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

Contact Information

ITU-T Study Group 16 / Question 2 Rapporteur	Paul E. Jones Cisco Systems, Inc. 7025 Kit Creek Road Research Triangle Park, NC 27709 USA	Tel: +1 919 392 6948 Fax: +1 919 392 7065 E-mail: paulej@packetizer.com
ITU-T Study Group 16 / Question 3 Rapporteur ITU-T Recommendation H.450.8 and H.341Editor	Glen Freundlich Avaya Communication 1300 W. 120th Avenue Westminster, CO 80234 USA	Tel: +1 303 538 2899 Fax: +1 303 538 3007 E-mail: ggf@avaya.com
ITU-T Recommendation H.225.0 Editor ITU-T Recommendation H.323 and Implementer's Guide Editor	Vivek Bhargava Cisco Systems, Inc. 7025 Kit Creek Road Research Triangle Park, NC 27709 USA	Tel: +1 919 392 6823 Fax: +1 919 392 7065 E-mail: vbhargava@packetizer.com
ITU-T Recommendation H.225.0 Annex G Editor	Michael Fortinsky VocalTec Communications, Ltd. 2 Maskit St. Herzeliya 46733 Israel	Tel: +972 9 970 7768 Fax: +972 9 956 1867 E-mail: mike@vocaltec.com
ITU-T Recommendation H.235 Editor	Martin Euchner Siemens AG ICN M NT 18 Hofmannstr. 51 D-81359 Muenchen Germany	Tel: +49 89 722 5 57 90 Fax: +49 89 722 4 68 41 E-mail: martin.euchner@icn.siemens.de
ITU-T Recommendation H.245 Editor	Mike Nilsson BT Labs Ipswitch United Kingdom	Tel: +44 1 473 645413 Fax: +44 1 473 643791 E-mail: 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: +49 89 722 34570 Fax: +49 89 722 23977 E-mail: 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: +1 613 592 8450 Fax: +1 613 592 9634 E-mail: 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 (2000), *Packet-Based multimedia communications systems*
- ITU-T Recommendation H.225.0 (2000), *Call signaling protocols and media stream packetization for packet based multimedia communications Systems*
- ITU-T Recommendation H.225.0 – Annex G (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 (2000), Security and encryption for H Series (H.323 and other H.245 based) multimedia terminals
- ITU-T Recommendation H.450.1 (1998), *Generic functional protocol for the support of supplementary services in H.323*
- ITU-T Recommendation H.450.2 (1998), *Call transfer supplementary service for H.323*
- ITU-T Recommendation H.450.3 (1998), *Call diversion supplementary service for H.323*
- ITU-T Recommendation H.450.4 (1999), *Call Hold Supplementary Service for H.323*
- ITU-T Recommendation H.450.5 (1999), *Call Park and Call Pickup Supplementary Services for H.323*
- ITU-T Recommendation H.450.6 (1999), *Call Waiting Supplementary Service for H.323*
- ITU-T Recommendation H.450.7 (1999), *Message Waiting Indication Supplementary Service for H.323*
- ITU-T Recommendation H.450.8 (2000), *Name Identification Supplementary Service For H.323*
- 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 (2000)

6.1.1 H.323 Annex L Section 3.4

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

[Begin Correction]

3.4 Encoding

...

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

...

[End Correction]

6.1.2 Calling party address information Correction

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

[Begin Correction]

7.8.2.1 Calling party address information

Calling party address information appears in the Setup message.

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

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

[End Correction]

[Begin Correction]

7.8.3.1 Gateway as originating endpoint

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

[End Correction]

[Begin Correction]

7.8.4.1 Gateway as terminating endpoint

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

[End Correction]

6.1.3 Status/Status Inquiry messages without explicit Call Identifiers

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

[Begin Correction]

7.3 Call signalling channel

...

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

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

[End Correction]

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

6.2.1 Registration Request (RRQ) Corrections

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

[Begin Correction]

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

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

[End Correction]

6.2.2 Section 7.6 H.225.0 Common Message Elements Correction

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

[Begin Correction]

...

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

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

…

[End Correction]

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

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

[Begin Correction]

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

~~H248PackagesDescriptor ::= PackagesDescriptor~~

~~H248SignalsDescriptor ::= SignalsDescriptor~~

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

…

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

```
                                                        -- invitation to a conference
                                                        -- (see H.323 8.4.3.8)
        genericDataReason               NULL,
        neededFeatureNotSupported       NULL,
        tunnelledSignallingRejected     NULL,
        invalidCID                      NULL
}

                                        ...

CircuitIdentifier ::= CHOICESEQUENCE
{
        cic             CicInfo OPTIONAL,
        group           GroupID OPTIONAL,
        ...
}

                                        ...

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

                                        ...
```

[End Correction]

6.2.4 Clarification for the usage of rasAddress

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

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

[Begin Correction]

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

[End Correction]

6.2.5 ReleaseCompleteReason to Cause IE mapping

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

[Begin Correction]

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

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

[End Correction]

6.2.6 Clarification for sending PNP numbers in Information messages

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

[Begin Correction]

Table 9/H.225.0 – Information Message Content

Information element	H.225.0 status (M/F/O)	Length in H.225.0
Protocol discriminator	M	1
Call reference	M	3
Message type	M	1
Sending complete	O	1
Display	O	2-82
Keypad facility	O	2-34
Signal	O	2-3
Called party number	O (Note)	2-35
User-user	M	2-131
Note: The Called Party Number IE will be used to carry numbers from a Private Numbering Plan when performing overlapped sending according to 8.1.12/H.323.		

[End Correction]

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

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

[Begin Correction]

Table 8/H.225.0 – Connect

Information element	H.225.0 status (M/F/O)	Length in H.225.0
Protocol discriminator	M	1
Call reference	M	3
Message type	M	1
Bearer capability	O ~~(Note)~~	5-6
Extended facility	O	8-*
Channel identification	FFS	NA
Facility	O	8-*
Progress indicator	O	2-4
Notification indicator	O	2-*

Display	O	2-82
Date/Time	O	8
Connected Number	O	2-*
Connected Sub-Address	O	2-23
Low layer compatibility	FFS	NA
High layer compatibility	FFS	NA
User-user	M	2-131
~~NOTE – Bearer capability is mandatory if the message is between a terminal and a gateway.~~		

[End Correction]

[Begin Correction]

Table 10/H.225.0 – Progress

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

[End Correction]

6.3 Technical and Editorial Corrections to ITU-T Recommendation H.245 (11/2000)

6.3.1 Syntax Errors in H.245v7

Description:	The following errors were discovered in the published H.245v7 specification. The corrections will be applied to the next version of H.245.

[Begin Correction]

```
MultiplexedStreamParameter ::=SEQUENCE
{
   multiplexFormat          MultiplexFormat,
   controlOnMuxStream       BOOLEAN,
   ...
}
```

[End Correction]

6.3.2 Clarification of collapsing/nonCollapsing generic parameters ambiguity

Description:	There is a possible interoperability problem because the collapsing/nonCollapsing nature of generic parameters in Annexes F, J, K, and L has not been specified. The text added to the parameters below clarifies this.

[Begin Correction]

TABLE F.1/H.245

Flow Control Capability Parameter for Bit Rate Management

Parameter name:	Flow Control Capability
Parameter description:	This is a Collapsing GenericParameter. The presence of this parameter indicates the capability to support the FlowcontrolIndication message.
Parameter identifier value:	0
Parameter status:	Optional
Parameter type:	Logical.
Supersedes:	-

TABLE F.2/H.245

Logical Channel Bit Rate Change Capability Parameter for Bit Rate Management

Parameter name:	Logical Channel Bit Rate Change Capability

Parameter description:	This is a Collapsing GenericParameter. The presence of this parameter indicates the capability to support the Logical Channel Rate Change Procedure, which uses the messages LogicalChannelRateRequest, LogicalChannelRateAcknowledge, LogicalChannelRateReject and LogicalChannelRateRelease.
Parameter identifier value:	1
Parameter status:	Optional
Parameter type:	Logical.
Supersedes:	-

TABLE F.3/H.245

RTCP Frequency Parameter for Bit Rate Management

Parameter name:	RTCP Frequency Capability
Parameter description:	This is a Collapsing GenericParameter. This indicates the frequency at which the terminal can send RTCP reports.
Parameter identifier value:	2
Parameter status:	Optional
Parameter type:	Unsigned32Min.
Supersedes:	-

TABLE J.1/H.245

136 ACELP Capability Parameter - maxAl-sduFrames

Parameter name:	MaxAl-sduFrames
Parameter description:	This is a Collapsing GenericParameter. Specifies the maximum number of audio frames per AL-SDU
Parameter identifier value:	0
Parameter status:	Shall be present for capability exchange and logical channel signalling. Shall not be present for mode request.
Parameter type:	UnsignedMin

Supersedes:	-

TABLE J.2/H.245

136 ACELP Capability Parameter - Comfort Noise

Parameter name:	ComfortNoise
Parameter description:	This is a Collapsing GenericParameter. Specifies that TIA/EIA 136 (IS-641) comfort noise is to be used in mode request. This parameter shall be used only in mode requests but not in capabilities because this capability is mandatory.
Parameter identifier value:	1
Parameter status:	Optional
Parameter type:	Logical
Supersedes:	-

TABLE J.3/H.245

136 ACELP Capability Parameter - Scrambled

Parameter name:	Scrambled
Parameter description:	This is a Collapsing GenericParameter. Specifies that scrambling is to be used in mode request. This parameter shall be used only in mode requests but not in capabilities because this capability is mandatory.
Parameter identifier value:	2
Parameter status:	Optional
Parameter type:	Logical
Supersedes:	-

TABLE K.1/H.245

136 US1 Capability Parameter - maxAl-sduFrames

Parameter name:	MaxAl-sduFrames
Parameter description:	This is a Collapsing GenericParameter. Specifies the maximum number of audio frames per AL-SDU

Parameter identifier value:	0
Parameter status:	Shall be present for capability exchange and logical channel signalling. Shall not be present for mode request.
Parameter type:	UnsignedMin
Supersedes:	-

TABLE K.2/H.245

136 US1Capability Parameter - Comfort Noise

Parameter name:	ComfortNoise
Parameter description:	This is a Collapsing GenericParameter. Specifies that comfort noise is to be used in mode request. This parameter shall be used only in mode requests but not in capabilities because this capability is mandatory.
Parameter identifier value:	1
Parameter status:	Optional
Parameter type:	Logical
Supersedes:	-

TABLE K.3/H.245

136 US1 Capability Parameter - Scrambled

Parameter name:	Scrambled
Parameter description:	This is a Collapsing GenericParameter. Specifies that scrambling is to be used in mode request. This parameter shall be used only in mode requests but not in capabilities because this capability is mandatory.
Parameter identifier value:	2
Parameter status:	Optional
Parameter type:	Logical
Supersedes:	-

TABLE L.1/H.245

IS-127 CDMA EVRC Capability Parameter - maxAl-sduFrames

Parameter name:	MaxAl-sduFrames
Parameter description:	This is a Collapsing GenericParameter. Specifies the maximum number of audio frames per AL-SDU
Parameter identifier value:	0
Parameter status:	Shall be present for capability exchange and logical channel signalling. Shall not be present for mode request.
Parameter type:	UnsignedMin
Supersedes:	-

TABLE L.2/H.245

CDMA EVRC Capability Parameter - Scrambled

Parameter name:	Scrambled
Parameter description:	This is a Collapsing GenericParameter. Specifies that scrambling is to be used in mode request.
Parameter identifier value:	2
Parameter status:	Optional
Parameter type:	Logical
Supersedes:	-

[End Correction]

6.3.3 Annex B Section 3.1 Open Logical Channel

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

[Begin Correction]

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

[End Correction]

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

6.4.1 Annex A Corrections

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

[Begin Correction]

A.5.2.4.1 Multipoint Control C&I

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

[End Correction]

Description:	New H.243 codepoints MVC, MVA, and MVR were approved in February 2000. To support those new codepoints, the following additions shall be added to the table in A.5.2.4.1 as shown below

[Begin Correction]

A.5.2.4.1 Multipoint Control C&I

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

[End Correction]

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

[Begin Correction]

A.5.2.4.4 Multipoint Control C&I

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

[End Correction]

6.4.2 Reference to ATM Forum Document

Description:	To help clarify the usage of H.246 with respect to ATM, a reference to an ATM Forum document has been proposed. This reference shall appear in next H.246 publication from the ITU.

[Begin Correction]

1 Scope

...

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

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

[End Correction]

[Begin Correction]

2 Normative References

...

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

[End Correction]

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

6.5.1 Connection Establishment Procedures

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

Editorial - Clause 7.0

[Begin Correction]

7.1 Introduction

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

[End Correction]

6.5.2 Key Management

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

Editorial - Clause D.7.1 Key Management

[Begin Correction]

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

[End Correction]

6.5.3 Key Update and Synchronization

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

Editorial - Clause D.7.2 Key Update and Synchronization

[Begin Correction]

- **encryptedData**: set to the result of the encrypted **KeySynchMaterial**.

[End Correction]

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

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

6.6.1.1 Actions at a Destination Entity

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

[Begin Correction]

1) Section 6.6, line 6

   Change:

   "**rejectUnrecognizedInvokePdu**"

   to

   "**rejectAnyUnrecognizedInvokePdu**"

2) Section 6.6, line 12

Change:

"**discardAnyUnrecognizedInvokePDU**"

to

"**discardAnyUnrecognizedInvokePdu**"

[End Correction]

6.6.1.2 Corrections to the ASN.1

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

[Begin Correction]

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

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

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

                                        ...

    ScreeningIndicator                  ::=     ENUMERATED {

            userProvidedNotScreened (0),
                        number was provided by a remote user
```

~~*, and has not been screened by a gatekeeper*~~

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

~~userProvidedVerifiedAndFailed (2),~~
~~*not used, value reserved.*~~

~~networkProvided (3),~~
~~*number was provided by a gatekeeper*~~

~~…~~ ~~}~~

[End Correction]

6.6.1.3 Clarifications to ROS APDUs

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

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

[Begin Correction]

Note:

In the *Invoke* APDU, the *invokeID* is an INTEGER constrained by a PER-visible constraint (InvokeIdSet = 0..65535) and is therefore encoded as a **constrained** INTEGER (16 bits, no length field). In the *ReturnResult* and *ReturnError* APDUs, however, the *invokeID* is encoded as an **unconstrained** INTEGER (with explicit length field) because the applicable constraint ("must be that for an outstanding operation…") is not PER-visible. In the *Reject* APDU the *invokeID* is also encoded as an **unconstrained** INTEGER (with explicit length field) since no constraint applies.

[End Correction]

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

6.6.2.1 Editorial Corrections

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

[Begin Correction]

1) Editorial - Clause 11.4.2, line 4 c)

Change:

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

to

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

2) Editorial - Clause 11.4.2, line 5 d)

Change:

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

to

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

3) Editorial - Clause 11.5.2, line 6 e)

Change:

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

to

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

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

Change:

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

to

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

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

Change:

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

to

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

6) Editorial - Clause 11.6.2, line 2

Change:

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

to

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

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

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

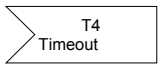
Change:

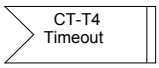
"T4 Timeout"

to

"CT-T4 Timeout"

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

change 

to 

[End Correction]

6.6.2.2 Clarification of CallIdentifier and ConferenceIdentifier

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

[Begin Correction]

10.7 Interactions with H.225.0 parameters

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

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

[End Correction]

6.6.2.3 Transfer without Consultation

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

[Begin Correction]

6 Messages and Information elements

...

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

[End Correction]

[Begin Correction]

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

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

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

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

[End Correction]

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

6.6.3.1 Editorial Correction in H.450.3

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

[Begin Correction]

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

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

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

Note: The text within the referred symbols remains unchanged.

change  to 

[End Correction]

6.6.3.2 Clarification of the CallIdentifier and ConferenceIdentifier

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

[Begin Correction]

9.9.3 Interactions with H.225.0 parameters

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

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

[End Correction]

6.6.3.3 Correction to the ASN.1

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

[Begin Correction]

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

[End Correction]

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

6.6.4.1 Change Relating to Interpretation APDU

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

[Begin Correction]

6 Messages and Information elements

...

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

[End Correction]

6.6.4.2 Feature Interaction between H.450.4 and H.450.2

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

[Begin Correction]

9.2.1 Call Transfer (H.450.2)

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

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

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

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

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

[End Correction]

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

6.6.5.1 Clarification of the CallIdentifier

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

[Begin Correction]

8.3 Interaction with H.225.0 parameters

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

[End Correction]

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

There are no corrections for H.450.6.

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

6.6.7.1 Change Relating to Interpretation APDU

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

[Begin Correction]

7.1.1 H.450.1 Supplementary Service APDU

...

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

[End Correction]

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

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

[End Correction]

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

6.7.2 Support for Expanded Country Code Values in T.35

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

[Begin Correction]

```
h323TermSystemt35CountryCode OBJECT-TYPE
        SYNTAX INTEGER (0..255)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "Country code, per T.35 Annex A."
::= { h323TermSystemEntry 5 }
h323TermSystemt35CountryCodeExtention OBJECT-TYPE
        SYNTAX INTEGER (0..255)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "Assigned nationally, unless the country code
            is 255, in which case this value shall contain
            the country code found in T.35 Annex B."
::= { h323TermSystemEntry 6 }
```

[End Correction]

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

	Transport address = GW_{B1} CALL SIGNALLING address Message type = sendSetup	
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 Message type = sendAccessRequest	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.

[End Correction]

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

[Begin Correction]

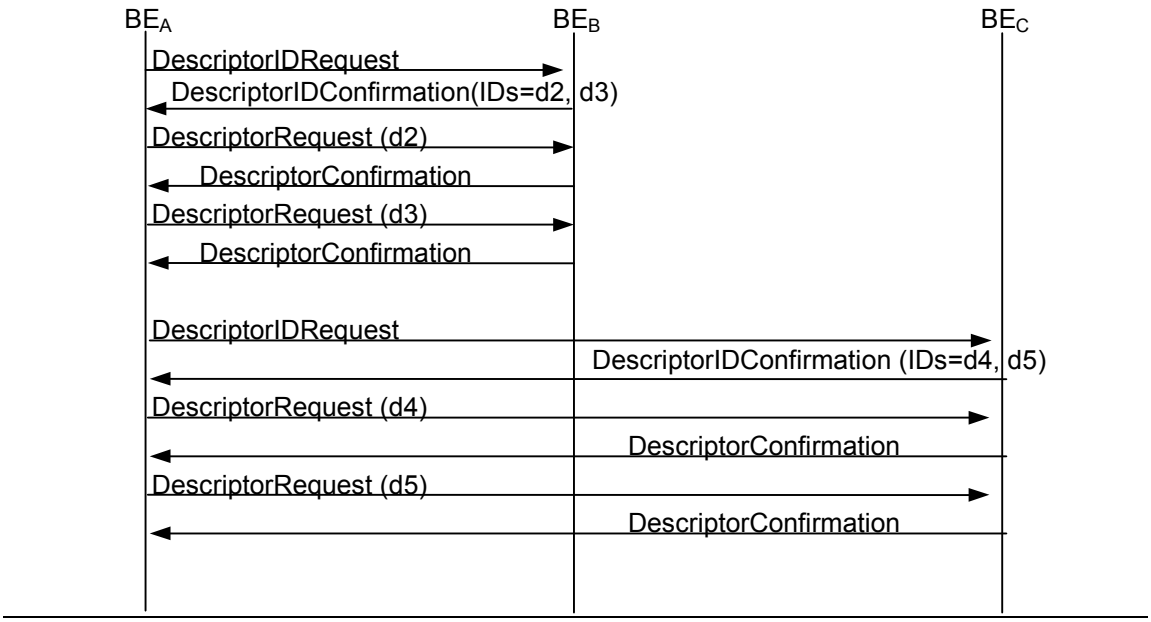


Figure G.8/H.225.0 - Example of Descriptor Exchange

[End Correction]

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

[Begin Correction]

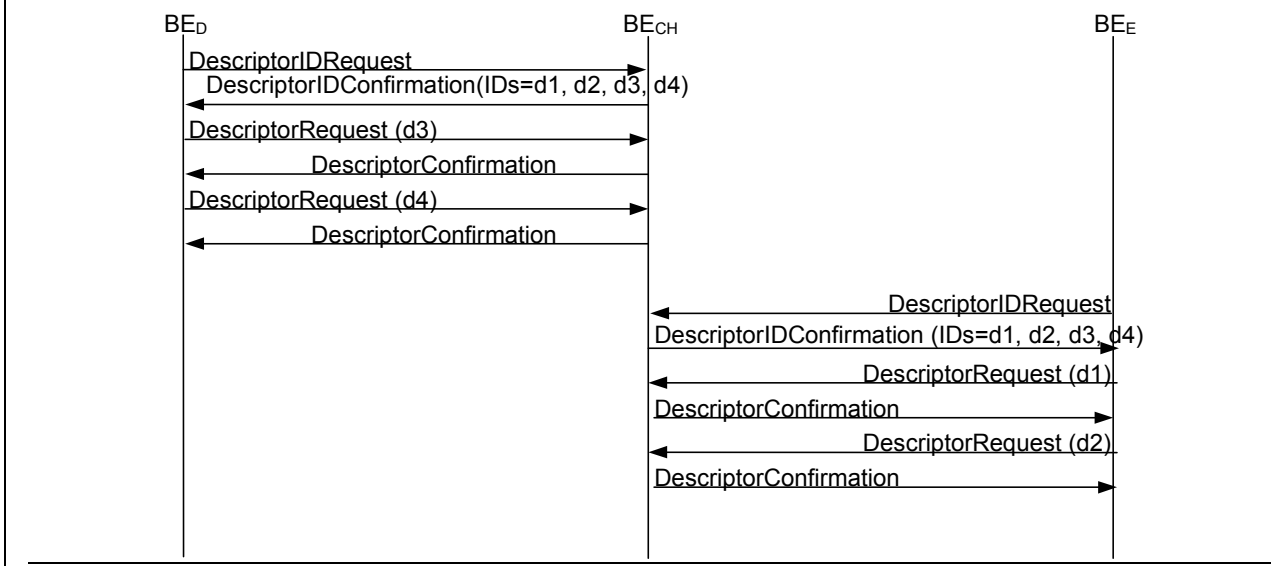
Administrative Domain	Template definition	Comment
D	Descriptor "d1": Pattern = 1908*	For calls to 1908*, an Access Request message is needed to get the destination's (i.e., a gateway) call signaling address.

	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 1908953*, the Setup can be sent directly to this particular gateway.
E	Descriptor "~~d1~~d3": Pattern = 1303538* 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 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.
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]

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

6.9.1 Additional Message Mappings

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

[Begin Correction]

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

[End Correction]

6.9.2 Changes for Call Diversion

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

[Begin Correction]

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

[End Correction]

6.9.3 Redirecting Number Replaced with Call Diversion and Redirection Number

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

[Begin Correction]

~~Redirecting number~~

~~NA~~

Call diversion information

See C.6.2.6

Redirection number restriction

See C.6.2.6

[End Correction]

6.9.4 Call Diversion with and without H.450.3

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

[Begin Correction]

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

~~For further study.~~

C.7.2.8.3.1 Gateways supporting H.450.3

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

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

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

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

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

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

C.7.2.8.3.2 Gateways not supporting H.450.3

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

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

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

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

IAM ->	SETUP ->
Redirecting number parameter Nature of address (1) Numbering plan (2) Address signal (3)	Redirecting number information element Type of number (6) Numbering plan (7)

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

6.9.5 New Release Complete / Cause Mappings

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

[Begin Correction]

Table 15/ANNEX C – Call clearing from the user

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

Table 52/ANNEX C – Call clearing during call establishment

←REL	**←RELEASE COMPLETE**
Cause parameter	Cause information element
Cause value No. x (Notes 1)	Cause value No. x
Cause parameter	ReleaseCompleteReason

47 – Resource Unavailable	newConnectionNeeded
127 – Interworking, unspecified	nonStandardReason
31 – Normal, unspecified	replaceWithConferenceInvite

[End Correction]

6.9.6 Single 64kbps Bearer FFS in Table 3

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

[Begin Correction]

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

SETUP→		**IAM→**
Bearer capability information element		Transmission medium
Information transfer capability	Information transfer rate	requirement parameter
Speech	Value non-significant	*Speech*
3.1 kHz audio	Value non-significant	*3.1 kHz audio*
Restricted digital information	For further studies	For further studies
Unrestricted digital information Or	*64 kbit/s unrestricted*	~~*3.1 kHz audio*~~ *FFS*
	2 × 64 kbit/s unrestricted	*2 × 64 kbit/s*
	384 kbit/s unrestricted	*384 kbit/s*
	1536 kbit/s unrestricted	*1536 kbit/s*
	1920 kbit/s unrestricted	*1920 kbit/s*
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 M, K and I. The corrected text is shown below.

[Begin Correction]

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

...

Backward call indicators

...

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

bit K ISDN user part indicator

1 *ISDN user part used all the way*

~~If bit I is 0 then:~~

bit M ISDN access indicator

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

[End Correction]

6.9.11 Clarification of Cut-Through Behavior

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

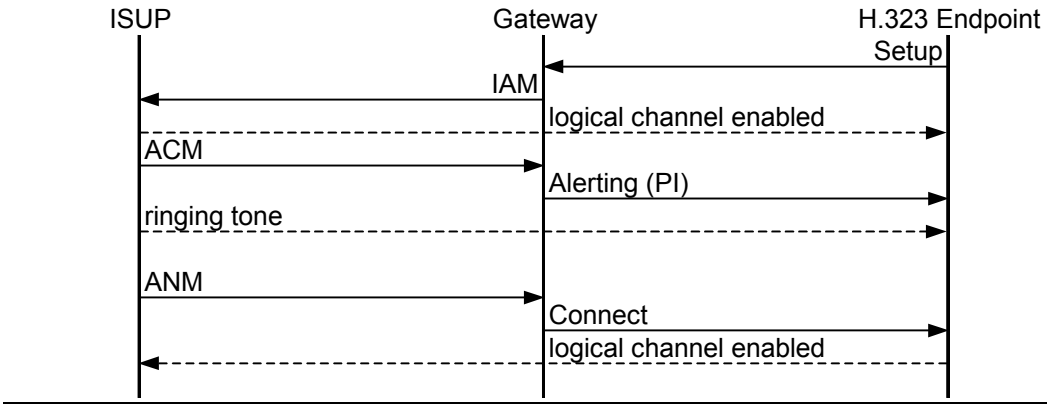
[Begin Correction]

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

In traditional telephone networks, cut-through occurs very early in the call (before the called party answers) to provide tones or announcements, and to eliminate clipping on answer while

the voice channel is being connected end-to-end. Section 8.1.7.4/H.323 describes the behavior for early cut-through (that is, cut-through before the H.225 Connect message).

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



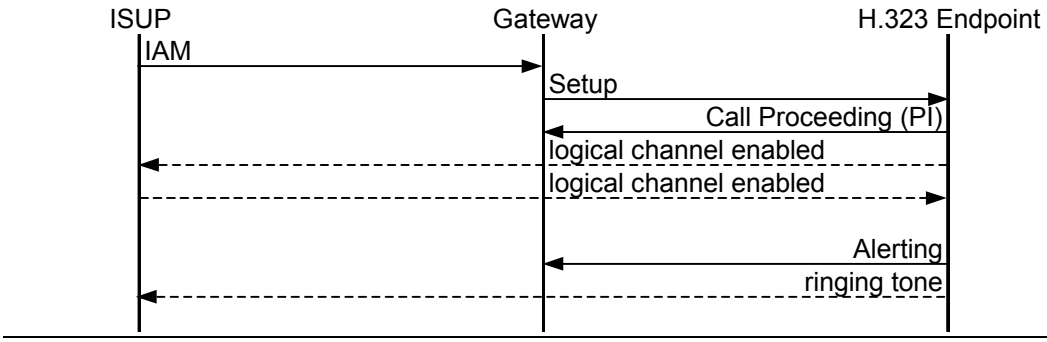
The notation "Alerting (PI)" indicates the presence of the progress indicator as described in section 8.1.7.4/H.323.

[End Correction]

[Begin Correction]

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

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



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

[End Correction]

6.10 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)
Annex J	1998 (Version 2)	N/A
Annex M.1	2000 (Version 4)	N/A
Annex M.2	2000 (Version 4)	N/A

7.12 Routing through Signaling Entities and Detecting Loops

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

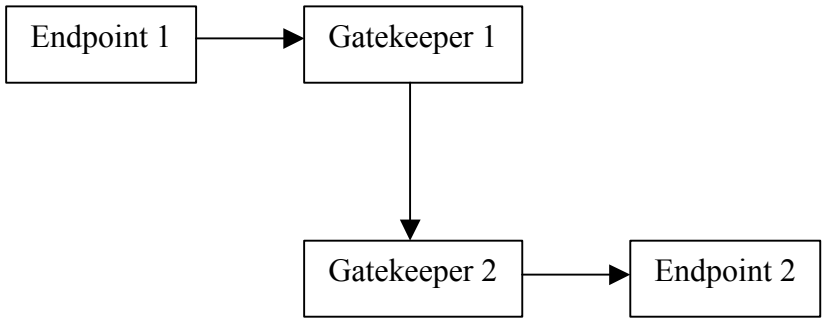


Figure 1 - Call placed through multiple gatekeepers

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

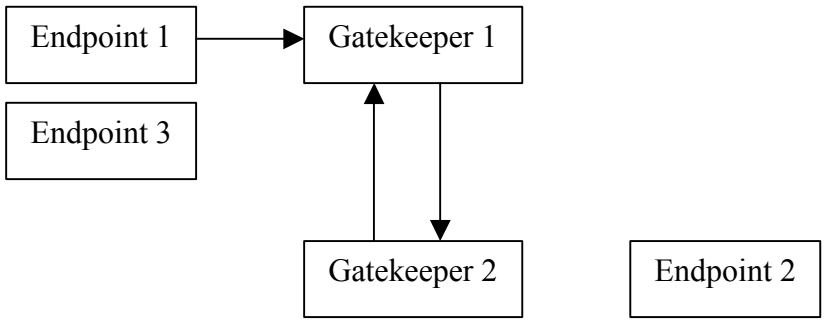


Figure 2 - Gatekeeper 2 re-routes call back to Gatekeeper 1

When GK1 receives the Setup message from GK2, it may inadvertently mistake the call as "bogus", since the Call Identifier will match an already existing call within the Gatekeeper. Implementers should consider this type of call scenario and be prepared to receive incoming calls that contain Call Identifiers for calls that are already being routed through the routing entity. The routing entity

should examine not only the Call Identifier, but also the destination address of the call (the call signaling address, aliases, or Called Party Number of the destination). In this case, the call is routed through GK1 with a destination address of EP2 is rerouted by GK2 to GK1, but with a destination address of EP3. In this way, the GK1 will properly handle call routing and rerouting, as well as prevent loops in the call signaling path.

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

The logic for Version 1 devices seems similar to that for Version 2 and higher devices, but there are issues when EP2 and EP3 are MCUs, for example. Suppose that EP2 is an MCU that is directing all calls to EP3. The first time a call is redirected to GK1, GK1 may realize that this is, indeed, a call redirection as described above. However, when the second call is redirected, GK1 has no means of distinguishing between the first redirected call and the second: the source address *may* be the same, the destination address is the same as the previously rerouted call (EP3), and the Conference ID is the same. So in this case, GK1 may have no choice but to assume that a loop has occurred and release the offending call. Although this is unfortunate, H.323v2 and higher systems do not suffer from this problem. What is important, though, is that loop detection is possible—even with version 1 systems.

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

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

8 Allocated Object Identifiers and Port Numbers

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

8.1 Allocated Object Identifiers

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

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

{ 0 0 h(8) 2250 version(0) *[v]* } H225.0 version numbers

Assigned values of *v*: 1-3

{ 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

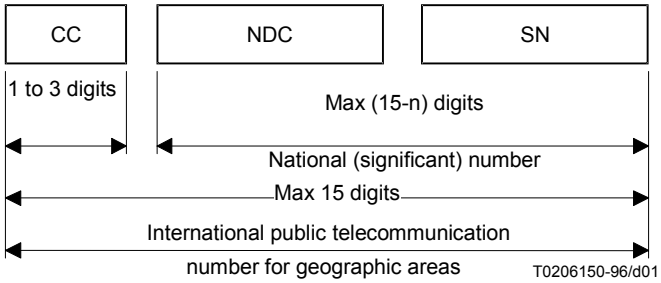
The following IP port numbers have been allocated for various components of H.323:

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 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.

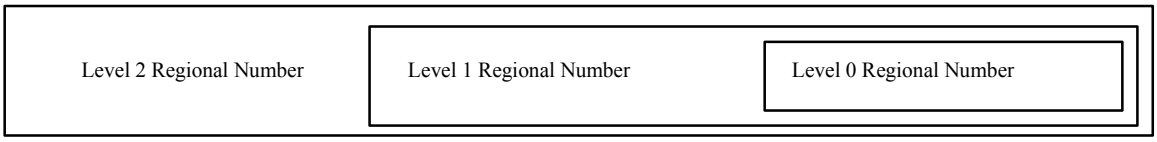


Figure – H.323 - Structure of a PNP Number with three levels of regions

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