESS:** **TEL:** **FAX:** **EMAIL:**	
AFFECTED RECOMMENDATIONS:	
DESCRIPTION OF PROBLEM:	
SUGGESTIONS FOR RESOLUTION:	

NOTE - Attach additional pages if more space is required than is provided above.