INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2001 - 2004

COM 16-R 5-E
December 2000
Original: English

Question(s): D, F, G/16, 1, 2, 3, 4 and 5/16

Texte disponible seulement en
Text available only in
Texto disponible solamente en
E

STUDY GROUP 16 – REPORT R 5

SOURCE*: Study Group 16 (Geneva meeting, 13-17 November 2000)

TITLE: Implementers Guide for H.323, H.225.0, H.245, H.246, H.283, H.235, H.450 Series, and H.341 Recommendations

---

*** Contact:** TSB

Tel: +41 22 730 5860
Fax: +41 22 730 5853
Email: tsbsg16@itu.int

Contact Information

ITU-T Study Group 16 / Question 13 Rapporteur ITU-T Recommendation H.323 and Implementer's Guide Editor	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 6801 paulej@packetizer.com
ITU-T Study Group 16 / Question 14 Rapporteur ITU-T Recommendation H.450.8 and H.341 Editor	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
ITU-T Recommendation H.225.0 Editor	Rich Bowen Cisco Systems, Inc. 7025 Kit Creek Road Research Triangle Park, NC 27709 USA	Tel: Fax: E-mail:	+1 919 392 3890 +1 919 392 6801 rkbowen@cisco.com
ITU-T Recommendation H.225.0 Annex G Editor	Michael Fortinsky VocalTec Communications, Ltd. 2 Maskit St. Herzeliya 46733 Israel	Tel: Fax: E-mail:	+972 9 970 7768 +972 9 956 1867 mike@vocaltec.com
ITU-T Recommendation H.235 Editor	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@icn.siemens.de
ITU-T Recommendation H.245 Editor	Mike Nilsson BT Labs Ipswitch United Kingdom	Tel: Fax: E-mail:	+44 1 473 645413 +44 1 473 643791 mike.nilsson@bt-sys.bt.co.uk
ITU-T Recommendation H.450.1, H.450.2, and H.450.3, H.450.4, H.450.5, H.450.6 Editor	Markku Korpi Siemens AG Munich Germany	Tel: Fax: E-mail:	+49 89 722 34570 +49 89 722 23977 korpim@sbs.de
ITU-T Recommendation H.450.7 Editor	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

Table of Contents

1 Introduction

This document is a compilation of reported defects identified with the 1999 decided edition of ITU-T Recommendation H.323 and related H.323-series Recommendations. 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.

2 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.

3 Defect Resolution Procedure

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 anyone interested in H.323 series Recommendations. 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 (1999), *Packet-Based multimedia communications systems*
- ITU-T Recommendation H.225.0 (1999), *Call signaling protocols and media stream packetization for packet based multimedia communications Systems*
- ITU-T Recommendation H.225.0 – Annex G (1999), *Communication Between Administrative Domains*
- ITU-T Recommendation H.245 (2000), *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 (1998), 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*
- 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.

6.1 Technical and Editorial Corrections to ITU-T Recommendation H.323 (1999)

6.1.1 Termination of Fast Connect when using H.245 Tunneling

Description:	An ambiguity exists regarding the termination of Fast Connect when using H.245 tunneling. The text below attempts to correct this ambiguity.

[Begin Correction]

8.1.7.2 Switching to H.245 procedures

After establishment of a call using the Fast Connect procedure, either endpoint may determine that it is necessary to invoke call features that require the use of H.245 procedures. Either endpoint may initiate the use of H.245 procedures at any point during the call, using tunneling as described in 8.2.1 (if **h245Tunneling** remains enabled) or a separate H.245 connection. The process for switching to a separate H.245 connection is described in 8.2.3.

It is possible to switch to H.245 procedures before the Fast Connect procedure completes. If **h245Tunneling** is enabled, the terminating party may start using tunneling as described in 8.2.1. If H.245 transport address is included in the Setup message, then the terminating party may start the switch to H.245 as described in 8.2.3.

When a call is established using the Fast Connect procedure, both endpoints shall keep the Q.931 Call Signalling Channel open until either the call is terminated or a separate H.245 connection is established.

...

[End Correction]

6.1.2 Tones and Announcements

Description:	H.323 does not explicitly describe how tones and announcements should be provided, although implicit procedures may be derived from the Q.931 heritage of H.225.0. The Fast Connect procedures allow "early cut through" of the media stream for providing ringback tones, but no mention is made of how the originating is supposed to decide if locally generated ringback shall be applied or not. The text below shall be added to H.323 to clarify this issue.

[Begin Correction]

8.1.7.4 Tones and announcements

Tones and announcements can be locally generated or passed in-band from the terminating endpoint.

On completing call setup, the endpoint on the terminating side shall decide if it will provide in-band tones or if locally generated tones at the originating side shall be used. Note that other type of indication can replace locally generated tones and announcement in some systems (visual indications on a screen for example: for the purpose of this section, they will be referred-to as locally generated tones and announcements). Locally generated tones, provided at the originating side, is the default. The terminating side may wish to provide in-band-generated tones and announcements, for example when the terminating endpoint is a gateway to an analogue network. To instruct the originating side not to generate locally generated tones such as ringback or busy, the terminating side shall open the media channel by responding to the Fast Connect request and send a Progress indicator information element with progress descriptor #1, *Call is not end-to-end ISDN; further call progress information may be available in-band*, or #8, *In-band information or an appropriate pattern is now available* in a Call Proceeding, Progress or Alerting message, or in a Connect message if an Alerting message was not sent. The response to the Fast Connect message shall be done before or at the same time the Progress indicator is sent (i.e., up to and including the same message the Progress indicator is sent). The terminating side can provide in-band tones or announcements (such as ringback or busy) as soon as the progress descriptor has been sent and the media channel has been opened. Note that the Progress indicator should be in an Alerting message only if the endpoint is being alerted. If another in-band tone, such as busy or re-order tone is provided, the Progress indicator should not be in an Alerting. When no appropriate call setup message is available, a Progress message can be used to carry the Progress indicator.

Note – When an endpoint or a Gatekeeper intervening in call signalling receives a Progress indicator information element in a Call Proceeding message, it will not be able to relay the Call Proceeding if the Call Proceeding message has already been sent to the originating side. In that case, the Progress indicator information element in the Call Proceeding message shall be mapped to a Progress indicator information element in a Progress message.

If the terminating side does not wish to provide far-end tones and announcements, it shall not send a Progress indicator information element with progress descriptor #1 or #8. To instruct the originating side that locally generated alerting shall be applied, the Alerting message shall be sent.

Upon receipt of an Alerting message, the originating side shall provide locally generated tones and announcement unless both the following conditions are true:

1) A media channel is available for "listening". The fastStart element could have been received in any message up to and including Alerting message.

2) A Progress indicator information element with progress descriptor #1, *Call is not end-to-end ISDN; further call progress information may be available in-band*, or #8, *In-band information or an appropriate pattern is now available*, was received in any message up to and including the Alerting message.

Upon receipt of a Release Complete message including a Cause information element, the originating side shall generate a tone or provide an indication appropriate to the received cause value. For example, if cause value #17, *User busy*, is received, the originating shall generate busy tone or provide an indication of user busy.

When locally generated tones and announcements are used, the Signal information element can optionally also be present to include more information about the type of signal to be provided.

[End Correction]

6.1.3 Correct H.245 Version for H.323 Version 1 Devices

Description:	An editorial error was discovered in the H.323 (1998) and H.323 (1999) publications. It specifies that for H.323 (1996) equipment, H.245 (1996) is required. The correct version of H.245 that should be specified in those Recommendations is H.245 (1997). The corrected text, taken from H.323 (1999), is shown below.

[Begin Correction]

Summary

...

Products claiming compliance with Version 1 of H.323 shall comply with all of the mandatory requirements of H.323 (1996) which references Recommendations H.225.0 (1996) and H.245 (~~1996~~1997). Version 1 products can be identified by H.225.0 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 2250 version (0) 1} and H.245 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 245 version (0) 2}. Products claiming compliance with Version 2 of H.323 shall comply with all of the mandatory requirements of this Recommendation, H.323 (1998), which references Recommendations H.225.0 (1998) and H.245 (1998). Version 2 products can be identified by H.225.0 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 2250 version (0) 2} and H.245 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 245 version (0) 3}. Products claiming compliance with Version 3 of H.323 shall comply with all of the mandatory requirements of this Recommendation, H.323 (1999), which references Recommendation H.225.0 (1999) and H.245 (1999). Version 3 products can be identified by H.225.0 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 2250 version (0) 3} and H.245 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 245 version (0) 5}.

...

[End Correction]

6.1.4 Clarification of Call Identification Fields

Description:	H.225.0 Version 3 introduced new fields for caller identification without procedural text describing the usage of those fields. To prevent interoperability issues, that procedural text is presented here and will be introduced into the next revision of H.323.

[Begin Correction]

7.8 Caller identification services

7.8.1 Description of services

This section describes the caller identification services, which includes:

- Calling party number presentation and restriction

- Connected party number presentation and restriction
- Called (Alerting) party number presentation and restriction
- Busy party number presentation and restriction

7.8.1.1 Calling party address presentation

Calling party address presentation is a feature which provides the alias address of the calling party to the called party. The calling party address may be provided by the calling endpoint or by the gatekeeper for Gatekeeper routed calls that originate in the packet network. When the call is routed through the gatekeeper with which the calling endpoint is registered, the Gatekeeper may provide a screening service that assures the address provided is actually that of the calling party. The Gatekeeper may also provide the calling party address when no address is provided by the calling party or when the calling party provides an address other than an address with which the calling party registered.

When a call originates in the switched circuit network and enters the packet network through a Gateway, the Gateway shall pass to the packet network the calling party number information provided from the switched circuit network.

7.8.1.2 Calling party address restriction

Calling party address restriction is a feature which allows the calling endpoint or the calling endpoint's Gatekeeper to restrict presentation of the calling party alias address to the called party. This feature may reside in the endpoint or in the Gatekeeper for Gatekeeper routed calls.

In some cases where calling party address restriction has been indicated, there may exist certain situations where the restriction is overridden (for example, if the called party provides some emergency service).

7.8.1.3 Connected party address presentation

Connected party address presentation is a feature which provides the alias address of the connected or answering party to the calling party. The connected party address may be provided by the connected endpoint or by the Gatekeeper for Gatekeeper routed calls. When the call is routed through the Gatekeeper with which the connected endpoint is registered, the Gatekeeper may provide a screening service that assures the address provided is actually that of the connected party. The Gatekeeper may also provide the connected party address when no address is provided by the connected party or when the connected party provides an address other than an address with which the connected party registered.

A Gateway shall pass connected party information received from the switched circuit network to the packet network.

7.8.1.4 Connected party address restriction

Connected party address restriction is a feature which allows the connected endpoint or the connected endpoint's Gatekeeper to restrict presentation of the connected party alias address to the calling party. This feature may reside in the endpoint or in the Gatekeeper for Gatekeeper routed calls.

In some cases where connected party address restriction has been indicated, there may exist certain situations where the restriction is overridden (for example, if the calling party provides some emergency service).

7.8.1.5 Called (alerting) party address presentation

Alerting party address presentation is a feature which provides the alias address of the alerting party to the calling party. The alerting party address may be provided by the alerting endpoint or by the Gatekeeper for Gatekeeper routed calls. When the call is routed through the gatekeeper with which the alerting endpoint is registered, the Gatekeeper may provide a screening service that assures the address provided is actually that of the alerting party. The Gatekeeper may also provide the alerting party address when no address is provided by the alerting party or when the alerting party provides an address other than an address with which the alerting party registered.

7.8.1.6 Called (alerting) party address restriction

Alerting party address restriction is a feature which allows the alerting endpoint or the alerting endpoint's Gatekeeper to restrict presentation of the alerting party alias address to the calling party. This feature may reside in the endpoint or in the Gatekeeper for Gatekeeper routed calls.

7.8.1.7 Busy party address presentation

Busy party address presentation is a feature which provides the alias address of the busy party to the calling party. The busy party address may be provided by the busy endpoint or by the Gatekeeper for Gatekeeper routed calls. When the call is routed through the gatekeeper with which the busy endpoint is registered, the Gatekeeper may provide a screening service that assures the address provided is actually that of the busy party. The Gatekeeper may also provide the busy party address when no address is provided by the busy party or when the busy party provides an address other than an address with which the busy party registered.

7.8.1.8 Busy party address restriction

Busy party address restriction is a feature which allows the busy endpoint or the busy endpoint's Gatekeeper to restrict presentation of the busy party alias address to the calling party. This feature may reside in the endpoint or in the Gatekeeper for Gatekeeper routed calls.

7.8.2 Messages and information elements

This section describes the various messages and information elements that allow H.323 devices to provide address presentation and restriction services.

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).

The presentationIndicator field in the Setup message carries information identical to the presentation indicator found in the Calling Party Number IE. The meaning and use of the presentation indicator is defined in Q.951.

The screeningIndicator field in the Setup message carries information identical to the screening indicator found in the Calling Party Number IE. The meaning and use of the screening indicator is defined in Q.951.

7.8.2.2 Connected party address information

Connected party address information appears in the Connect message.

When address information represents a telephone number, the relevant information may appear in the Connected Number IE, including the presentation indicator and screening indicator. This is the recommended mode of operation for the case where a PSTN Gateway sends a Connect message on the packet network.

Alternatively, connected party information may appear in the connectedAddress, presentationIndicator, and screeningIndicator fields of the Connect message. This mode of operation is required when connectedAddress is not in any form of telephone number (i.e., connectedAddress is not type dialedDigits or partyNumber).

The presentationIndicator field in the Connect message carries information identical to the presentation indicator found in the Connected Number IE. The meaning and use of the presentation indicator is defined in Q.951.

The screeningIndicator field in the Connect message carries information identical to the screening indicator found in the Connected Number IE. The meaning and use of the screening indicator is defined in Q.951.

7.8.2.3 Called (alerting) party address information

Alerting party address information appears in the Alerting message.

Alerting party information may appear in the alertingAddress, presentationIndicator, and screeningIndicator fields of the Alerting message.

The presentationIndicator field in the Alerting message carries information identical to the presentation indicator found in the Connected Number IE. The meaning and use of the presentation indicator is defined in Q.951.

The screeningIndicator field in the Alerting message carries information identical to the screening indicator found in the Connected Number IE. The meaning and use of the screening indicator is defined in Q.951.

7.8.2.4 Busy party address information

Busy party address information appears in the Release Complete message.

Busy party information may appear in the busyAddress, presentationIndicator, and screeningIndicator fields of the Release Complete message.

The presentationIndicator field in the Release Complete message carries information identical to the presentation indicator found in the Connected Number IE. The meaning and use of the presentation indicator is defined in Q.951.

The screeningIndicator field in the Release Complete message carries information identical to the screening indicator found in the Connected Number IE. The meaning and use of the screening indicator is defined in Q.951.

7.8.3 Actions at the originating endpoint

This section describes the procedural aspects required to provide caller identification services at the originating endpoint.

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.

A Gateway in receipt of a Connect message shall copy the Connected Number IE from the Connect message from the packet network to the Connect message to be sent to the ISDN. If the Connected Number IE is not present in the Connect message, the Gateway shall convert connectedAddress, presentationIndicator, and screeningIndicator into a Connected Number IE, if that connectedAddress represents some form of telephone number. If connectedAddress does not represent some form of telephone number or if the Connected Number IE is not present in the Connect message, the Gateway shall omit the Connected Number IE from the Connect message sent to the ISDN.

A Gateway in receipt of an Alerting message with alerting party information or a Release Complete message with busy party information shall convert the party information to the signaling format of the Gateway's circuit side if the signaling format supports this party information.

7.8.3.2 Terminal or MCU as originating endpoint

For calls originated on the packet network, the originating terminal or MCU may send a Setup message with either the Calling Party Number IE with presentation and screening indicators or with sourceAddress, presentationIndicator, and screeningIndicator fields. In either case, the screening indicator shall indicate "user provided not screened". As an example, if the caller wants to block identification to the called party, the presentation indicator would be set to "presentation restricted", but the caller's number would still appear in the Calling Party Number IE. In Gatekeeper routed cases, the calling party's Gatekeeper may add this information if it is missing or incorrect and the called party's Gatekeeper may remove the caller's identification information if appropriate. The calling party's Gatekeeper or the called party's Gatekeeper may also add or remove address information based on local policy.

A terminal or MCU in receipt of a Connect, Alerting, or Release Complete message should honor the presentation indicator when presenting address information to the user.

7.8.4 Actions at the terminating endpoint

This section describes the procedural aspects required to provide caller identification services at the terminating endpoint.

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 signaling 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, 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.

The Gateway shall send a Connect message on the packet network with the Connected Number IE containing the same information as was found in the signaling format supported in the telephone network. In the case of a Q.931 Connect message received by a Gateway from the ISDN, connected party information resides in the Connected Number IE.

7.8.4.2 Terminal or MCU as terminating endpoint

A terminal or MCU in receipt of the Setup message should honor the presentation indicator when presenting caller information to the user.

For calls answered on the packet network, the answering terminal or MCU may include in the Connect message either the Connected Number IE or connectedAddress, presentationIndicator, and screeningIndicator fields. In either case, the terminal or MCU shall set the screeningIndicator to indicate "user provided not screened". In Gatekeeper routed cases, the answering party's Gatekeeper may add this information if it is missing or incorrect and the calling party's Gatekeeper may remove the answering party's address information if appropriate.

A terminal or MCU may provide address information in the Alerting message, using the alertingAddress, presentationIndicator, and screeningIndicator found in the Alerting message. If the address is provided, the terminal or MCU shall set the screeningIndicator to indicate "user provided not screened". In Gatekeeper routed cases, the answering party's Gatekeeper may add this information if it is missing or incorrect and the calling party's Gatekeeper may remove the answering party's address information if appropriate. The answering party's Gatekeeper or the calling party's Gatekeeper may also add or remove address information based on local policy.

A busy terminal or MCU may provide address information in the Release Complete message, using the busyAddress, presentationIndicator, and screeningIndicator found in the Release Complete message. If the address is provided, the terminal or MCU shall set the screeningIndicator to indicate "user provided not screened". In Gatekeeper routed cases, the answering party's Gatekeeper may add this information if it is missing or incorrect and the calling party's Gatekeeper may remove the answering party’s address information if appropriate.

7.8.5 Actions at a gatekeeper

In Gatekeeper routed scenarios, the Gatekeeper may provide identification information or may provide a screening service. Services that may be provided by a Gatekeeper depend on the type of endpoint served. This section describes the procedural aspects required to provide caller identification services when the Gatekeeper routes the call signalling.

7.8.5.1 Gateway as originating endpoint

In Gatekeeper routed cases, a Gatekeeper should not modify the information found in the Setup message sent from a Gateway. This assumes that the telephone network has provided correct information.

7.8.5.2 Terminal or MCU as originating endpoint

In Gatekeeper routed cases, a Gatekeeper may provide calling party information when the calling party is not a Gateway. The Gatekeeper may provide a calling party address if the calling party did not provide one or if the Gatekeeper determines the address is not correct. If the Gatekeeper provides an address other than that sent in the Setup message, the Gatekeeper shall set the screening indicator to indicate "network provided". If the Gatekeeper verifies the address information sent in the Setup message, but does not modify the address information, the Gatekeeper shall set the screening indicator to indicate "user provided, verified, and passed". If the Gatekeeper determines that the address information sent in the Setup message is incorrect, but does not modify the address information, the Gatekeeper shall set the screening indicator to indicate "user provided, verified, and failed". The Gatekeeper may set the presentation indicator to provide service to the endpoint. The Gatekeeper may allow the endpoint to override the endpoint's service by specifying a different presentation (for example, restricting presentation for the current call when the endpoint's service is to allow presentation).

7.8.5.3 Gateway as terminating endpoint

In Gatekeeper routed cases, a Gatekeeper should not modify the information found in the Connect message sent from a Gateway. This assumes that the telephone network has provided correct information.

7.8.5.4 Terminal or MCU as terminating endpoint

In Gatekeeper routed cases, a Gatekeeper may provide connected, alerting, or busy party information when the connected, alerting, or busy party is not from a Gateway. The Gatekeeper may provide a connected party (or alerting party, or busy party) address if none was provided by the connected party (or alerting party, or busy party), or if the Gatekeeper determines the address is not correct. If the Gatekeeper provides an address other than that sent in the Connect, Alerting, or Release Complete message, the Gatekeeper shall set the screening indicator to indicate "network provided". If the Gatekeeper verifies the address information sent in the Connect, Alerting, or Release Complete message, but does not modify the address information, the Gatekeeper shall set the screening indicator to indicate "user provided, verified, and passed". If the Gatekeeper determines that the address information sent in the Connect, Alerting, or Release Complete message is incorrect, but does not modify the address information, the Gatekeeper shall set the screening indicator to indicate "user provided, verified, and failed". The Gatekeeper may set the presentation indicator to provide service to the endpoint. The Gatekeeper may allow the endpoint to override the endpoint's service by specifying a different presentation (for example, restricting presentation for the current call when the endpoint's service is to allow presentation).

[End Correction]

6.1.5 Clarification of the Fast Connect Procedure

Description:	It was noted that some text within the Fast Connect procedure was ambiguous. This section attempts to clarify some issues within section 8.1.7.1 of H.323 (1999).

[Begin Correction]

8.1.7.1 Proposal, selection and opening of media channels

...

In an **openLogicalChannel** which proposes a channel for transmission from the calling endpoint to the called endpoint, the **forwardLogicalChannelParameters** element shall contain parameters specifying the characteristics of the proposed channel, and the **reverseLogicalChannelParameters** element shall be omitted. Each such **OpenLogicalChannel** structure shall have a unique **forwardLogicalChannelNumber** value. Alternative proposals for the same transmit channel shall contain the same **sessionID** value in **H2250LogicalChannelParameters**. The **mediaChannel** element shall be omitted in the proposal; it will be provided by the called endpoint should the proposal be accepted. The other **H2250LogicalChannelParameters** and **dataType** shall be set to correctly describe the transmit capabilities of the calling endpoint associated with this proposed channel. The calling endpoint may choose not to propose any channels for transmission from the calling endpoint to the called endpoint, such as if it desires to use H.245 procedures later to establish such channels.

In the Setup message, each **openLogicalChannel** that proposes a channel for transmission from the calling endpoint to the called endpoint shall contain the **mediaControlChannel** element (indicating the reverse RTCP channel) in the **H2250LogicalChannelParameters** element of the **forwardLogicalChannelParameters** structure.

In an **openLogicalChannel** which proposes a channel for transmission from the called endpoint to the calling endpoint, the **reverseLogicalChannelParameters** element shall be included and contain parameters specifying the characteristics of the proposed channel. The **forwardLogicalChannelParameters** element must also be included (because it is not optional), with the **dataType** element set to **nullData**, **multiplexParameters** set to **none**, and all optional elements omitted. Alternative proposals for the same receive channel shall contain the same **sessionID** value in **H2250LogicalChannelParameters**. All alternative **OpenLogicalChannel** structures, that propose a channel for transmission from the called endpoint to the calling endpoint, shall contain the same **sessionID** and the same **mediaChannel** value. The other **H2250LogicalChannelParameters** and **dataType** within **reverseLogicalChannelParameters** shall be set to correctly describe the receive capabilities of the calling endpoint associated with this proposed channel. The calling endpoint may choose not to propose any channels for transmission from the called endpoint to the calling endpoint, such as if it desires to use H.245 procedures later to establish such 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 slightly different parameters.

...

When accepting a proposed channel for transmission from called endpoint to calling endpoint, the called endpoint shall return the corresponding **OpenLogicalChannel** structure to the calling endpoint, inserting a unique **forwardLogicalChannelNumber** into the ~~**forwardLogicalChannelParameters**~~ **OpenLogicalChannel** structure and a valid mediaControlChannel element (indicating the reverse RTCP channel) into the **H2250LogicalChannelParameters** element of the **reverseLogicalChannelParameters** structure. ~~All **mediaControlChannel** elements inserted by the called endpoint for the same **sessionID** for both directions shall have the same value.~~ The called endpoint may begin transmitting media on the accepted channel according to the parameters specified in **reverseLogicalChannelParameters** immediately after sending the Q.931 response containing **fastStart**, unless **mediaWaitForConnect** was set to TRUE in which case it must wait until after sending the Connect message.

When accepting a proposed channel for transmission from the calling endpoint to the called endpoint, the called endpoint shall return the corresponding **OpenLogicalChannel** structure to the calling endpoint. The called endpoint shall insert valid **mediaChannel** and

mediaControlChannel fields (indicating the RTCP channel going in the same direction) into the **h2250LogicalChannelParameters** element of the **forwardLogicalChannelParameters** structure. All **mediaControlChannel** elements inserted by the called endpoint for the same **sessionID** for both directions shall have the same value. The called endpoint shall then prepare to immediately receive media flow according to the parameters specified in **forwardLogicalChannelParameters**. The calling endpoint may begin transmitting media on the accepted and opened channels upon receipt of the Q.931 response containing **fastStart**, and may release any resources allocated to reception on proposed channels that were not accepted.

[End Correction]

6.1.6 Call Linkage

Description:	It has become apparent that for certain applications, such as Automatic Call Distribution and Billing, there is a need to "link" calls together when certain supplementary services are performed. Some implementers have attempted to use the Call Identifier for this purpose, but it is not well suited for the task. The section is introduced to overcome this shortcoming and to provide implementers with the necessary tools.

[Begin Correction]

10.3 Call Linkage in H.323

10.3.1 Description

Call Linkage in H.323 is an optional feature. A term "shall" within this section 10.3 shall be interpreted as a mandatory requirement provided the Call Linkage feature is supported.

10.3.1.1 General description

The Thread Identification feature allows different calls or call independent signalling connections – those that logically belong together from a service's or application's point of view in terms of their progression – to be linked together.

The Global Call Identification feature allows a call or a call independent signalling connection to be identified by one unique identifier that is applicable to the call or call independent signalling connection end-to-end without regards to its route or its history.

NOTE – The Call Identifier is defined in section 7.5 as a globally unique identifier for a call. A new basic call from the same endpoint/entity or a new call as part of a service scenario would use a new Call Identifier value.

10.3.1.2 Service definitions

10.3.1.2.1 Thread identification, thread ID, TID

A value assigned to calls that are logically linked together for the purpose of correlating them. If two or more calls are logically linked together (e.g. due to service interactions), the current Thread ID of one of these calls is assigned to all of the other linked calls.

10.3.1.2.2 Global call identification, global call ID, GID

A value assigned to an end-to-end call to uniquely identify that call from end-to-end. If different calls are being transformed into a new call (i.e. due to service interactions), the

GIDs of the old calls are updated (if already assigned previously) or assigned by a new GID value for the new end-to-end call.

NOTE – A call that is being transformed out of different call legs due to certain services may end up having call legs with different Call Identifiers. The Call Identifier is therefore not suitable to uniquely identify a call end-to-end.

10.3.2 Invocation and operation

A Call ID shall be assigned to each new call that is set up (see Section 7.5). Due to service interactions, different Call IDs may be assigned to different parts (call legs) of a call.

A Global Call ID may be assigned either at call establishment time, while in the active state or while call establishment/call clearing is in progress when two or more calls are being transformed into a new call due to certain services being invoked or due to an application request.

A Global Call ID may be changed during the lifetime of the call due to the call being transformed.

A Thread ID may be assigned either at call establishment time, while in the active state or while call establishment/call clearing is in progress when two or more calls are logically linked together due to certain services being invoked or due to an application request.

The Thread ID may be changed during the lifetime of a call (e.g. due to service interactions).

10.3.3 Interaction with H.450 supplementary services

Interactions with H.450 supplementary services for which standards were available at the time of publication of this Recommendation are specified below.

For the Call ID, no interactions with other supplementary services apply, as it shall be unique for each new call. All interactions described in this section apply only to the Global Call ID and/or the Thread ID.

A Global Call ID and a Thread ID may be assigned, regardless of a supplementary service invocation, as part of the basic call establishment. Specific feature interactions are described below for specific supplementary service invocations.

10.3.3.1 Call transfer

This section describes the usage of the Call Linkage fields when using H.450.2.

10.3.3.1.1 Transfer without consultation

The Thread ID of the transferred call shall be inherited from the Thread ID of the primary call. The Thread ID of the primary call shall therefore be provided by the transferring endpoint to the transferred endpoint along with the call transfer request. If the primary call does not have an assigned Thread ID, the transferring endpoint shall generate one. If the transferred entity does not receive a Thread ID along with the call transfer request, it shall inherit the Thread ID that was assigned to the primary call at call establishment time. If no Thread ID is available to inherit from at all, the transferred endpoint shall generate a Thread ID and assign it to both the transferred call (in call establishment message) and the primary call (in call clearing message).

A new Global Call ID shall be assigned to a transferred call. If a Gatekeeper establishes the transferred call on behalf of a transferred endpoint, the Gatekeeper shall assign the same

Global Call ID to the remaining call leg of the primary call. This ensures that the resulting call after successful transfer has one unique GID end-to-end.

10.3.3.1.1 Transfer with consultation

At the time of transfer, the transferred call shall be assigned the same Thread ID as the former primary call if:

a) the primary call is an incoming call and the secondary call is an outgoing call, or

b) both calls are incoming calls and the primary call has been established before the secondary call, or

c) both calls are outgoing calls and the primary call has been established before the secondary call.

At the time of transfer, the transferred call shall be assigned the same Thread ID as the former secondary call if:

a) the secondary call is an incoming call and the primary call is an outgoing call, or

b) both calls are incoming calls and the secondary call has been established before the primary call, or

c) both calls are outgoing calls and the secondary call has been established before the primary call.

The Thread ID appropriate for the transferred call (either based on primary or secondary call depending on the situation) shall be provided by the transferring endpoint to the transferred endpoint along with the call transfer request. If the call from which the Thread ID shall be inherited (either primary or secondary call) does not have assigned a Thread ID, the transferring endpoint shall generate one. If the transferred endpoint does not receive a Thread ID along with the call transfer request (e.g. transferring endpoint does not support call linkage), it shall generate a Thread ID that shall be inherited from the primary call if possible.

At the time of transfer, the transferred entity shall assign a new GID value to the transferred call. If a Gatekeeper established the transferred call on behalf of a transferred endpoint, the Gatekeeper shall assign the same GID to the remaining call leg of the primary call. A Gatekeeper acting on behalf of the transferred-to endpoint shall assign the same GID to the remaining part of the secondary call. This ensures that the resulting call after successful transfer has one unique GID end-to-end.

A transferring entity may, as an option, choose to "join" the primary call and the secondary call together. The call linkage rules for the resulting call ("joined" call) shall be the same as specified for a transferred call above.

10.3.3.2 Call diversion

This section describes the usage of the Call Linkage fields when using H.450.3.

The originating call, the forwarding and the forwarded call shall use the same Thread ID.

The Thread ID of the forwarded call and the originating call shall be inherited from the Thread ID of the forwarding call. The served endpoint shall therefore assign a Thread ID to the forwarding call (if not already assigned as part of the basic call) and shall provide this Thread ID to the re-routing entity along with the call forwarding request. The re-routing entity shall use this Thread ID as the Thread ID for the establishment of the forwarded call. In addition, the originating call leg (if any) shall be assigned/updated with this Thread ID as well.

If the re-routing entity does not receive a Thread ID along with the call forwarding request, it shall inherit the Thread ID that was assigned to the forwarding call at call establishment time. If no Thread ID is available to inherit from at all, the re-routing endpoint shall generate a Thread ID and assign it to the forwarding call, the forwarded call, and to the originating call.

A new GID shall be assigned to the end-to-end call from the calling user (i.e., diverted user) to the diverted-to user by assigning a new GID in the forwarded call Setup and assigning (or updating) the same GID to the originating call leg (if any).

10.3.3.3 Call hold and consultation

This section describes the usage of the Call Linkage fields when using H.450.4.

A consultation call shall use the same Thread ID as the first call.

NOTE – Whether a call is considered being a consultation call rather than a further basic call is the decision of the endpoint.

A consultation call shall use a new Global Call ID.

10.3.3.4 Call park/call pickup

This section describes the usage of the Call Linkage fields when using H.450.5.

The parked call shall have the same Thread ID as the primary call; however, it shall use a different GID.

If available, the Thread ID shall be used for associating call independent signalling connections (indicating group notifications and pickup requests), the call from a calling/parked user to the picking-up user, and a previously alerting/parked call.

NOTE – Call Park/Pickup contains a specific call pickup id that is used by the picking-up user.

The call independent signalling connections used as part of Call Park / Call Pickup shall use new GIDs. The call from the calling user/parked user to the picking-up user shall have a new end-to-end global GID.

10.3.3.5 Call waiting

There is no interaction with Call Linkage and H.450.6.

10.3.3.6 Message waiting indication

There is no interaction with Call Linkage and H.450.7.

10.3.3.7 Name identification service

There is no interaction with Call Linkage and H.450.8.

[End Correction]

ASN.1 changes required to support Call Linkage appears in section 6.4.9.

6.1.7 Early Termination of Fast Connect

Description:	A race condition exists in the text of H.323 that states that opening a separate H.245 connection terminates the Fast Connect procedure. The problem is that, due to certain network conditions, an endpoint may receive an H.245 connection prior to receiving a Connect message. This should not result in an early termination of Fast Connect.

[Begin Correction]

8.2.1 Encapsulation of H.245 messages within Q.931 messages

...

The calling endpoint shall *not* include both a **fastStart** element and encapsulated H.245 messages in **h245Control** in the same **SETUP** message, since the presence of the encapsulated H.245 messages would override the Fast Connect procedure. A calling endpoint may, however, include both a **fastStart** element and set **h245Tunneling** to TRUE within the same **SETUP** message; likewise, a called endpoint may include **fastStart** and set **h245Tunneling** to TRUE within the same Q.931 response. In this case, the Fast Connect procedures are followed, and the H.245 connection remains "unestablished" until actual transmission of the first tunneled H.245 message or opening of the separate H.245 connection.~~ The sending of encapsulated H.245 messages or the initiation of the separate H.245 connection by either endpoint prior to the sending of a Q.931 message containing **fastStart** by the called endpoint terminates the Fast Connect procedures.~~

...

[End Correction]

6.1.8 Assignment of the maintainConnection Field

Description:	Implementers have noted that the text in H.323 is not clear on the subject of whether the maintainConnection field shall remain "constant" or may be changed during a call. This section attempts to clarify that issue.

[Begin Correction]

7.3 Call signalling channel

...

The Call Signalling Channel may be established prior to the actual need to signal a call, and the channel may remain connected between calls. An entity may indicate this capability by setting the **maintainConnection** flag to TRUE in messages that it sends on the Call Signalling Channel. In addition, an endpoint which has this capability should indicate this when it registers with a gatekeeper. This will allow a gatekeeper that utilizes gatekeeper routing to connect to the endpoint at any point after registration. If the connection drops while no call or signalling is active, neither end shall attempt to open the connection until signalling is needed.

The value of the **maintainConnection** flag sent by an entity over a given Call Signalling Channel shall be the same for every message containing this field for the duration of the Call Signalling Channel. This does not preclude an entity from setting this value to TRUE for one Call Signaling Channel and FALSE for another Call Signalling Channel.

[End Correction]

6.1.9 Third Party Pause and Re-routing

Description:	An editorial error has been discovered in the published H.323v3 text in the

	section Third Party Pause and Re-routing. The text below shows the correction that shall be applied to that text. This erroneous text contradicts text that appears several paragraphs above, which states that an endpoint shall retain the state of receive logical channels that may be open.

[Begin Correction]

8.4.6 Third party initiated pause and re-routing

...

NOTE – A non-empty capability set shall not be sent to an endpoint until all of its transmit ~~and receive~~ logical channels have been closed. A switching entity should also send an H.450 redirection indication Facility message if the endpoint is being re-routed.

[End Correction]

6.1.10 Clarifying the URQ/UCF/URJ Exchange from the Endpoint to the GK

Description:	The text in H.323 Sections 7.2.2 is contradictory in what message a Gatekeeper should return in response to a URQ from an endpoint. In addition, there may be reasons for returning a URJ or UCF, which may be a Gatekeeper policy matter. The text below shows the modified text.

[Begin Correction]

7.2.2 Endpoint registration

...

An endpoint may cancel its registration by sending an Unregister Request (URQ) message to the Gatekeeper. ~~The Gatekeeper shall respond with an Unregister Confirmation (UCF) message.~~ This allows endpoints to change the alias address associated with its Transport Address, or vice versa. ~~If the endpoint was not registered with the Gatekeeper, it shall return an Unregister Reject (URJ) message to the endpoint.~~ The Gatekeeper shall respond with either an Unregister Confirmation (UCF) message or an Unregister Reject (URJ) message according to Gatekeeper policy.

A Gatekeeper may cancel the registration of an endpoint by sending an Unregister Request (URQ) message to the endpoint. The endpoint shall respond with an Unregister Confirmation (UCF) message. The endpoint shall attempt to re-register with a Gatekeeper prior to initiating any calls. This may require the endpoint to register with a new Gatekeeper.

[End Correction]

6.1.11 BRQ/BRJ/BCF Exchange

Description:	Inconsistencies were found between the H.323 text and the H.225.0 text relating to the BRQ/BRJ/BCF exchange. Table 18 and section 7.12 of H.225.0 suggested that an endpoint may return a BRJ message, whereas H.323 did not allow this possibility. The text below shows the changes that

	shall be applied to H.323.

[Begin Correction]

8.4.1 Bandwidth changes

...

A Gatekeeper wishing to change the transmitted bit rate of Endpoint 1 sends a BRQ message to Endpoint 1. If the request is for a decrease in bit rate and the endpoint has the ability to support the requested bit rate, Endpoint 1 shall ~~always~~ comply by reducing its aggregate bit rate and returning a BCF. If Endpoint 1 cannot support the requested bit rate, the endpoint may return a BRJ. Endpoint 1 may initiate the appropriate H.245 signalling to inform Endpoint 2 that bit rates have changed. This will allow Endpoint 2 to inform its Gatekeeper of the change. If the request is for an increase, the endpoint may increase its bit rate when desired and allowed by the Gatekeeper.

If the Gatekeeper wishes to increase the bandwidth used by the endpoint, the endpoint may return a BCF to indicate acceptance of the new higher bit rate or a BRJ to indicate that it rejects the additional bandwidth. The endpoint should only accept the higher bit rate if the endpoint is prepared to utilize the additional bandwidth.

[End Correction]

6.1.12 Empty fastStart Element and Usage of the Facility Message for fastStart

Description:	There has been some confusion over semantics of an empty fastStart element. It was never the intent that an empty fastStart element could or should be used. In addition, it is illegal for an entity to send two Call Proceeding messages to a calling entity, according to Q.931. When a fastStart element is received in a Call Proceeding message after an signaling entity (such as a routed Gatekeeper) as already sent a Call Proceeding message, a Facility message shall be used to carry the fastStart data. This text is added to clarify these points.

[Begin Correction]

8.1.7 Fast Connect Procedure

...

The calling endpoint may begin transmitting media (according to the channels opened) immediately upon receiving a Q.931 message containing **fastStart**. Thus, the called endpoint must be prepared to immediately receive media on the channels it accepted in the Q.931 message containing **fastStart**. Note that national requirements may prohibit calling endpoints from transmitting media prior to receipt of a Connect message; it is the responsibility of the endpoint to comply with applicable requirements.

Note – An entity shall not send an empty **fastStart** element in any message (i.e., a **fastStart** element shall contain at least one **OpenLogicalChannel** proposal). If an endpoint does receive a **fastStart** element that contains no **OpenLogicalChannel** proposals, it shall ignore the **fastStart** element.

Note – When an endpoint or a gatekeeper intervening in call signalling receives a **fastStart** element in a Call Proceeding message, it will not be able to relay the Call Proceeding if the Call Proceeding message has already been sent to the originating side. In that case, the **fastStart** element in the Call Proceeding message shall be mapped to a **fastStart** element in a Facility message.

[End Correction]

6.1.13 perCallInfo in an IRR

Description:	There has been some confusion over when an endpoint shall include the perCallInfo sequence in an IRR message. This text clarifies that issue.

[Begin Correction]

8.4.2 Status

In order for the Gatekeeper to determine if an endpoint is turned off, or has otherwise entered a failure mode, the Gatekeeper may use the Information Request (IRQ)/Information Request Response (IRR) message sequence (see Recommendation H.225.0) to poll the endpoints at an interval decided by the manufacturer. The polling interval shall be greater than 10 s. This message may also be used by a diagnostic device as described in 11.2.

When an endpoint transmits an IRR, it shall include the **perCallInfo** field in order to provide details about calls to the Gatekeeper. If the Gatekeeper sends an IRQ requesting information for all calls and no calls are active or for a single call that is no longer active or for which the endpoint has no information, the endpoint shall omit the **perCallInfo** field from the IRR.

[End Correction]

6.1.14 Misleading "Call Proceeding Messages"

Description:	Text in section 8.1.8.1 suggests that multiple Call Proceeding messages may be sent to a calling entity for a single call. This is illegal, in fact. This text attempts to clarify the offending text.

[Begin Correction]

8.1.8.1 Gateway in-bound call setup

When an external terminal calls a network endpoint via the Gateway, call setup between the Gateway and the network endpoint proceeds the same as the endpoint-to-endpoint call setup. The Gateway may need to issue a Call Proceeding message~~s~~ to the external terminal while establishing the call on the network.

[End Correction]

6.1.15 Usage of Facility or Progress in place of Call Proceeding

Description:	As mentioned in the previous section, it is illegal to send multiple Call Proceeding messages in a call. However, a Call Proceeding may be

	generated locally and then one may be received from the remote endpoint at some later point in time. Any information in that message may be carried in a Facility message or Progress message, as appropriate. This text tries to clarify that point.

[Begin Correction]

8.7 Intermediate signalling entities

Intermediate entities in the signalling path, such as Gatekeepers that route call signalling, use the Facility message or the Progress message to convey any new information (such as Q.931 information elements, CallProceeding-UUIE fields, and encapsulated H.245 messages) received in a Call Proceeding message to the other endpoint if the entity has already sent a Call Proceeding message. This will allow the entity, for example, to transmit the **fastStart** element to facilitate proper establishment of a Fast Connect call and/or a Progress Indicator to indicate the presence of in-band tones and announcements.

[End Correction]

6.1.16 Dynamically Indicating Support for multipleCalls

Description:	It is possible to dynamically indicate that an endpoint can support multiple calls over the Call Signaling Channel. This text is introduced to clarify the procedure used to accomplish this task.

[Begin Correction]

7.3 Call signalling channel

...

The Call Signalling Channel may carry signalling for many concurrent calls, using the Call Reference Value to associate the message with the call. An entity indicates its ability to handle multiple concurrent calls on the same call signalling connection by setting the **multipleCalls** flag to TRUE in messages that it sends on the Call Signalling Channel. An entity may dynamically set the value of the **multipleCalls** field in order to indicate its present ability to support multiple connections along the Call Signalling Channel. If an endpoint wishes to change the value of **multipleCalls** at a time when no other H.225.0 messages are being exchanged across the Call Signalling Channel, it shall transmit the **multipleCalls** field via a Facility message with the CRV set to the Global Call Reference as shown in Figure 4-5/Q.931.

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.

...

[End Correction]

6.1.17 Tunneling non-H.323 protocols in an H.323 call

Description:	H.323v4 introduces a new feature that allows entities to carry QSIG, ISUP, or other protocols in the H.225.0 call signaling channel. However, it may be desirable to allow H.323v2 and H.323v3 entities to also support tunneling. The following text describes how an H.323v2 or H.323v3 entity may provide tunneling facilities. Note that this text is for informational purposes only as it does not define a standard means of tunneling: it utilized the non-standard fields that currently exist and merely point out that such usage is possible, if desirable.

Tunneling support in H.323 version 2 and H.323 version 3 entities

H.323 Version 2 and H.323 Version 3 had no defined procedures for tunneling. However, equipment manufacturers may desire to provide some support for tunneling non-H.323 signaling protocols within these older versions of H.323. To do so, the **nonStandardControl** field in any H.225.0 call signalling message may be used to pass non-H.323 protocols. The **object** field shall be selected as the type of **nonStandardIdentifier** and shall be set to the OBJECT IDENTIFIER of the protocol that is to be tunneled and the **data** field shall contain the actual tunneled message. Note, however, that there are no defined procedures for indicating support for tunneled protocols; therefore, tunneling support shall be considered optional in older H.323 entities. The decision to use or not use tunneling in older H.323 entities shall be addressed through equipment provisioning.

H.323 version 4 and higher entities shall utilize the procedures defined in 10.4.1 through 10.4.4 when tunneling is desired and when communicating with other H.323 version 4 or higher entities.

6.1.18 Alternate Transport Addresses

Description:	The ASN.1 in H.225.0 relating to the usage of Annex E/H.323 was found to be in error. The correction not only corrected the problem, but also expanded the scope of the field, as there is a definite need to indicate an expanded list of alternate transport mechanisms for H.323 call signaling. The below text describes the usage of those fields and will be included in the next published version of H.323

7.2.9 Alternate transport addresses

An endpoint may indicate support for alternate transport protocols by providing the alternateTransportAddresses field in the RRQ message. The Gatekeeper may instruct the endpoint as to which signalling transport protocol to use for making calls by including the useSpecifiedTransport field in the RCF or ACF message. The Gatekeeper shall include in the useSpecifiedTransport field only those protocols for which the endpoint has indicated its support. The endpoint, upon receipt of the useSpecifiedTransport field, shall use the specified transport to establish the call.

The Gatekeeper may give the endpoint a choice of transport protocols to use for call signaling by including the alternateTransportAddresses field in the RCF or ACF message without including the useSpecifiedTransport field. In this case the endpoint shall either use the protocol specified in the destCallSignalAddress field or select among the transports indicated in the alternateTransportAddresses field.

The Gatekeeper may also provide the alternateTransportAddresses of and endpoint registered with it to an H.323 entity in an LCF message.

6.1.19 Intermediate Signaling Entities

Description:	One issue that has caused confusion among implementers is the one of signaling the proper protocol version when calls are routed through an intermediate signaling entity. This text is intended to clarify the procedure.

[Begin Correction]

8.7 Intermediate signalling entities

...

Since some features in the H.323, such as third party pause and rerouting, require that the signalling entities know exactly what version of the protocol is being used by the other entities in a call and because the **protocolIdentifier** may change after receiving the first call signalling message and at other times during the call, such as when a call is rerouted to a different entity, entities that rely on version-specific features should determine the version of the other entities in a call by examining the **protocolIdentifier** in the Setup and Connect message at the very least. During a call, a call may be rerouted to a different entity that uses a different version of the protocol. In such a case, entities that rely on version-specific features should again determine the version of the entity to which the call may have been switched. If H.245 signalling is tunneled, the endpoint may use the call signalling message containing the tunneled non-empty terminal capability set message in order to determine the version of the remote endpoint. If a separate H.245 Channel is used, an entity may send a Status Inquiry message and determine the protocol version by examining the **protocolIdentifier** in the resulting Status message. In either case, the version of H.245 used by the other entity is signalled in the non-empty capability set message.

An intermediate signalling entity may signal its own protocol version when replying to a Setup message (e.g., to send a Call Proceeding message prior to establishing communication with the called party) or when initiating an outbound connection and may continue to signal its protocol version number if it has the ability to properly handle messages sent between entities using different versions of the protocol. In such cases where the intermediate signalling entity cannot properly handle messages between entities that use different versions of the protocol, it shall report to each of the two entities in the call the minimum value of its protocol version and the version reported by the other entity. In this way, all entities in the call will know what version-specific features are supported by every entity in the call signalling path.

[End Correction]

6.1.20 Re-routing a Fast Connect Initiated Call

Description:	Questions have arisen regarding the procedure a Gatekeeper should follow in order to re-route a call that was initiated as a Fast Connect call. The text below is intended to clarify that procedure.

[Begin Correction]

8.1.7.5 Third party re-routing of a fast connect initiated call

It is recommended that if an intermediate signalling entity wishes to re-route a Fast Connect initiated call for which it has already sent the **fastStart** or **fastConnectRefused** element to the calling endpoint, it shall open the H.245 Channel with the calling endpoint and put the endpoint into a paused state as described in 8.4.6 prior to re-routing the call. Additionally, the intermediate signalling entity shall not use Fast Connect for the re-routed call leg. Note, however, that if the intermediate signalling entity has not sent a **fastStart** or **fastConnectRefused** element to the calling endpoint and has not opened the H.245 Channel with the calling endpoint, it may re-route the call without any additional signalling to the calling endpoint and may use Fast Connect for the re-routed call leg.

[End Correction]

6.1.21 Fast Connect and H.245 Signaling Issues

Description:	Questions have arisen regarding the procedure a Gatekeeper should follow in order to re-route a call that was initiated as a Fast Connect call. The text below is intended to clarify that procedure.

[Begin Correction]

8.1.7.2 Switching to H.245 procedures

...

If the calling endpoint utilizes Fast Connect to initiate a call, it shall not initiate H.245 until the called endpoint has returned **fastStart, fastConnectRefused, h245Address,** or the Connect message. Note that a calling H.323 Version 2 endpoint may start H.245 tunneling even before one of these conditions if it chooses, in spite of the fact that it initiated a Fast Connect call. While this behavior is strongly discouraged in H.323 Version 2 systems, Version 3 and newer endpoints needs to be aware of this behavior. In addition, if an H.323 Version 2 endpoint initiates H.245 in this manner, the Version 3 or newer endpoint shall assume that Fast Connect is terminated and shall not send a **fastStart** element.

The called endpoint shall not initiate H.245 before returning **fastConnectRefused** or **fastStart**. A called endpoint that returns the **h245Address** element in any message up to and including the Connect message, and which has not already explicitly accepted or rejected Fast Connect, shall also return either **fastStart** or **fastConnectRefused** in the same message. An H.323 endpoint prior to Version 3 may not return **fastStart** or **FastConnectRefused**. For backward compatibility with older endpoints, H.323 endpoints may assume that Fast Connect is refused, irrespective of the protocol version of the called endpoint, if the called endpoint sends the **h245Address** element without also sending **fastStart** or **fastConnectRefused** in the same or previous message as the message containing the **h245Address.**

Note that a race condition exists in the case where a separate H.245 connection is used to initiate H.245 from the called endpoint to a calling endpoint that supplied its **h245Address** in the Setup message. For this reason, it is recommended that if an endpoint accepts Fast

Connect and initiates H.245 in parallel, it should introduce a delay between sending the H.2250.0 message containing the **fastStart** element and the initiation of the separate H.245 connection. Endpoint should be prepared for a late arrival of the **fastStart** element in this scenario. H.323 version 2 endpoint will assume that Fast Connect is refused if the H.245 Channel is opened prior to receive the **fastStart** element.

[End Correction]

6.1.22 Enforcing symmetric codec operation

Description:	Implementers have been confused over the meaning of "receive and transmit" capabilities and the reality in the market is that many DSPs require symmetric codec operation. For this reason, the following additions are added to H.323v4.

[Begin Correction]

6.2.8.1 Capabilities exchange

...

Receive-and-Transmit capabilities describe the terminal's ability to receive and transmit information streams when these capabilities are not independent and are required to be the same in both directions. For example, an endpoint might support only symmetrical codec operation for its codecs (G.711 both ways, or G.729 both ways, but not G.711 one way or G.729 the other way). A slave should reorder its codec preference in the same order as the master, e.g., if the slave's preference is {G.729, G.711} and the master's preference is {G.711, G.729}, the slave should reorder its preference to {G.711, G.729}. If the slave has not already sent its terminal capability set yet, it should send the reordered set. If the terminal capability set has already proceeded, it should consider its preferences as reordered when proceeding to opening logical channels.

...

When symmetrical codec operation is used (i.e., when the **receiveAndTransmitVideoCapability** or **receiveAndTransmitAudioCapability** are used), the master may reject an **openLogicalChannel** request from the slave if the master requires the user of symmetrical codecs and the proposed channel is not symmetrical. These conflict resolution procedures are described in H.245/C.4.1.3. The reason field in the **openLogicalChannelReject** shall be **masterSlaveConflict**.

Note – The master may send a **requestMode** to the slave with the proper codec before sending the **openLogicalChannelReject** to explicitly request a specific codec.

[End Correction]

6.2 Technical and Editorial Corrections to ITU-T Recommendation H.225.0 (1999)

6.2.1 Usage of the XRS Message

Description:	A technical problem was discovered with the text in all versions of the H.225.0 document relating to the XRS message, including H.225.0 (1999).

	The text stated that if a RAS message was not understood, an XRS message is returned with the RequestSequenceNum set to zero. However, zero is an invalid value, as the range for that field is 1 to 65535.

[Begin Correction]

7.17 Message Not Understood

This message is sent whenever an H.323 endpoint receives a RAS message it can decode, but does not understand. If a RAS message cannot be decoded it should be ignored.

RequestSeqNum – Shall be the **requestSeqNum** of the unknown message~~, if it can be decoded, and zero otherwise~~.

...

[End Correction]

6.2.2 Packetization of the G.722.1 bit stream for use with the Real Time Protocol (RTP)

Description:	The following text has been added to describe the packetization of G.722.1 audio. The text will appear in a new section of Annex F of H.225.0.

[Begin Correction]

F.6 G.722.1

For information on the packetization of G.722.1 bit stream, refer to Annex A/G.722.1.

[End Correction]

6.2.3 Packetization of G.722.1

Description:	The following text has been added to describe the packetization of G.722.1 audio. The text will appear in Annex B of H.225.0.

[Begin Correction]

Table B.1/H.225.0 – Properties of audio encodings

Encoding	Sample/frame	Bits/sample	ms/frame
G722	Sample	8	
G722.1	Frame	N/A	20
G728	Frame	N/A	2.5
PCMA	Sample	8	
PCMU	Sample	8	
G723	Frame	N/A	30
G729	Frame	N/A	10
GSM	Frame	N/A	20

[End Correction]

[Begin Correction]

Table B.2/H.225.0 – Payload Types (PT) for standard audio and video encodings

PT	Encoding name	Audio/video (A/V)	Clock rate (Hz)	Channels (audio)
0	PCMU	A	8 000	1
8	PCMA	A	8 000	1
9	G722	A	8 000	1
Dynamic	G722.1	A	16 000	1
4	G723	A	8 000	1
15	G728	A	8 000	1
18	G729	A	8 000	1
31	H261	V	90 000	
34	H263	V	90 000	
3	GSM	A	8 000	1
96-127	Dynamic	?		
NOTE – Payload types 1-7, 10-14, 16-30, and 30-95 are reserved. See Appendix II for more information.				

[End Correction]

6.2.4 Correction to Values in Table 12/H.225.0

Description:	An error was pointed out in the length value for the cause IE in the Release Complete message. The following text shows the correct changes to Table 12/H.225.0.

[Begin Correction]

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
Cause	CM (Note)	~~1~~2-32
Facility	O	8-*
Notification indicator	O	2-*
Display	O	2-82
Signal	O	2-3
User-to-User	M	2-131
NOTE – Either the Cause IE or the **ReleaseCompleteReason** shall be present.		

[End Correction]

6.2.5 Support for New Annexes in G.729

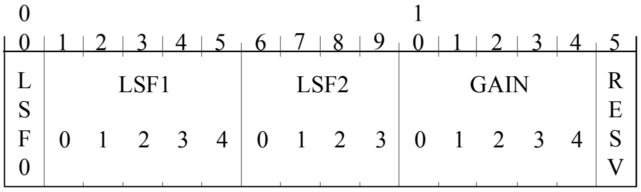
Description:	The following text shall be inserted into in Annex F/H.225.0 to support new annexes to G.729.

[Begin Correction]

F.3 G.729

...

A Voice Activity Detector (VAD) and Comfort Noise Generator (CNG) algorithm in Annex B/G.729 is recommended~~ for digital simultaneous voice and data applications and can be used in conjunction with Recommendation G.729 or Annex A/G.729~~. This algorithm is applied to Annexes F/G.729 (6.4 Kbps with VAD/CNG) and G/G.729 (11.8 Kbps with VAD/CNG), and Annex B/G.729 (G.729 and Annex A/G.729 with VAD/CNG), Annex I/G.729. A G.729 or Annex A/G.729 frame contains 10 octets, Annex D/G.729 and Annex E/G.279 contain 8 and 15 octets, respectively, while the Annexes B/G.729, F/G.729, and G/G.729 comfort noise frame occupies 2 octets, as shown in Figure F.3:



RESV = Reserved (zero)

T1529860-98

Figure F.3 – Annexes B/G.729, F/G.729, and G/G.729 CNG packetization format

...

[End Correction]

6.2.6 Clarification of Alternate Gatekeeper Procedures

Description:	Ambiguities have been identified in the procedures for "alternate gatekeepers". This section specifies changes to be applied to H.225.0 to clarify the procedures.

[Begin Correction]

7.6 H.225.0 common message elements

...

A gatekeeper may send an endpoint a list of alternate gatekeepers in various messages. When communicating with its gatekeeper, an endpoint that implements the alternate gatekeeper mechanism shall replace any previously received list of alternate gatekeepers with the most recently received list of alternate gatekeepers. It is possible for an alternate gatekeeper to send a list of alternate gatekeepers. If an endpoint sends a request to an alternate gatekeeper that will potentially become its permanent gatekeeper, it shall accept the new list of alternate gatekeepers. Otherwise, if the alternate gatekeeper will not potentially become its permanent gatekeeper, any list of alternate gatekeepers received shall be ignored. A gatekeeper may potentially become an endpoint’s permanent gatekeeper if either the current gatekeeper becomes unresponsive or if the "altGKisPermanent" flag is set to TRUE in the "AltGKInfo" structure.

If the Gatekeeper wishes to clear the endpoint's list of Alternate Gatekeepers, such as when the Gatekeeper is reconfigured to not use Alternate Gatekeepers, it shall return an empty list of Alternate Gatekeepers to the endpoint in the RCF message.

When an endpoint is redirected to a new permanent alternate Gatekeeper, either as a result of the current gatekeeper becoming unresponsive or by receiving an explicit redirection message (xRJ), the endpoint shall not send a URQ to its current Gatekeeper. Additionally, all subsequent requests, including those for existing calls, shall be directed to the new permanent gatekeeper.

If the endpoint's gatekeeper becomes unresponsive and either no alternate gatekeeper list was provided or all alternate gatekeepers are also unresponsive, the endpoint shall attempt to discover a new gatekeeper and register with it according to the procedures defined in H.323. Note that procedures for handling existing calls in this scenario are for further study.

...

[End Correction]

The following correction shall be applied to sections 7.8.3, 7.9.3, 7.10.3, 7.11.3, 7.12.3, 7.13.3, 7.14.3, and 7.15.4.

[Begin Correction]

altGKInfo – Optional information about alternative gatekeepers. If this information is supplied, an endpoint should retransmit the request to one of the alternate gatekeepers listed. If an alternate gatekeeper rejects the request without supplying alternate gatekeeper information, the endpoint shall accept the rejection. If an alternate gatekeeper does not respond or returns a rejection with alternate gatekeeper information, the endpoint may send the request to another alternate~~ in the list~~. (Refer to section 7.6 for the specific procedures on handling multiple lists of alternate gatekeepers.)

[End Correction]

In addition to the changes specified above, the following sections shall also contain these additional amendments.

[Begin Correction]

7.8.3 GatekeeperReject (GRJ)

...

altGKInfo – Optional information about alternative gatekeepers. If this information is supplied, an endpoint should retransmit the request to one of the alternate gatekeepers listed. If an alternate gatekeeper rejects the request, the endpoint shall accept the rejection. If an alternate gatekeeper does not respond, the endpoint may send the request to another alternate in the list. For this message, endpoints shall ignore the actual **altGKisPermanent** flag and the **needToRegister** flags in the **AlternateGK** sequence and assume the values are TRUE.

...

[End Correction]

[Begin Correction]

7.9.3 RegistrationReject (RRJ)

...

altGKInfo – Optional information about alternative gatekeepers. If this information is supplied, an endpoint should retransmit the request to one of the alternate gatekeepers listed. If an alternate gatekeeper rejects the request, the endpoint shall accept the rejection. If an alternate gatekeeper does not respond, the endpoint may send the request to another alternate in the list. If an endpoint has not yet successfully registered with a gatekeeper, the endpoint shall ignore the actual **needToRegister** flags in the **AlternateGK** sequence and assume the values are TRUE.

...

[End Correction]

A comment has been added to the **AltGKInfo** sequence to explain the usage of the **altGKisPermanent** field. Refer to the ASN.1 revisions in section 6.4.9 for this text.

6.2.7 Usage of Keypad Facility IE

Description:	SET devices (Annex F/H.323) shall support transmission of DTMF as Keypad Information Elements in the H.225.0 call signaling connection (e.g. using Information messages). However, there is no established method for carrying a hookflash indication in this information element. H.225.0 shall be modified as described below to allow the hookflash indication to be transmitted.

[Begin Correction]

7.2.2.16 Keypad facility

Encoded following Figure 4-24/Q.931. The use of the exclamation point character "!" shall represent a hookflash indication. Endpoints not supporting reception of the hookflash indication shall ignore the "!" if received.

[End Correction]

6.2.8 Order of Information Elements in H.225.0 Call Signalling Messages

Description:	Ambiguities have been identified with respect to the ordering of Information Elements in H.225.0 Call Signaling Messages. Table 8/H.225.0 suggests an ordering of information elements that is inconsistent with Q.931. That was not intended as the ordering of information elements is specified in Q.931. The table and text below will appear in the next revision of H.225.0.

[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
~~High layer compatibility~~	~~FFS~~	~~NA~~
Low layer compatibility	FFS	NA
High layer compatibility	FFS	NA
User-to-User	M	2-131
~~Connected Number~~	~~O~~	~~2-*~~
~~Connected Sub-Address~~	~~O~~	~~2-23~~
NOTE – Bearer capability is mandatory if the message is between a terminal and a gateway.		

[End Correction]

[Begin Correction]

7.1 Use of Q.931 messages

...

Each H.225.0 endpoint shall be able to interpret and generate the information elements mandated in the following for the respective Q.931 and H.450 messages. It may interpret and generate the optional information elements as defined below as well. It also may interpret other information elements of Q.931, or other Q series or H.450 protocols. The endpoints shall be able to ignore unknown information elements contained in a Q.931 or H.450 message without disturbing operation. Procedures for receiving unrecognized "comprehension required" information elements shall apply according to 5.8.7.1/Q.931.

Information Elements shall be encoded according to Q.931, except where modified in this Recommendation. However, Q.931 shall always dictate the proper ordering of Information Elements within a message, regardless of the order of elements listed within this Recommendation.

...

[End Correction]

6.2.9 Changes to the H.225.0 ASN.1

Description:	This section details the changes to the published ASN.1 for H.225.0.

[Begin Correction]

```
H323-UU-PDU ::= SEQUENCE
{
        h323-message-body   CHOICE
        {
                setup               Setup-UUIE,
                callProceeding      CallProceeding-UUIE,
                connect             Connect-UUIE,
                alerting            Alerting-UUIE,
                information         Information-UUIE,
                releaseComplete     ReleaseComplete-UUIE,
                facility            Facility-UUIE,
                ...,
                progress            Progress-UUIE,
                empty               NULL          -- used when a FACILITY message is sent,
                                                  -- but the Facility-UUIE is not to be invoked
                                                  -- (possible when transporting supplementary
                                                  -- services messages)
        },
        nonStandardData             NonStandardParameter OPTIONAL,
        ...,
        h4501SupplementaryService           SEQUENCE OF OCTET STRING OPTIONAL,
                                            -- each sequence of octet string is defined as one
                                            -- H4501SupplementaryService APDU as defined in
                                            -- Table 3/H.450.1

        h245Tunneling               BOOLEAN,
                                            -- if TRUE, tunneling of H.245 messages is enabled
        h245Control                 SEQUENCE OF OCTET STRING OPTIONAL,
                                            -- each octet string may contain exactly
                                            -- one H.245 PDU
        nonStandardControl          SEQUENCE OF NonStandardParameter OPTIONAL,
        callLinkage                 CallLinkage OPTIONAL
}

Alerting-UUIE ::= SEQUENCE
{
        protocolIdentifier          ProtocolIdentifier,
        destinationInfo             EndpointType,
        h245Address                 TransportAddress OPTIONAL,
        ...,
        callIdentifier              CallIdentifier,
        h245SecurityMode            H245Security OPTIONAL,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
        fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
        multipleCalls                       BOOLEAN,
        maintainConnection                  BOOLEAN,
        alertingAddress             SEQUENCE OF AliasAddress OPTIONAL,
        presentationIndicator       PresentationIndicator OPTIONAL,
        screeningIndicator          ScreeningIndicator OPTIONAL,
        fastConnectRefused          NULL OPTIONAL
}

CallProceeding-UUIE ::= SEQUENCE
{
        protocolIdentifier          ProtocolIdentifier,
        destinationInfo             EndpointType,
        h245Address                 TransportAddress OPTIONAL,
        ...,
        callIdentifier              CallIdentifier,
        h245SecurityMode            H245Security OPTIONAL,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
        fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
        multipleCalls                       BOOLEAN,
        maintainConnection                  BOOLEAN,
        fastConnectRefused                  NULL OPTIONAL
}

Connect-UUIE ::= SEQUENCE
{
        protocolIdentifier          ProtocolIdentifier,
        h245Address                 TransportAddress OPTIONAL,
        destinationInfo             EndpointType,
        conferenceID                        ConferenceIdentifier,
        ...,
        callIdentifier              CallIdentifier,
        h245SecurityMode            H245Security OPTIONAL,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
        fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
        multipleCalls                       BOOLEAN,
        maintainConnection                  BOOLEAN,
        language                    SEQUENCE OF IA5String(SIZE (1..32)) OPTIONAL,
                                                        -- RFC1766 language tag
        connectedAddress            SEQUENCE OF AliasAddress OPTIONAL,
        presentationIndicator       PresentationIndicator OPTIONAL,
        screeningIndicator          ScreeningIndicator OPTIONAL,
        fastConnectRefused                  NULL OPTIONAL
}

Information-UUIE ::=SEQUENCE
{
        protocolIdentifier  ProtocolIdentifier,
        ...,
        callIdentifier              CallIdentifier,
        tokens              SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
        fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
        fastConnectRefused                  NULL OPTIONAL
}

ReleaseComplete-UUIE ::= SEQUENCE
{
        protocolIdentifier          ProtocolIdentifier,
        reason                      ReleaseCompleteReason OPTIONAL,
        ...,
        callIdentifier              CallIdentifier,
        tokens              SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
        busyAddress                 SEQUENCE OF AliasAddress OPTIONAL,
        presentationIndicator       PresentationIndicator OPTIONAL,
        screeningIndicator          ScreeningIndicator OPTIONAL
}

Setup-UUIE ::= SEQUENCE
{
        protocolIdentifier          ProtocolIdentifier,
        h245Address                 TransportAddress OPTIONAL,
        sourceAddress               SEQUENCE OF AliasAddress OPTIONAL,
        sourceInfo                  EndpointType,
        destinationAddress          SEQUENCE OF AliasAddress OPTIONAL,
        destCallSignalAddress       TransportAddress OPTIONAL,
        destExtraCallInfo           SEQUENCE OF AliasAddress OPTIONAL,          -- Note 1
        destExtraCRV                SEQUENCE OF CallReferenceValue OPTIONAL,    -- Note 1
        activeMC                    BOOLEAN,
        conferenceID                        ConferenceIdentifier,
        conferenceGoal              CHOICE
        {
                create                          NULL,
                join                            NULL,
                invite                          NULL,
                ...,
                capability-negotiation          NULL,
                callIndependentSupplementaryService   NULL
        },
        callServices                QseriesOptions  OPTIONAL,
        callType                    CallType,
        ...,
        sourceCallSignalAddress     TransportAddress OPTIONAL,
        remoteExtensionAddress      AliasAddress OPTIONAL,
        callIdentifier              CallIdentifier,
        h245SecurityCapability      SEQUENCE OF H245Security OPTIONAL,
        tokens              SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
        fastStart                   SEQUENCE OF OCTET STRING OPTIONAL,
        mediaWaitForConnect         BOOLEAN,
        canOverlapSend              BOOLEAN,
        endpointIdentifier          EndpointIdentifier OPTIONAL,
        multipleCalls                       BOOLEAN,
        maintainConnection                  BOOLEAN,
        connectionParameters        SEQUENCE -- additional gateway parameters
        {
                connectionType                  ScnConnectionType,
                numberOfScnConnections          INTEGER (0..65535),
                connectionAggregation       ScnConnectionAggregation,
                ...
        } OPTIONAL,
        language                SEQUENCE OF IA5String(SIZE (1..32)) OPTIONAL,
                                                        -- RFC1766 language tag
        presentationIndicator   PresentationIndicator OPTIONAL,
        screeningIndicator      ScreeningIndicator OPTIONAL
}

Facility-UUIE ::= SEQUENCE
{
        protocolIdentifier      ProtocolIdentifier,
        alternativeAddress      TransportAddress OPTIONAL,
        alternativeAliasAddress SEQUENCE OF AliasAddress OPTIONAL,
        conferenceID                    ConferenceIdentifier OPTIONAL,
        reason                  FacilityReason,
        ...,
        callIdentifier          CallIdentifier,
        destExtraCallInfo       SEQUENCE OF AliasAddress OPTIONAL,
        remoteExtensionAddress  AliasAddress OPTIONAL,
        tokens          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
        conferences             SEQUENCE OF ConferenceList OPTIONAL,
        h245Address             TransportAddress OPTIONAL,
        fastStart               SEQUENCE OF OCTET STRING OPTIONAL,
        multipleCalls                   BOOLEAN,
        maintainConnection              BOOLEAN,
        fastConnectRefused              NULL OPTIONAL
}

Progress-UUIE ::= SEQUENCE
{
        protocolIdentifier      ProtocolIdentifier,
        destinationInfo         EndpointType,
        h245Address             TransportAddress OPTIONAL,
        callIdentifier          CallIdentifier,
        h245SecurityMode        H245Security OPTIONAL,
        tokens          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
        fastStart               SEQUENCE OF OCTET STRING OPTIONAL,
        ...,
        multipleCalls                   BOOLEAN,
        maintainConnection              BOOLEAN,
        fastConnectRefused              NULL OPTIONAL
}

CallLinkage ::= SEQUENCE
{
        globalCallId            GloballyUniqueID  OPTIONAL,
        threadId                GloballyUniqueID  OPTIONAL,
        ...
}

Endpoint ::= SEQUENCE
{
        nonStandardData         NonStandardParameter OPTIONAL,
        aliasAddress            SEQUENCE OF AliasAddress OPTIONAL,
        callSignalAddress       SEQUENCE OF TransportAddress OPTIONAL,
        rasAddress                  SEQUENCE OF TransportAddress OPTIONAL,
        endpointType                        EndpointType OPTIONAL,
        tokens              SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
        priority                    INTEGER(0..127) OPTIONAL,
        remoteExtensionAddress  SEQUENCE OF AliasAddress OPTIONAL,
        destExtraCallInfo           SEQUENCE OF AliasAddress OPTIONAL,
        ...,
        alternateTransportAddresses     AlternateTransportAddresses OPTIONAL
}

AlternateTransportAddresses ::= SEQUENCE
{
        annexE                      SEQUENCE OF TransportAddress OPTIONAL,
        ...
}

UseSpecifiedTransport ::= CHOICE
{
        tcp                         NULL,
        annexE                      NULL,
        ...
}

AltGKInfo ::=SEQUENCE
{
        alternateGatekeeper                 SEQUENCE OF AlternateGK,
        altGKisPermanent            BOOLEAN,
                        -- It is illegal to set this flag to FALSE and to set the
                        -- "needToRegister" flag inside an AlternateGK structure to TRUE.
        ...
}

RegistrationRequest ::= SEQUENCE --(RRQ)
{
        requestSeqNum                       RequestSeqNum,
        protocolIdentifier                  ProtocolIdentifier,
        nonStandardData                     NonStandardParameter OPTIONAL,
        discoveryComplete                   BOOLEAN,
        callSignalAddress                   SEQUENCE OF TransportAddress,
        rasAddress                          SEQUENCE OF TransportAddress,
        terminalType                                EndpointType,
        terminalAlias                       SEQUENCE OF AliasAddress OPTIONAL,
        gatekeeperIdentifier                        GatekeeperIdentifier  OPTIONAL,
        endpointVendor                      VendorIdentifier,
        ...,
        alternateEndpoints                  SEQUENCE OF Endpoint OPTIONAL,
        timeToLive                          TimeToLive OPTIONAL,
        tokens                      SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                        SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue                 ICV OPTIONAL,
        keepAlive                           BOOLEAN,
        endpointIdentifier                  EndpointIdentifier OPTIONAL,
        willSupplyUUIEs                     BOOLEAN,
        maintainConnection                          BOOLEAN,
        ~~supportsAnnexECallSignalling  BOOLEAN,~~
        alternateTransportAddresses         AlternateTransportAddresses OPTIONAL
}
```

```
RegistrationConfirm ::= SEQUENCE --(RCF)
{
        requestSeqNum                   RequestSeqNum,
        protocolIdentifier              ProtocolIdentifier,
        nonStandardData                 NonStandardParameter OPTIONAL,
        callSignalAddress               SEQUENCE OF TransportAddress,
        terminalAlias                   SEQUENCE OF AliasAddress OPTIONAL,
        gatekeeperIdentifier                    GatekeeperIdentifier  OPTIONAL,
        endpointIdentifier              EndpointIdentifier,
        ...,
        alternateGatekeeper                     SEQUENCE OF AlternateGK OPTIONAL,
        timeToLive                      TimeToLive OPTIONAL,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ICV OPTIONAL,
        willRespondToIRR                BOOLEAN,
        preGrantedARQ                   SEQUENCE
        {
                makeCall                                BOOLEAN,
                useGKCallSignalAddressToMakeCall        BOOLEAN,
                answerCall                              BOOLEAN,
                useGKCallSignalAddressToAnswer          BOOLEAN,
                ...,
                irrFrequencyInCall                      INTEGER (1..65535) OPTIONAL,
                                                                -- in seconds; not
                                                                -- present if GK
                                                                -- does not want IRRs
                totalBandwidthRestriction                BandWidth OPTIONAL,
                                                                -- total limit for all
                                                                -- concurrent calls
~~              useAnnexECallSignalling                 BOOLEAN~~
                alternateTransportAddresses             AlternateTransportAddresses OPTIONAL,
                useSpecifiedTransport                   UseSpecifiedTransport OPTIONAL
        } OPTIONAL,
        maintainConnection              BOOLEAN
}

AdmissionRequest ::= SEQUENCE --(ARQ)
{
        requestSeqNum           RequestSeqNum,
        callType                CallType,
        callModel               CallModel OPTIONAL,
        endpointIdentifier      EndpointIdentifier,
        destinationInfo         SEQUENCE OF AliasAddress OPTIONAL,   -- Note 1
        destCallSignalAddress   TransportAddress OPTIONAL,           -- Note 1
        destExtraCallInfo       SEQUENCE OF AliasAddress OPTIONAL,
        srcInfo                 SEQUENCE OF AliasAddress,
        srcCallSignalAddress    TransportAddress OPTIONAL,
        bandWidth               BandWidth,
        callReferenceValue      CallReferenceValue,
        nonStandardData         NonStandardParameter OPTIONAL,
        callServices            QseriesOptions  OPTIONAL,
        conferenceID                    ConferenceIdentifier,
        activeMC                BOOLEAN,
        answerCall              BOOLEAN,  -- answering a call
        ...,
        canMapAlias                     BOOLEAN,  -- can handle alias address
        callIdentifier          CallIdentifier,
        srcAlternatives         SEQUENCE OF Endpoint OPTIONAL,
        destAlternatives              SEQUENCE OF Endpoint OPTIONAL,
        gatekeeperIdentifier                GatekeeperIdentifier OPTIONAL,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                  SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue           ICV OPTIONAL,
        transportQOS                  TransportQOS OPTIONAL,
        willSupplyUUIEs               BOOLEAN,
        callLinkage                   CallLinkage OPTIONAL
}

AdmissionConfirm ::= SEQUENCE --(ACF)
{
        requestSeqNum                 RequestSeqNum,
        bandWidth                     BandWidth,
        callModel                     CallModel,
        destCallSignalAddress         TransportAddress,
        irrFrequency                        INTEGER (1..65535) OPTIONAL,
        nonStandardData               NonStandardParameter OPTIONAL,
        ...,
        destinationInfo               SEQUENCE OF AliasAddress OPTIONAL,
        destExtraCallInfo             SEQUENCE OF AliasAddress OPTIONAL,
        destinationType               EndpointType OPTIONAL,
        remoteExtensionAddress        SEQUENCE OF AliasAddress OPTIONAL,
        alternateEndpoints            SEQUENCE OF Endpoint OPTIONAL,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                  SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue           ICV OPTIONAL,
        transportQOS                  TransportQOS OPTIONAL,
        willRespondToIRR              BOOLEAN,
        uuiesRequested                UUIEsRequested,
        language                      SEQUENCE OF IA5String(SIZE (1..32)) OPTIONAL,
                                                                -- RFC1766 language tag
        ~~useAnnexECallSignalling  BOOLEAN~~
        alternateTransportAddresses         AlternateTransportAddresses OPTIONAL,
        useSpecifiedTransport               UseSpecifiedTransport OPTIONAL
}

AdmissionReject ::= SEQUENCE --(ARJ)
{
        requestSeqNum                 RequestSeqNum,
        rejectReason                  AdmissionRejectReason,
        nonStandardData               NonStandardParameter OPTIONAL,
        ...,
        altGKInfo                     AltGKInfo OPTIONAL,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        ~~callSignalAddress             SEQUENCE OF TransportAddress OPTIONAL,~~
        cryptoTokens                  SEQUENCE OF CryptoH323Token OPTIONAL,
        callSignalAddress             SEQUENCE OF TransportAddress OPTIONAL,
        integrityCheckValue           ICV OPTIONAL
}

AdmissionRejectReason ::= CHOICE
{
        calledPartyNotRegistered NULL,          -- cannot translate address
        invalidPermission             NULL,          -- permission has expired
        requestDenied                 NULL,          -- no bandwidth available
        undefinedReason               NULL,
        callerNotRegistered           NULL,
        routeCallToGatekeeper         NULL,
	invalidEndpointIdentifier	NULL,
	resourceUnavailable		NULL,
	...,
	securityDenial			NULL,
	qosControlNotSupported	NULL,
	incompleteAddress		NULL,
~~	routeCallToSCN			SEQUENCE OF PartyNumber,~~
	aliasesInconsistent		NULL,		-- multiple aliases in request identify distinct people
	routeCallToSCN			SEQUENCE OF PartyNumber
}

BandwidthRequest ::= SEQUENCE --(BRQ)
{
	requestSeqNum			RequestSeqNum,
	endpointIdentifier		EndpointIdentifier,
	conferenceID				ConferenceIdentifier,
	callReferenceValue		CallReferenceValue,
	callType			CallType OPTIONAL,
	bandWidth			BandWidth,
	nonStandardData			NonStandardParameter OPTIONAL,
	...,
	callIdentifier			CallIdentifier,
	gatekeeperIdentifier			GatekeeperIdentifier OPTIONAL,
	tokens			SEQUENCE OF ClearToken OPTIONAL,
	cryptoTokens			SEQUENCE OF CryptoH323Token OPTIONAL,
	integrityCheckValue		ICV OPTIONAL,
	answeredCall				BOOLEAN,
	callLinkage			CallLinkage OPTIONAL
}

LocationConfirm ::= SEQUENCE --(LCF)
{
	requestSeqNum			RequestSeqNum,
	callSignalAddress		TransportAddress,
	rasAddress			TransportAddress,
	nonStandardData			NonStandardParameter OPTIONAL,
	...,
	destinationInfo			SEQUENCE OF AliasAddress OPTIONAL,
	destExtraCallInfo		SEQUENCE OF AliasAddress OPTIONAL,
	destinationType			EndpointType OPTIONAL,
	remoteExtensionAddress	SEQUENCE OF AliasAddress OPTIONAL,
	alternateEndpoints		SEQUENCE OF Endpoint OPTIONAL,
	tokens			SEQUENCE OF ClearToken OPTIONAL,
	cryptoTokens			SEQUENCE OF CryptoH323Token OPTIONAL,
	integrityCheckValue		ICV OPTIONAL,
~~	supportsAnnexECallSignalling	BOOLEAN~~
	alternateTransportAddresses	AlternateTransportAddresses OPTIONAL
}

LocationRejectReason ::= CHOICE
{
	notRegistered			NULL,
	invalidPermission		NULL,		-- exclusion by administrator or feature
	requestDenied			NULL,		-- cannot find location
	undefinedReason			NULL,
	...,
	securityDenial			NULL,
~~	routeCallToSCN			SEQUENCE OF PartyNumber,~~
	aliasesInconsistent		NULL,		-- multiple aliases in request identify distinct people
	routeCallToSCN		SEQUENCE OF PartyNumber,
}

DisengageRequest ::= SEQUENCE --(DRQ)
{
	requestSeqNum		RequestSeqNum,
	endpointIdentifier	EndpointIdentifier,
	conferenceID			ConferenceIdentifier,
	callReferenceValue	CallReferenceValue,
	disengageReason		DisengageReason,
	nonStandardData		NonStandardParameter OPTIONAL,
	...,
	callIdentifier		CallIdentifier,
	gatekeeperIdentifier		GatekeeperIdentifier OPTIONAL,
	tokens		SEQUENCE OF ClearToken OPTIONAL,
	cryptoTokens		SEQUENCE OF CryptoH323Token OPTIONAL,
	integrityCheckValue	ICV OPTIONAL,
	answeredCall			BOOLEAN,
	callLinkage		CallLinkage OPTIONAL
}

InfoRequest ::= SEQUENCE --(IRQ)
{
	requestSeqNum		RequestSeqNum,
	callReferenceValue	CallReferenceValue,
	nonStandardData		NonStandardParameter OPTIONAL,
	replyAddress			TransportAddress OPTIONAL,
	...,
	callIdentifier		CallIdentifier,
	tokens		SEQUENCE OF ClearToken OPTIONAL,
	cryptoTokens		SEQUENCE OF CryptoH323Token OPTIONAL,
	integrityCheckValue	ICV OPTIONAL,
	uuiesRequested		UUIEsRequested OPTIONAL,
	callLinkage		CallLinkage OPTIONAL
}

InfoRequestResponse ::= SEQUENCE --(IRR)
{
	nonStandardData		NonStandardParameter OPTIONAL,
	requestSeqNum		RequestSeqNum,
	endpointType			EndpointType,
	endpointIdentifier	EndpointIdentifier,
	rasAddress		TransportAddress,
	callSignalAddress	SEQUENCE OF TransportAddress,
	endpointAlias		SEQUENCE OF AliasAddress OPTIONAL,
	perCallInfo		SEQUENCE OF SEQUENCE
	{
		nonStandardData		NonStandardParameter OPTIONAL,
		callReferenceValue	CallReferenceValue,
		conferenceID			ConferenceIdentifier,
		originator		BOOLEAN OPTIONAL,
		audio			SEQUENCE OF RTPSession OPTIONAL,
		video			SEQUENCE OF RTPSession OPTIONAL,
		data			SEQUENCE OF TransportChannelInfo OPTIONAL,
		h245			TransportChannelInfo,
		callSignaling		TransportChannelInfo,
		callType		CallType,
		bandWidth		BandWidth,
		callModel		CallModel,
          ...,
          callIdentifier              CallIdentifier,
          tokens                SEQUENCE OF ClearToken OPTIONAL,
          cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
          substituteConfIDs           SEQUENCE OF ConferenceIdentifier,
          pdu                         SEQUENCE OF SEQUENCE
          {
               h323pdu          H323-UU-PDU,
               sent             BOOLEAN          -- TRUE is sent, FALSE is received
          } OPTIONAL,
          callLinkage                 CallLinkage OPTIONAL
     } OPTIONAL,
     ...,
     tokens                     SEQUENCE OF ClearToken OPTIONAL,
     cryptoTokens                     SEQUENCE OF CryptoH323Token OPTIONAL,
     integrityCheckValue              ICV OPTIONAL,
     needResponse                     BOOLEAN
}
```

[End Correction]

6.2.10 Call Linkage

Description:	A description for the new CallLinkage fields found ARQ, BRQ, DRQ, IRQ, and IRR messages is defined below.

[Begin Correction]

CallLinkage – The contents of this field is typically controlled by a call linkage service. For the procedures and semantics of this field refer to H.323 section 10.3 “Call Linkage in H.323”.

[End Correction]

6.2.11 Missing Field Descriptions

Description:	It was pointed out that there were some field descriptions missing for some of the H323-UU-PDU elements in H.225.0. Below is the text for those descriptions.

[Begin Correction]

nonStandardData – This field carries information not defined in this Recommendation (for example, proprietary data).

h4501SupplementaryService – This field carries a sequence of H4501SupplementaryService APDUs as defined in Table 3/H.450.1.

h245Tunneling – This element is set to TRUE if tunneling of H.245 messages is enabled.

h245Control – This field carries a sequence of tunneled H.245 PDUs.

nonStandardControl – This field contains a sequence of non-standard data elements that may be used in addition to or instead of the single **nonStandardData** field.

[End Correction]

6.2.12 Early Indication of the Refusal of Fast Connect

Description:	It has become apparent that there is a need for a called party to indicate to the calling party its acceptance or refusal of the Fast Connect procedures. A new field has been added to various H.225.0 messages to allow explicit indication that Fast Connect is refused. This will be incorporated into the next H.225.0 Recommendation.

For each message in the ASN.1 that contains the **fastConnectRefused**, the following definition shall apply.

[Begin Correction]

fastConnectRefused – A called endpoint should return this element in any message up to and including the Connect message when establishing a call to indicate that it refuses the Fast Connect procedure.

[End Correction]

6.2.13 Missing Release Complete Reasons in Table 5/H.225.0

Description:	New release complete reasons were added to H.225.0, but the cause IE mappings are not shown in Table 5/H.225.0. Below shows the additions made to table 5/H.225.0.

[Begin Correction]

Table 5/H.225.0 – Release Complete Reason to cause IE mapping

ReleaseCompleteReason code	Corresponding Q.931/Q.850 cause value
newConnectionNeeded	47 – Resource Unavailable
nonStandardReason	127 – Interworking, unspecified
replaceWithConferenceInvite	31 – Normal, unspecified

[End Correction]

6.2.14 Encoding the Extension Bit of Octet 3 of the Calling Party Number IE

Description:	Section 7.2.2.6 specifies that the encoding of the extension bit shall always be '1'. This is contradictory, since octet 3a of the calling party number may be present. This bit should be encoded following the rules of Table 4-9/Q.931. The text below shows the corrections to be applied, which will appear in the next published version of H.225.0.

[Begin Correction]

7.2.2.6 Calling party number

This information element is encoded following Figure 4-16/Q.931 and Table 4-11/Q.931.

~~*Octet No. 3 Extension (bit 8)*~~

~~– Set to '1'.~~

[End Correction]

6.2.15 Sending the h245Address field in Facility (extracted from Call Proceeding)

Description:	A called endpoint may return a Call Proceeding containing the H.245 Address of the endpoint. However, it is possible that a signaling entity in the middle, such as a routed Gatekeeper, has already generated a Call Proceeding message. If the Gatekeeper wants to use this opportunity to convey the H.245 address, it may use a Facility message. However, a distinction must be made between a Facility that simply contains an H.245 address, versus one where the other endpoint is explicitly requesting the initiation of H.245.

[Begin Correction]

7.4.1 Facility

...

reason – More information about the facility message. In case the message is sent by an intermediate signalling entity as a means of forwarding information from a Call Proceeding message, this field shall be set to **undefinedReason**.

...

h245Address – This is a specific transport address on which the endpoint or gatekeeper sending this facility would like the recipient to establish H.245 signalling. This field may be present when the **reason** is set to **undefinedReason** when an intermediate signalling entity is trying to convey the **h245Address** field from the Call Proceeding. The receiving entity is instructed to initiate H.245 only when the **reason** is **startH245**.

[End Correction]

6.2.16 Progress Message

Description:	An error was identified in the H.225.0 (1999) document, which stated that the Progress Indicator IE is optional. The Progress Indicator IE is, in fact, mandatory for the Progress message. Below shows the corrected text. This correction will be applied to H.225.0 (2000).

[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	~~O~~M	2-4
Notification indicator	O	2-*
Display	O	2-82
High layer compatibility	FFS	NA
User-to-User	M	2-131
NOTE – The Bearer capability information element is mandatory if the message is between a terminal and a gateway.		

[End Correction]

6.3 Technical and Editorial Corrections to ITU-T Recommendation H.245 (02/2000)

6.3.1 Enforcing Symmetric Codecs

Description:	Implementers have been confused over the meaning of "receive and transmit" capabilities and the reality in the market is that many DSPs require symmetric codec operation. For this reason, the following additions are added to H.245 (11/2000).

[Begin Correction]

The following behaviour is recommended to minimise the chance of endpoints attempting to open conflicting logical channels when the slave endpoint has symmetric capability limitations. When the master and the slave have indicated choices of receive capabilities for a particular media type, the slave should attempt to open a logical channel for the master's most preferred capability for which it has capability, as given by the order the master has expressed its capabilities; and the master should attempt to open a logical channel for its most preferred capability for which the slave has capability, as given by the order it has expressed its capabilities.

For example, if the master has declared capability for G.723.1, G.729, and G.711 and the slave has indicated capability for G.711 and G.729, with the most preferable being listed first in both cases, then both master and slave should attempt to open logical channels for G.729.

After the request to open a logical channel has been rejected by the master, the slave is responsible for opening a non-conflicting channel.

When the slaves detects a conflict and the master does not reject a conflicting open logical channel, the slave should close the conflicting channel. In the case of conflicting logical channels due to symmetric capability limitations, the slave should open an appropriate logical channel using the replacement for procedure, and in due course close the conflicting logical channel.

[End Correction]

6.3.2 Inconsistencies between the Text and Table G.4 in H.245v6

Description:	It was found that the current Annex G to H.245 have an inconsistency between the body text and table G.4. These errors will be corrected in H.245 (11/2000).

[Begin Correction]

Table G.1 below defines the capability identifier for ISO/IEC 14496-1 [49] capabilities. Tables G.2 to G.6 define the associated capability parameters for ISO/IEC 14496-1. These parameters shall only be included as **genericDataCapability** within the **DataCapability** structure and as **genericDataMode** within the **DataMode** structure. For capability exchange, ~~only~~ streamType and profileAndLevel ~~and objectType~~ shall be specified, and objectType may be specified. When opening a logical channel (forward or reverse) either ES_ID or objectDescriptor shall be specified.

Further information about the usage of the ISO/IEC 14496-1 Generic Capability is included in Annex F to H.324 version 1998.

[End Correction]

[Begin Correction]

TABLE G.4/H.245

Capability Parameter objectType

Parameter name:	objectType
Parameter description:	This is a nonCollapsing GenericParameter. objectType indicates the set of tools to be used by the decoder of the bitstream contained in one logical channel as given in • Table 8 of ISO/IEC 14496-1 ("objectTypeIndication Values ") for streamType = 0x04 or 0x05 • Table 7 of ISO/IEC 14496-1 ("graphicsProfileLevelIndication Values") for streamType = 0x03 For all other values of streamType, objectType is not

	defined and shall therefore not be used.
Parameter identifier value:	2
Parameter status	Optional. ~~Shall not be present for Capability Exchange. Shall be present for Logical Channel Signalling. May be present for Mode Request.~~ For streamType = 0x04 or 0x05, shall not be present for Capability Exchange, shall be present for Logical Channel Signaling. May be present for Mode Request. For streamType = 0x03, shall be present for Capability Exchange, shall be present for Logical Channel Signlaing. May be present for ModeRequest. For other streamType values, shall not be present.
Parameter type:	unsignedMax. Shall be in the range 0..255.
Supersedes:	-

[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 (1998)

6.5.1 Key Escrow Usage

Description:	A minor inconsistency has been discovered in the Recommendation H.235 Section 6.6.1. This change does not affect behavior or implementations in any way. This change will be applied to H.235v2 when published by the ITU.

[Begin Correction]

6.6.1 Key Escrow

Although not specifically required for operation, this recommendation contains provision for entities utilizing the H.235 protocol to support ~~key recovery~~the facility known as trusted third party (TTP) within the signalling elements.

[End Correction]

6.5.2 H.235 Control Channel References

Description:	A typographical error has been discovered in section 8 of the Recommendation H.235. This change does not affect behavior or implementations in any way. This change will be applied to H.235v2 when published by the ITU.

[Begin Correction]

8.2 Unsecured H.245 Channel Operation

Alternatively, the H.245 channel may operate in an unsecured manner and the two entities open a secure logical channel with which to perform authentication and/or shared-secret derivation. For example TLS or IPSEC may be utilized by opening a logical channel with the datatype containing a value for ~~**encryptionData**~~**h235Control**. This channel could then be used to derive a shared secret which protects any media session keys or to transport the **EncryptionSync**.

[End Correction]

6.5.3 Multipoint Procedure Section Reference

Description:	A minor error in a section reference has been discovered in Recommendation H.235 section 9.

[Begin Correction]

9.1 Authentication

Authentication shall occur between an endpoint and the MC(U) in the same manner that it would in a point-to-point conference. The MC(U) shall set the policy concerning level and stringency of authentication. As stated in section~~0~~ 6.6, the MC(U) is trusted; existing endpoints in a conference may be limited by the authentication level employed by the MC(U). New **ConferenceRequest** / **ConferenceResponse** commands allow endpoints to obtain the certificates of other participants in the conference from the MC(U). As outlined in H.245 procedures, endpoints in a multipoint conference may request other endpoint certificates via the MC, but may not be able to perform direct cryptographic authentication within the H.245 channel.

[End Correction]

6.5.4 Introduction to Authentication

Description:	The introductory text (paragraph 1) to Section 10 of Recommendation H.235 in unclear and potentially misleading. The corrected text is shown below.

[Begin Correction]

10.1 Introduction

Authentication is in general based either on using a shared secret (you are authenticated properly if you know the secret) or on public key based methods with certifications (you prove your identity by possessing the correct private key). A shared secret and the subsequent use of symmetric cryptography requires a prior contact between the communicating entities. A prior face-to-face or secure contact can be replaced by generating or exchanging the shared secret key with methods based on public key cryptography, e.g. by Diffie-Hellman key exchange. The communication parties in the key generation and exchange have to be authenticated for example by using digitally signed messages; otherwise the communication parties cannot be sure with whom they share the secret.

This Recommendation presents authentication methods based on subscription, i.e. there must be a prior contact for sharing a secret, and authentication methods where public key cryptography is directly used in authentication or it is used for generating the shared secret.

~~There are two types of authentication that may be utilized. The first type is symmetric encryption-based that requires no prior contact between the communicating entities. The second type is based on the ability to have some prior shared secret (further referenced as "subscription" based). Two forms of subscription-based authentication are provided: password and certificate.~~

[End Correction]

6.5.5 Diffie-Hellman Exchange with Optional Authentication

Description:	Two errors have been discovered in the labelling of parameters of arguments in the Diffie-Hellman exchange described in Recommendation H.235 section 10.2. Additionally, the note concerning authentication needs to be clarified. Phase 1: As this correction affects implementations, which utilize this mechanism to provide authentication during the Diffe-Hellman exchange. Note that if these optional parameters are not utilized (denoted by italics below and in the original recommendation) no implementation changes are needed.

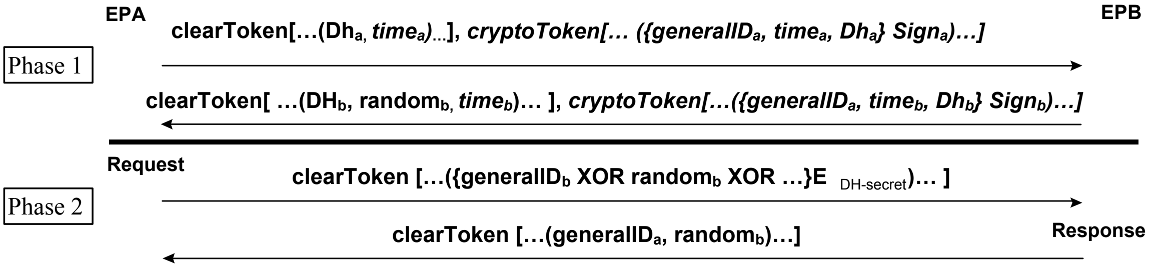
	Phase 2: The identifier (**generalID**) passed from in the second exchange (e.g. Response) should be that of the recipient of the Response message (e.g. EPA).

[Begin Correction]

10.2 Diffie-Hellman with optional Authentication

...

Note - If the messages are exchanged over an insecure channel, then digital signatures (or other message origin authentication method) must be used in order to authenticate the parties between whom the secret will be shared. An optional signature element may also be provided these are illustrated in italics below.


EPA
EPB
clearToken[…(Dh$_a$, *time$_a$*)…], *cryptoToken[… ({generalID$_a$, time$_a$, Dh$_a$} Sign$_a$)…]*
Phase 1
clearToken[…(DH$_b$, random$_b$, *time$_b$*)…], *cryptoToken[…({generalID$_a$, time$_b$, Dh$_b$} Sign$_b$)…]*
Request
clearToken […({generalID$_b$ XOR random$_b$ XOR …}E $_{\text{DH-secret}}$)…]
Phase 2
clearToken […(generalID$_a$, random$_b$)…]
Response

...

[End Correction]

6.5.6 Introduction to Subscription Based Authentication

Description:	The introductory text (paragraph 1) to Section 10.3.1 of Recommendation H.235 in unclear and potentially misleading. The text shown below shall be added as the new final paragraph of that section.

[Begin Correction]

10.3.1 Introduction

...

Note - In all cases where timestamps are generated and passed as part of a security exchange, implementers should take the following precautions. The time stamp granularity should be fine enough that it is guaranteed to increment with each message. If this is not guaranteed, replay attacks are possible. (e.g. if the timestamp only increments by the minute, then an endpoint 'C' can spoof endpoint 'A' within duration of one minute after endpoint 'A' has sent a message to endpoint 'B').

[End Correction]

6.5.7 Password with Hashing

Description:	The text to Section 10.3.3 of revision 1 of H.235 Recommendation has been determined to be unclear with respect to parameters that are passed in the

	exchange of messages. The included text should be added as a new, final paragraph.

[Begin Correction]

10.3.3 Password with Hashing

...

Note 3: The cryptoHashedToken structure is used to pass the parameters used in this exchange. Included in this structure are the 'clear' versions of parameters needed to compute the hashed value. Implementers should include the timestamp in the hashedVals and should not include the password. (E.g. both the password and the 'generalID' should be known a priori by the recipient).

Note 4: The hashing function shall be applied to the EncodedGeneralToken structure that includes at least the ID, timestamp and password fields. The password value should NOT be passed in the ClearToken.

[End Correction]

6.5.8 Corrections to Annex A

Description:	An omission in the ASN.1 syntax for H.235 has been discovered. Specifically, an identifier is missing from the **ClearToken** structure in the case where the **ClearToken** structure is placed directly into the message. The absence of this identifier will not allow multiple **ClearTokens** included in a single RAS message to be associated with individual uses. Additionally, **ClearTokens** may be defined for different uses that have the same format and these need to be differentiated by the **tokenOID**.

[Begin Correction]

```
ClearToken          ::= SEQUENCE  -- a `token' may contain multiple value types.
{
    tokenOID            OBJECT IDENTIFIER,
    timeStamp           TimeStamp OPTIONAL,
    password            Password OPTIONAL,
    dhkey               DHset OPTIONAL,
    challenge           ChallengeString OPTIONAL,
    random              RandomVal OPTIONAL,
    certificate         TypedCertificate OPTIONAL,
    generalID           Identifier OPTIONAL,
    nonStandard         NonStandardParameter OPTIONAL,
    . . .
}

--  An object identifier should be placed in the tokenOID field when a
--  ClearToken is included directly in a message (as opposed to being
--  encrypted).  In all other cases, an application should use the
--  object identifier { 0 0 } to indicate that the tokenOID value is not present.
```

[End Correction]

6.5.9 Corrections to Annex B

Description:	A number of typographical errors have been discovered in Annex B. The corrected text is shown below.

[Begin Correction]

2. Signalling and Procedures

...

One purpose of H.225.0 exchanges as they relate to H.323 security, is to provide a mechanism to set up the secure H.245 channel. Optionally, authentication may occur during the exchange of H.225.0 messages. This authentication may be certificate or password based, utilizing encryption and/or hashing (i.e. signing). The specifics of these modes of operation are described in sections (~~0-0~~4.2-4.3)

...

[End Correction]

[Begin Correction]

4.1 Introduction

This annex will not explicitly provide any form of message privacy between gatekeepers and endpoints. There are two types of authentication that may be utilized. The first type is symmetric encryption based that requires no prior contact between the endpoint and Gatekeeper. The second type is subscription based and will have two forms, password or certificate. All of these forms are derived from the procedures shown in sections ***[change these to document cross-references]*** 10.2, 10.3.2, 10.3.3 and 10.3.4. In this annex, the generic labels (EPA and EPB) showed in the aforementioned sections will represent the Endpoint and Gatekeeper respectively.

...

[End Correction]

[Begin Correction]

4.2 Endpoint-Gatekeeper Authentication (Non-Subscription Based)

This mechanism may provide the Gatekeeper with a cryptographic link that a particular endpoint, which previously registered, is the same one that issues subsequent RAS messages. It should be noted that this might not provide any authentication of the Gatekeeper to the endpoint, unless the optional signature element is included. The establishment of the identity relationship occurs when the terminal issues the GRQ as outlined in H.323 section ***[change to cross-reference]*** 7.2.1. The Diffie-Hellman exchange shall occur in conjunction with the GRQ and GCF messages as shown in the first phase of section 0. This shared secret key shall now be used on any subsequent RRQ/URQ from the terminal to the gatekeeper. If a Gatekeeper operates in this mode and receives a GRQ without a token containing the DHset or an acceptable algorithm value, it shall return a securityDenial reason code in the DRJ.

Terminal (**xRQ**):

1) The terminal shall provide all of the information in the message as described in the appropriate H.225.0 sections.

2) The terminal shall encrypt the GatekeeperIdentifier (as returned in the **GCF**) using the shared secret key that was negotiated. This shall be passed in the ~~**cryptoToken**~~ **clearToken** (see section 10.2) as the **generalID**.

The 16 bits of the random and then the requestSeqNum shall be XOR'd with each 16 bits of the GatekeeperIdentifier. If the GatekeeperIdentifier does not end on an even 16 boundary, the last 8 bits of the GatekeeperIdentifier shall be XOR'd with the least significant octet of the random value and then requestSeqNum. The GatekeeperIdentifier shall be encrypted using the selected algorithm in the GCF (~~integrity~~**algorithmOID**) and utilizing the entire shared secret.

The following example illustrates this procedure:

RND16: 16 bit value of the Random Value

SQN16: 16 bit value of requestSeqNum

BMPX: the Xth BMP character of GatekeeperIdentifier

BMP1' = (BMP1) XOR (RND16) XOR (SQN16)

BMP2' = (BMP2) XOR (RND16) XOR (SQN16)

BMP3' = (BMP3) XOR (RND16) XOR (SQN16)

BMP4' = (BMP4) XOR (RND16) XOR (SQN16)

BMP5' = (BMP5) XOR (RND16) XOR (SQN16)

:

:

BMPn' = (BMPn) XOR (RND16) XOR (SQN16)

...

[End Correction]

[Begin Correction]

5.1 Gateway

As stated in section ***[change to cross reference]*** 6.6, an H.323 Gateway should be considered a trusted element. This includes protocol gateways (H.323-H.320 etc.…) and security gateways (proxy/firewalls). The media privacy can be assured between the communicating endpoint and the gateway device; but what occurs on the far side of the gateway should be considered insecure by default.

[End Correction]

6.5.10 Corrections to Appendix I

Description:	A typographical error has been discovered with respect to a section reference. The corrected text is shown below.

[Begin Correction]

4.2 Password

...

The encryption key is constructed from the user's password using the procedure described in section 1~~3.3.3.341~~0.3.2 of H.235. The resulting octet "string" is then used as the DES key to encrypt the **challenge**.

...

[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]

6.6.2 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)

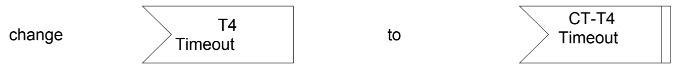
Change:

"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.

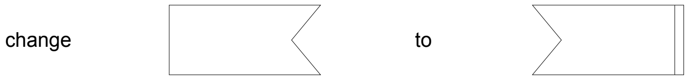
[Begin Correction]

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.



[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)

There are no corrections for H.450.8.

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 {
            callSignalStatsCallConnections~~In~~
               Counter32,
~~            callSignalStatsCallConnectionsOut~~
~~               Counter32,~~
            callSignalStatsAlertingMsgs~~In~~
               Counter32,
~~            callSignalStatsAlertingMsgsOut~~
~~               Counter32,~~
            callSignalStatsCallProceedings~~In~~
               Counter32,
~~            callSignalStatsCallProceedingsOut~~
~~               Counter32,~~
            callSignalStatsSetupMsgs~~In~~
               Counter32,
~~            callSignalStatsSetupMsgsOut~~
~~               Counter32,~~
            callSignalStatsSetupAckMsgs~~In~~
               Counter32,
~~            callSignalStatsSetupAckMsgsOut~~
~~               Counter32,~~
            callSignalStatsProgressMsgs~~In~~
               Counter32,
```

~~callSignalStatsProgressMsgsOut~~
~~Counter32,~~
callSignalStatsReleaseCompleteMsgs~~In~~
Counter32,
~~callSignalStatsReleaseCompleteMsgsOut~~
~~Counter32,~~
callSignalStatsStatusMsgs~~In~~
Counter32,
~~callSignalStatsStatusMsgsOut~~
~~Counter32,~~
callSignalStatsStatusInquiryMsgs~~In~~
Counter32,
~~callSignalStatsStatusInquiryMsgsOut~~
~~Counter32,~~
callSignalStatsFacilityMsgs~~In~~
Counter32,
~~callSignalStatsFacilityMsgsOut~~
~~Counter32,~~
callSignalStatsInfoMsgs~~In~~
Counter32,
~~callSignalStatsInfoMsgsOut~~
~~Counter32,~~
callSignalStatsNotifyMsgs~~In~~
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 (1999)

6.8.1 Multiple Usage Indications for the Same Call

Description:	H.225 Annex G does not fully define the behavior when more than one **UsageIndication** message is received for the same **callIdentifier** and **senderRole**, although **usageCallStatus** of **callInProgress** implies that there will be another later **UsageIndication**. This text clarifies the text in Annex G/H.225.0 and will be inserted into the next version of Annex G published by the ITU.

[Begin Correction]

G.7.4.1 Multiple Usage Indications for the Same Call

Multiple Usage Indications for the same call provide increasingly more up to date information on the same media types, or usage information about new media types created in the same call. Also, since border elements may take over calls while being in progress, not all the Usage Indications necessarily originate from the same border element. The following rules define the semantics:

1. UsageIndication received with a usageCallStatus of callInProgress implies a subsequent UsageIndication with the same callIdentifier and senderRole should be received. If the recipient is configured for fault recovery it may choose to conclude after a configured time interval with no further UsageIndication messages, that a fault has occurred and recover whatever data it can from the received UsageIndication messages.
2. Subsequent UsageIndication messages with the same usageField ids should report a startTime matching the endTime of the previous message (although this may be impossible for an alternate border element). Recipients shall assume each report is for a distinct period. Other information in the usageField overrides the information received in previous messages with the same usageField id.
3. A border element should send a new Usage Indication for each change in the media type during the call, e.g., audio stopped and fax started, or a codec has changed. If multiple media types are engaged at the same time (e.g. audio & video) they should be reported in the same UsageIndication message.

[End Correction]

[Begin Correction]

G.7.4 Usage Information Exchange

Administrative domains may request other domains to provide them information about the usage of resources in specific calls. UsageIndication messages may be provided at any stage of the call. Also, multiple usage indications may be sent for the same call, each one with possibly more up to date information, or reporting on consecutive call segments or different media type usage. See section 1.7.4.1 for detail.

...

[End Correction]

[Begin Correction]

G.8.2.28 Usage Indication

Report call details and usage information. This message is sent with respect to the last UsageSpecification element received by the BE concerning the call.

Field	Description
CallInfo	The call for which the indication applies.
AccessTokens	The access tokens for the call. These are the tokens that were received in the address template used for the call, and propagated in the AccessRequest / Setup message for the same call.
SenderRole	The role of the sender of the indication: • Originator – originating party. • Destination – terminating party. • NonStandard – other.
UsageCallStatus	The current status of the call: • preConnect • callInProgress • callEnded • RegistrationLost
SourceAddress	E.164 or e-mail address of the caller party. In case of E.164 this designates the ANI/CLI.
DestAddress	E.164 or e-mail address for the called party,
StartTime	The time the call started in UTC format. Relevant only for calls that passed the setup stage. For multiple media types used in the call, each media type should report a different StartTime, corresponding to the time at which that media stream started.

For periodic messages StartTime should correspond with the EndTime of the previous message.

EndTime — The time the call ended in UTC format. Relevant only for ended calls. For multiple media types used in the call, each media type shall report a different EndTime corresponding to the time at which that media stream ended. For periodic messages, EndTime is the time which ends a reporting period.

TerminationCause — The reason for the end of the call. Relevant only for ended calls.

usageInformation — Set of fields of information. Each field is represented by a UsageField which can be a standard or non-standard. Standard UsageFields are for future study.

...

[End Correction]

6.8.2 Identifying the Terminated Service Relationship

Description:	In the **ServiceRelease** message, there is no information to identify the service relationship that is being terminated.

[Begin Correction]

G.8.2 Message Definitions

...

ServiceID — This identifer identifies a particular service relationship session between two border elements. Whenever a border element receives a ServiceRequest message requesting the establishment of a new service relationship (which is indicated by the absence of the service ID field in the ServiceRequest message), it allocates a **globally** unique serviceID and returns it to the sender of the ServiceReque st message in the ServiceConfirm message.

Once a service relationship has been established, the service ID is included in all subsequent messages with the border element (e.g. usage indication, descriptorID request, descriptor request, access request). This is used by the recipient border element to check if it has a service relationship with the sender of the message.

...

[End Correction]

[Begin Correction]

G.8.2.6 Service Confirmation

A border element in receipt of a ServiceRequest message responds with a ServiceConfirmation message to indicate that it agrees to establish a service relationship. Every new service relationship is identified by a service identifier. Whenever a border element receives a ServiceRequest message without a service ID, it allocates a unique service ID and returns it to the sender of the service request message in the "service confirm" message. If the border element already has a service relationship with the border element that sent the ServiceRequest message, sending ServiceConfirmation indicates that the terms of the original relationship are terminated and replaced with the new terms. The ServiceConfirmation message shall contain the same service ID that was sent in the ServiceRequest message. A border element that receives a ServiceRequest message containing a service ID that it does not recognize shall respond with a ServiceRejection message.

...

[End Correction]

6.8.3 Need to Provide a replyAddress when using Bi-directional Connections

Description:	Currently a request message sent over bi-directional connection oriented transport like TCP is not expected to have the **replyAddress** element in the **AnnexGCommonInfo**.**replyAddress**. This implies that a receiver can send data to the sender only as long as the TCP connection is up. This results in a problem if a "response" needs to be sent to the sender after the original TCP connection has been released, because the receiver does not have the transport address of the sender. E.g.: this could happen when a **ServiceRelease** needs to be generated long after the establishment of a service relationship. The following corrections shall be applied to Annex G/H.225.0.

[Begin Correction]

G.8.2 Message Definitions

...

ReplyAddress This is the address to which to send the reply to a request message. All request messages shall include a **replyAddress** except for cases where the address can be derived from the transport layer. On IP networks, if the sender of the request message is listening on the default port (2099), then the reply address need not be included. In such a case, the receiver obtains the transport address of the sender by appending default port (2099) to the IP address of the sender as received in the IP header of the request packet.

[End Correction]

A footnote shall also be added to the "ReplyAddress" definition that reads:

> BEs are assumed not to be hidden behind network address translation (NAT) devices, thus it is not required to prefer the transport address over the **replyAddress**, as is the case for RAS messages.

6.8.4 Sending UsageIndications without a Service Relationship

Description:	Currently Annex G specifications mandate that usage Indication message cannot be sent out unless there is a service relationship between two border elements. Since a border element is not mandated to have a service relationship in a secured environment (or in an environment where security issues are handled by non-Annex G procedures), it is limiting that such border elements cannot exchange usage indication messages. The following corrections shall be applied to Annex G/H.225.0.

[Begin Correction]

G.7.4 Usage Information Exchange

...

Usage Indications may be exchanged irrespective of whether the two border elements have a service relationship between them. However the policy of a border element may not allow such exchanges without a service relation. In such a case, the border element may reject the usage indication message, with an error code **noServiceRelationship**.~~Usage Indications may be exchanged only if the two border elements have service relationship between them.~~

...

[End Correction]

[Begin Correction]

G.8.2.5 Service Request

...

The recipient of the ServiceRequest may indicate alternate border elements that the sender of ServiceRequest may try for backup service. Establishing a service relationship~~Establishment of a service relationship is mandatory for Usage Indication message exchanges. Otherwise, it~~ is an optional procedure, although a border element's policy may require such a relationship.

...

[End Correction]

6.8.5 Changes to the ASN.1 in Annex G/H.225.0

Description:	This section shows the changes to the ASN.1 required to support the changes and corrections to Annex G/H.225.0.

[Begin Correction]

Message Syntax

...

```
AnnexGCommonInfo    ::= SEQUENCE
{
    sequenceNumber          INTEGER(0...65535),
    version                 AnnexGVersion,
    hopCount                INTEGER (1...255),
    replyAddress            SEQUENCE OF TransportAddress OPTIONAL,
    integrityCheckValue     ICV OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL
    nonStandard             SEQUENCE OF NonStandardParameter OPTIONAL,
    ...,
    serviceID               ServiceID   OPTIONAL
}

ServiceID                   ::= GloballyUniqueID

UsageCallStatus ::= CHOICE
{
    preConnect              NULL,       -- Call has not started
    callInProgress          NULL,       -- Call is in progress
    callEnded               NULL,       -- Call ended
    ...,
    registrationLost        NULL   -- Uncertain if call ended or not
}

UsageSpecification ::= SEQUENCE
{
    sendTo                  ElementIdentifier,
    when SEQUENCE
    {
        never       NULL OPTIONAL,
        start       NULL OPTIONAL,
        end         NULL OPTIONAL,
        period      INTEGER(1..65535) OPTIONAL,   -- in seconds
        failures    NULL OPTIONAL,
        ...
    },
    required            SEQUENCE OF OBJECT IDENTIFIER,
    preferred           SEQUENCE OF OBJECT IDENTIFIER,
    ...,
    sendToBEAddress  AliasAddress OPTIONAL
}

GlobalTimeStamp         ::=     IA5String (SIZE(14))
                                    -- UTC in the form YYYYMMDDHHmmSS
                                    -- where YYYY = year, MM = month, DD = day,
                                    -- HH = hour, mm = minute, SS = second
                                    -- (for example, 19981219120000 for noon
                                    -- 19 December 1998)

ServiceRejectionReason ::= CHOICE
{
    serviceUnavailable  NULL,
    serviceRedirected       NULL,
    security                NULL,
    continue                NULL,
    undefined               NULL,
    ...,
    unknownServiceID        NULL
```

}

...

[End Correction]

6.8.6 Clarification Relating to Service Relationships

Description:	The text in the section describing the fields for the Usage Specification suggests that an endpoint should have a service relationship with a border element, but this is entirely optional. The text altered to clarify the fact that this is, indeed, optional.

[Begin Correction]

G.8.2.4.5 Usage Specification

SendTo Border element to send the UsageIndication messages to. ~~Since~~ If the sender ~~should have~~has a service relationship with that border element, this is the element identifier returned in the ServiceConfirmation message.

[End Correction]

6.8.7 Corrections for the Usage Indication Rejection

Description:	The reasons for a Usage Indication Rejection in the field descriptions do not align with the ASN.1 and are also not fully defined. The corrected text is shown below.

[Begin Correction]

G.8.2.30 Usage Indication Rejection

Reason This is the reason the border element rejected the UsageIndication message. Choices are:

- ~~InvalidCall~~ UnknownCall - The call specified in the UsageIndication is not a recognized call.

- Incomplete - The UsageIndication did not contain all the information required by the UsageSpecification that applies to this UsageIndication.

- Security – The UsageIndication did not meet the recipient's security requirements.

- NoServiceRelationship- The recipient will exchange this information only after establishment of a service relationship.

- Undefined – The reason for rejecting the UsageIndication does not match any of the other choices.

[End Correction]

6.8.8 Corrections to tables and Diagrams

Description:	It was pointed out that there are unintended ambiguous identifiers assigned as zone descriptor values in the tables and figures in sections 1.9.1, 1.9.1.1, 1.9.2, and 1.9.2.1. The diagrams below replace the coresponding tables/figures those sections.

The table in 1.9.1 should be replaced with the below table.

[Begin Correction]

Administrative Domain	Template definition	Comment
A	Descriptor “d1”: Pattern = 1732* Transport address = BE$_A$ call signal address Message type = sendSetup	Signaling for any call into AD A will be through AD A’s border element.
B	Descriptor “~~d1~~d2”: Pattern = 1908* Transport address = BE$_B$ annex g address Message type = sendAccessRequest Descriptor “~~d2~~d3”: Pattern = 1908953* Transport address = GW$_{B1}$ CALL SIGNALLING address Message type = sendSetup	For calls to 1908*, an AccessRequest message is needed to get the destination’s (i.e., a gateway) call signaling address. For calls to 1908953*, the Setup can be sent directly to this particular gateway.
C	Descriptor “~~d1~~d4”: Pattern = 1303538* Transport address = GK$_{C1}$ call signal address Message type = sendSetup Descriptor “~~d2~~d5”: Pattern = 1303* Transport address = BE$_C$ annex g address	Calls to 1303538* will be routed through this particular gatekeeper. Calls to 1303* can be signalled directly to the destination gateway, but an AccessRequest must be sent to obtain the gateway’s call signaling address.

	Message type = sendAccessRequest	

[End Correction]

The figure in section 1.9.1.1 shall be replaced with the table below.

[Begin Correction]

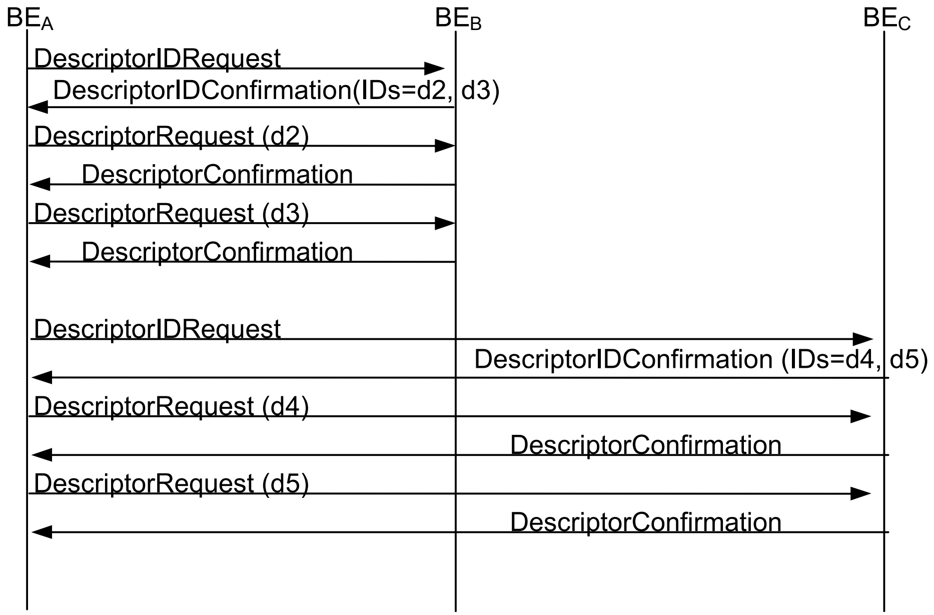


Figure G.8/H.225.0 - Example of Descriptor Exchange

[End Correction]

The table in 1.9.2 should be replaced with the below table.

[Begin Correction]

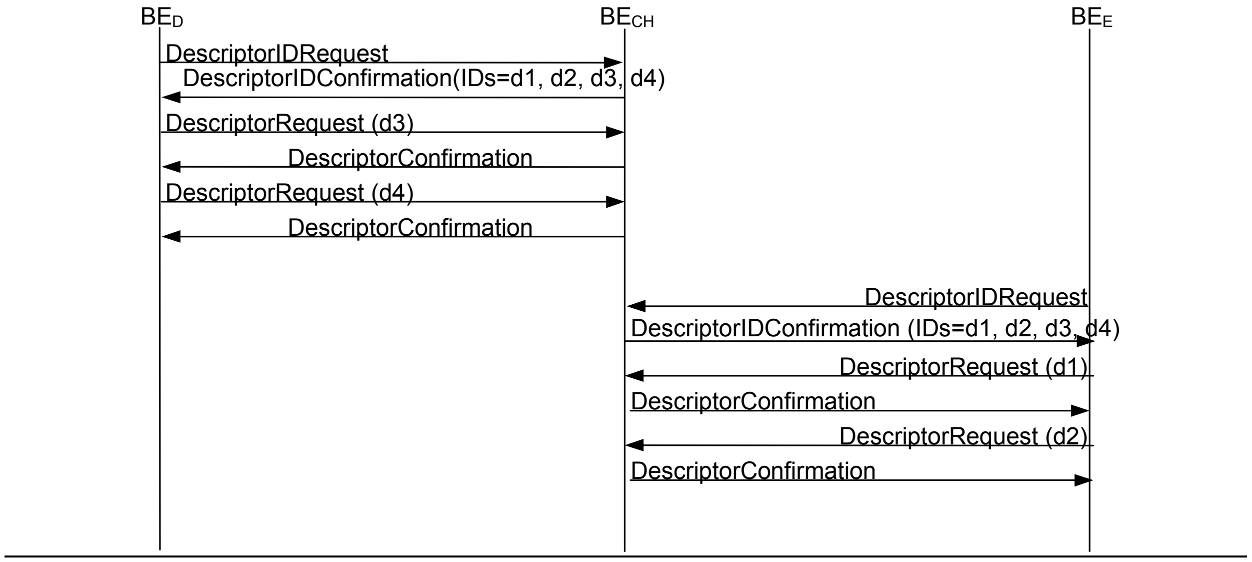
Administrative Domain	Template definition	Comment
D	Descriptor "d1": Pattern = 1908* Transport address = BE$_D$ annex g address Message type = sendAccess Request Descriptor "d2": Pattern = 1908953* Transport address = GW$_{D1}$ Call Signalling address Message type = sendSetup	For calls to 1908*, an Access Request message is needed to get the destination's (i.e., a gateway) call signaling address. For calls to 1908953*, the Setup can be sent directly to this particular gateway.
E	Descriptor "~~d1~~d3": Pattern = 1303538*	Calls to 1303538* will be routed through this particular gatekeeper.

	Transport address = GK_{E1} call signal address Message type = sendSetup Descriptor "~~d2~~d4": Pattern = 1303* Transport address = BE_E annex g address Message type = sendAccess Request	Calls to 1303* can be signalled directly to the destination gateway, but an AccessRequest must be sent to obtain the gateway's call signaling address.
CH	Descriptor "d1": Pattern = 1908* Transport address = BE_D annex g address Message type = sendAccess Request Descriptor "d2": Pattern = 1908953* Transport address = GW_{D1} call signalling address Message type = sendSetup Descriptor "d3": Pattern = 1303538* Transport address = GK_{E1} call signal address Message type = sendSetup Descriptor "d4": Pattern = 1303* Transport address = BE_E annex g address Message type = sendAccess Request	The clearing house obtains descriptors from other ADs and holds this information for distribution during descriptor exchange.

[End Correction]

The figure in section 1.9.2.1 shall be replaced with the figure below.

[Begin Correction]



[End Correction]

6.8.9 Receiving Descriptors

Description:	The wording of section G.7.1.2 specifies that a border element can request only statically configured templates from a remote border element. This is not correct - any template can be requested.

[Begin Correction]

G.7.1.2 Receiving Descriptors

A border element may request ~~the statically configured~~ templates from another border element. The response to the request is decided by the border element from which the templates are being requested.

...

[End Correction]

6.8.10 Corrections Related to UTC

Description:	Various time-related fields should be specified as UTC.

[Begin Correction]

G.8.2.2 Descriptor Information

Descriptor information uniquely identifies the descriptor and indicates the last time the descriptor changed.

Field	Description
DescriptorID	This is a globally unique identifier used to identify this descriptor from among many possible descriptors.
LastChanged	This is the UTC date and time this descriptor was last changed.

[End Correction]

[Begin Correction]

G.8.2.3.2 Pricing Information

...

ValidFrom	This is the UTC date and time from which this information is valid.
ValidUntil	This is the UTC date and time at which this information expires.

[End Correction]

6.8.11 Editorial Corrections

Description:	Editorial Corrections G.8.2.3.3 - changed "describing" to "descending" G.8.2.19 - changed "CallInfoNeeded" to "needCallInformation" G.8.2.27 - add missing descriptions to reason codes G.8.2.28 - change "usageInformation" to "usageFields"

[Begin Correction]

G.8.2.3.3 Contact Information

...

Security	Security mechanism in ~~describing~~ descending order of preference to be used when communicating with contact.

[End Correction]

[Begin Correction]

G.8.2.19 Access Rejection

...

Reason	... • ~~CallInfoNeeded~~ needCallInformation – Specific call information was not present in the request.

[End Correction]

[Begin Correction]

G.8.2.27 Usage Rejection

The UsageRejection message is sent in response to a UsageRequest message to indicate that the recipient rejected the request and will not send the usage indications subsequently.

Field	**Description**
Reason	This is the reason the border element rejected the UsageRequest. Choices are: • InvalidCall - The call specified in the UsageRequest is not a recognized call. • Security - The UsageRequest did not meet the recipient's security requirements. • Unavailable - The recipient does not have usage information for the requested call. • noServiceRelationship - The recipient will exchange this information only after establishment of a service relationship. • Undefined - The reason for rejecting the UsageRequest does not match any of the other choices.

[End Correction]

[Begin Correction]

G.8.2.28 Usage Indication

...

~~usageInformation~~usageFields	Set of fields of information. Each field is represented by a *UsageField* which can be a standard or non-standard. Standard UsageFields are for future study.

[End Correction]

6.8.12 Directing UsageIndications to Specific Border Elements

Description:	The "sendTo" field in the UsageSpecification is an identifier, and the border element receiving this field might not, in all cases, know how to resolve this identifier to the address of a destination border element to which UsageIndication messages should be sent. An additional field of type

	AliasAddress was added to the UsageSpecification structure to allow a border element that receives a UsageSpecification to always be able to determine the address to where UsageIndication messages should be sent.

Refer to section 6.8.5 for ASN.1 additions.

[Begin Correction]

G.8.2.4.5 Usage Specification

...

sendToBEAddress This is a resolvable address that, when resolved, specifies the address of a border element to which UsageIndication messages shall be sent. If the resolution of this field results in more than one address (for example, in the case where a DNS query returns a list of addresses), the border element shall send the UsageIndication messages to only one border element from the list.

If the border element does not succeed in sending to one address, it may choose another address from the list and attempt to send the UsageIndication messages to the new address. The border element may continue attempting each additional address in the list until it either receives a UsageIndicationConfirmation, a UsageIndicationRejection, or until there are no further addresses to attempt.

Note that the "sendToBEAddress" field is different from the "sendTo" field in the UsageSpecification. The "sendTo" field is an identifier. It can be the identifier of a specific border element (e.g., "border_element1"), or it can be an identifier that logically represents a set of border elements (e.g., "border elements of my company").

The "sendToBEAddress" field resolves to one or more addresses.

[End Correction]

6.8.13 Rejecting Service Requests Due to Unknown ServiceID Value

Description:	A deficiency was noted in Annex G wherein it was not possible for a Border Element to inform another Border Element that the reason that a ServiceRequest is rejected is due to the fact that an unknown service ID is provided. This correction is shown here and will appear in the next version of Annex G/H.225.0.

Refer to section 6.8.5 for ASN.1 additions.

[Begin Correction]

reason This is the reason the border element rejected the ServiceRequest.

Choices are:

. . .

- unknownServiceID - the serviceID field contained in the ServiceRequest message is not recognized by the border element

[End Correction]

6.8.14 Corrections Relating to user-user Information Element

Description:	Clarify handling of user-user information element received in a SCN Q.931 message.

7.2.2.31 User-user

Encoded following Figure 4-36/Q.931 and Table 4-26/Q.931, as modified here.

The user-user information element shall be used by all H.323 entities to convey H.323-related information. Actual user-user information to be exchanged only between the involved terminals, that is, data received in the user-user information element of a received SCN Q.931 message, is nested in the H323-UserInformation PDU as protocol discriminator and user-information (to which no restrictions apply).

[End Correction]

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)	Redirecting number information element Type of number (6)

Address signal (3)	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 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	*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*
Or	*1536 kbit/s unrestricted*	*1536 kbit/s*
	1920 kbit/s unrestricted	*1920 kbit/s*
Unrestricted digital information with tones/announcements	*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 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 *terminating access non-ISDN*

[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]

[Begin Correction]

E.2.3.2 Well-known port

UDP port **2517** shall be used for the well-known port.~~ Entities may transmit from any random port~~ All messages pertaining to a single session shall be transmitted from the same IP address and port.

[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]

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)

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.

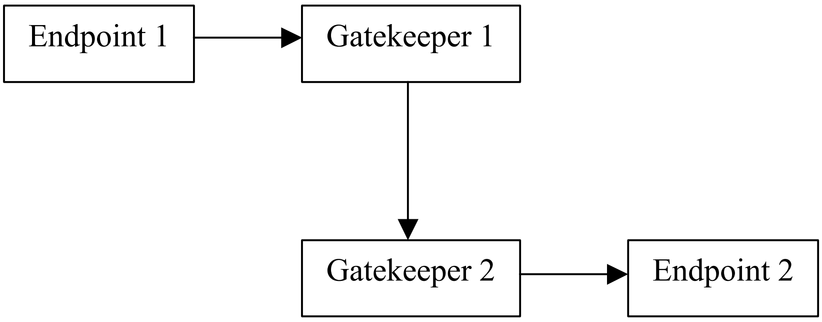


Figure 1 - Call placed through multiple gatekeepers

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.

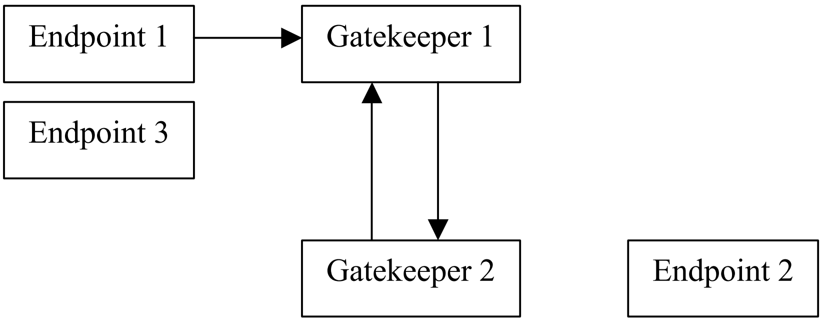


Figure 2 - Gatekeeper 2 re-routes call back to Gatekeeper 1

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*: 1-3	
{ 0 0 h(8) 2250 annex(1) g(7) version(0) *[v]* }	H225.0 Annex G version numbers
Assigned values of *v*: 1	
{ 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*: 1-6	
{ 0 0 h(8) 245 generic-capabilities(1) video(0) *[c]* }	Generic video capabilities
Assigned values of *c*: Is14496-2(0)	
{ 0 0 h(8) 245 generic-capabilities(1) audio(1) *[c]* }	Generic audio capabilities
Assigned values of *c*: none	
{ 0 0 h(8) 245 generic-capabilities(1) data(2) *[c]* }	Generic data capabilities
Assigned values of *c*: none	
{ 0 0 h(8) 245 generic-capabilities(1) control(3) *[c]* }	Generic control capabilities
Assigned values of *c*: Logical-channel-bit-rate-management(0)	
{ 0 0 h(8) 245 generic-capabilities(1) multiplex(4) *[c]* }	Generic multiplex capabilities
Assigned values of *c*: none	
{ 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:

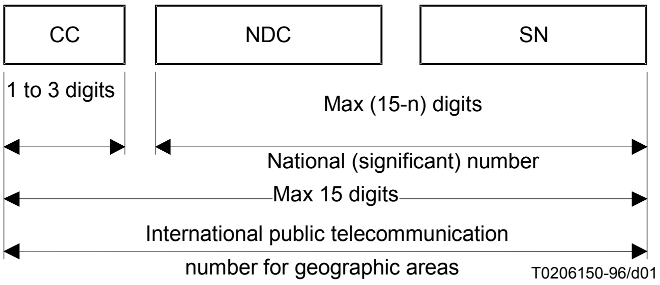
1300	TLS secured call signalling
1718	Multicast RAS Signalling
1719	Unicast RAS Signalling
1720	TCP call signalling
2099	Annex G/H.225.0 Signalling

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
NDC
SN
1 to 3 digits
Max (15-n) digits
National (significant) number
Max 15 digits
International public telecommunication
number for geographic areas
T0206150-96/d01

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.

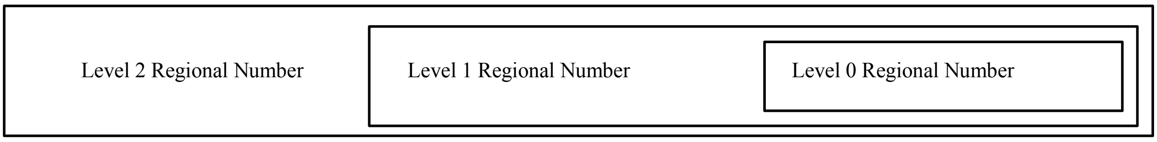


Figure – H.323 - Structure of a PNP Number with three levels of regions

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	calledName	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