INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.323

(07/2003)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Systems and terminal equipment for audiovisual services

Packet-based multimedia communications systems

ITU-T Recommendation H.323

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
SYSTEMS AND TERMINAL EQUIPMENT FOR AUDIOVISUAL SERVICES	**H.300–H.399**
SUPPLEMENTARY SERVICES FOR MULTIMEDIA	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND AND TRIPLE-PLAY MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation H.323

Packet-based multimedia communications systems

Summary

This Recommendation describes terminals and other entities that provide multimedia communications services over Packet Based Networks (PBN) which may not provide a guaranteed Quality of Service. H.323 entities may provide real-time audio, video and/or data communications. Support for audio is mandatory, while data and video are optional, but if supported, the ability to use a specified common mode of operation is required, so that all terminals supporting that media type can interwork.

The packet based network over which H.323 entities communicate may be a point-to-point connection, a single network segment, or an internetwork having multiple segments with complex topologies.

H.323 entities may be used in point-to-point, multipoint, or broadcast (as described in ITU-T Rec. H.332) configurations. They may interwork with H.310 terminals on B-ISDN, H.320 terminals on N-ISDN, H.321 terminals on B-ISDN, H.322 terminals on Guaranteed Quality of Service LANs, H.324 terminals on GSTN and wireless networks, V.70 terminals on GSTN, and voice terminals on GSTN or ISDN through the use of Gateways.

H.323 entities may be integrated into personal computers or implemented in stand-alone devices such as videotelephones.

Note that the title of H.323 (1996) was "Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service". The title changed in Version 2 to be consistent with its expanded scope.

Products claiming compliance with Version 1 of H.323 shall comply with all of the mandatory requirements of H.323 (1996) which references ITU-T Recs H.225.0 (1996) and H.245 (1997). Version 1 products shall be identified by H.225.0 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 2250 version (0) 1} and H.245 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 245 version (0) 2}.

Products claiming compliance with Version 2 of H.323 shall comply with all of the mandatory requirements of this Recommendation, H.323 (1998), which references ITU-T Recs H.225.0 (1998) and H.245 (1998 or later). Version 2 products shall be identified by H.225.0 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 2250 version (0) 2} and H.245 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 245 version (0) x}, where "x" is 3 or higher.

Products claiming compliance with Version 3 of H.323 shall comply with all of the mandatory requirements of this Recommendation, H.323 (1999), which references ITU-T Recs H.225.0 (1999) and H.245 (1999 or later). Version 3 products shall be identified by H.225.0 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 2250 version (0) 3} and H.245 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 245 version (0) x}, where "x" is 5 or higher.

Products claiming compliance with Version 4 of H.323 shall comply with all of the mandatory requirements of this Recommendation, H.323 (2000), which references ITU-T Recs H.225.0 (2000) and H.245 (2000 or later). Version 4 products shall be identified by H.225.0 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 2250 version (0) 4} and H.245 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 245 version (0) x}, where "x" is 7 or higher.

Products claiming compliance with Version 5 of H.323 shall comply with all of the mandatory requirements of this Recommendation, H.323 (2003), which references ITU-T Recs H.225.0 (2003) and H.245 (02/2003 or later). Version 5 products shall be identified by H.225.0 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 2250 version (0) 5} and H.245 messages containing a **protocolIdentifier** = {itu-t (0) recommendation (0) h (8) 245 version (0) x}, where "x" is 9 or higher.

This version of ITU-T Rec. H.323 integrates without further modifications Annexes M3 (07/2001), P (01/2003), Q (07/2001) and R (07/2001), as well as Annex O, approved independently 07/2003.

Source

ITU-T Recommendation H.323 was approved by ITU-T Study Group 16 (2001-2004) under the ITU-T Recommendation A.8 procedure on 14 July 2003.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation H.323

Packet-based multimedia communications systems

1 Scope

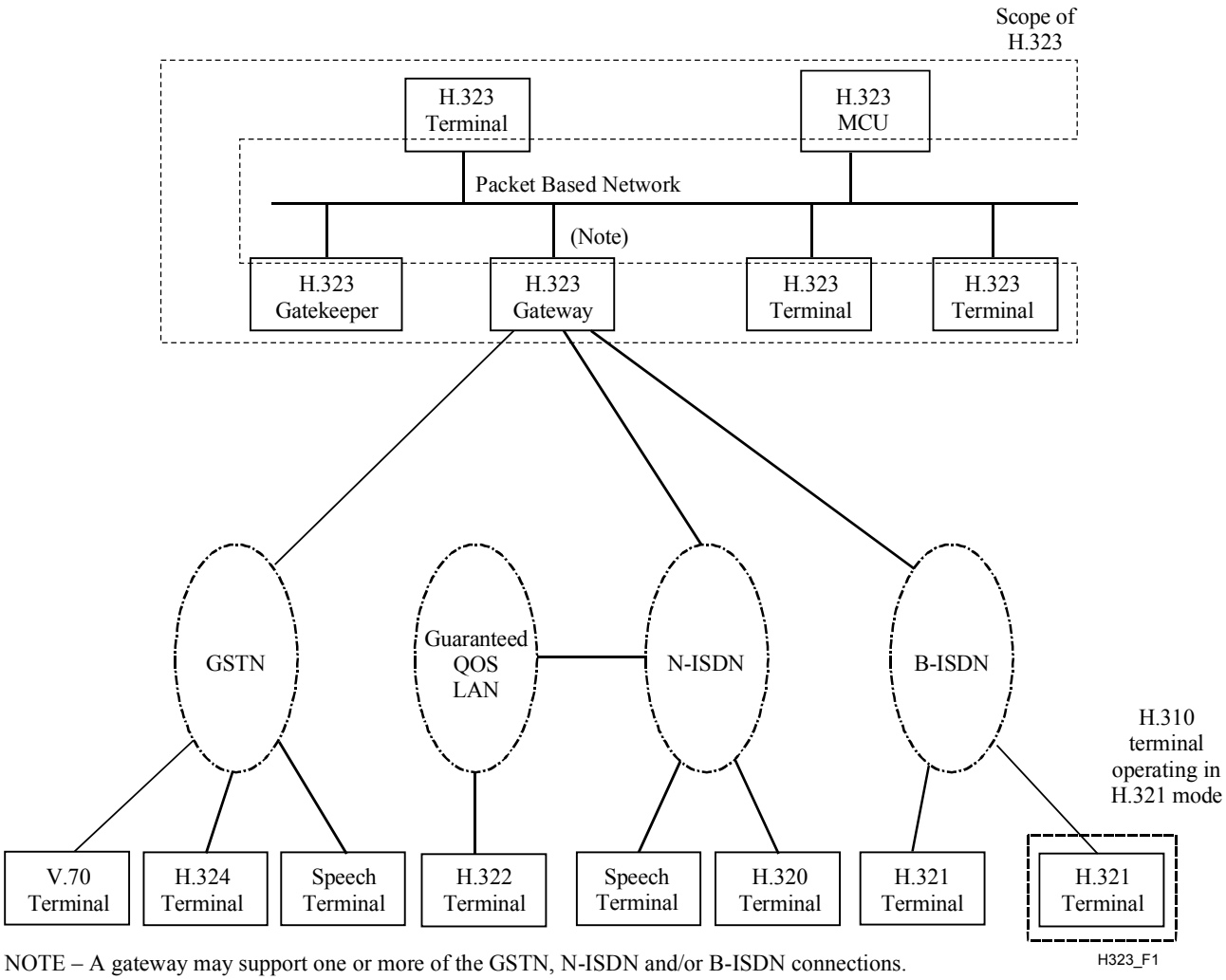
This Recommendation covers the technical requirements for multimedia communications systems in those situations where the underlying transport is a Packet Based Network (PBN) which may not provide a guaranteed Quality of Service (QOS). These packet based networks may include Local Area Networks, Enterprise Area Networks, Metropolitan Area Networks, Intra-Networks and Inter-Networks (including the Internet). They also include dial up connections or point-to-point connections over the GSTN or ISDN which use an underlying packet based transport such as PPP. These networks may consist of a single network segment, or they may have complex topologies which incorporate many network segments interconnected by other communications links.

This Recommendation describes the components of an H.323 system. This includes Terminals, Gateways, Gatekeepers, Multipoint Controllers, Multipoint Processors and Multipoint Control Units. Control messages and procedures within this Recommendation define how these components communicate. Detailed descriptions of these components are contained in clause 6.

H.323 terminals provide audio and optionally video and data communications capability in point-to-point or multipoint conferences. Interworking with other H-series terminals, GSTN or ISDN voice terminals, or GSTN or ISDN data terminals is accomplished using Gateways. See Figure 1. Gatekeepers provide admission control and address translation services. Multipoint Controllers, Multipoint Processors and Multipoint Control Units provide support for multipoint conferences.

The scope of H.323 does not include the network interface, the physical network or the transport protocol used on the network. Examples of these networks include but are not limited to:

– Ethernet (IEEE 802.3);
– Fast Ethernet (IEEE 802.3u);
– FDDI;
– Token Ring (IEEE 802.5);
– ATM.



NOTE – A gateway may support one or more of the GSTN, N-ISDN and/or B-ISDN connections.

Figure 1/H.323 – Interoperability of H.323 terminals

2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[1] ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

[2] ITU-T Recommendation H.245 (2003), *Control protocol for multimedia communication*.

[3] ITU-T Recommendation G.711 (1988), *Pulse Code Modulation (PCM) of voice frequencies*.

[4] ITU-T Recommendation G.722 (1988), *7 kHz audio-coding within 64 kbit/s*.

[5] ITU-T Recommendation G.723.1 (1996), *Speech coders: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s*.

[6] ITU-T Recommendation G.728 (1992), *Coding of speech at 16 kbit/s using low-delay code excited linear prediction*.

[7] ITU-T Recommendation G.729 (1996), *Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)*.

[8] ITU-T Recommendation H.261 (1993), *Video codec for audiovisual services at p × 64 kbit/s*.

[9] ITU-T Recommendation H.263 (1998), *Video coding for low bit rate communication*.

[10] ITU-T Recommendation T.120 (1996), *Data protocols for multimedia conferencing*.

[11] ITU-T Recommendation H.320 (1999), *Narrow-band visual telephone systems and terminal equipment*.

[12] ITU-T Recommendation H.321 (1998), *Adaptation of H.320 visual telephone terminals to B-ISDN environments*.

[13] ITU-T Recommendation H.322 (1996), *Visual telephone systems and terminal equipment for local area networks which provide a guaranteed quality of service*.

[14] ITU-T Recommendation H.324 (2002), *Terminal for low bit-rate multimedia communication*.

[15] ITU-T Recommendation H.310 (1998), *Broadband audiovisual communication systems and terminals*.

[16] ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*.

[17] ITU-T Recommendation Q.932 (1998), *Digital subscriber signalling system No. 1 – Generic procedures for the control of ISDN supplementary services*.

[18] ITU-T Recommendation Q.950 (2000), *Supplementary services protocols, structure and general principles*.

[19] ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (USC) – Part 1: Architecture and basic multilingual plane*.

[20] ITU-T Recommendation E.164 (1997), *The international public telecommunication numbering plan*.

[21] 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*.

[22] ITU-T Recommendation H.235 (2003), *Security and encryption for H-series (H.323 and other H.245-based) multimedia terminals*.

[23] ITU-T Recommendation H.332 (1998), *H.323 extended for loosely coupled conferences*.

[24] ITU-T Recommendation H.450.1 (1998), *Generic functional protocol for the support of supplementary services in H.323*.

[25] ITU-T Recommendation I.363.5 (1996), *B-ISDN ATM Adaptation Layer specification: Type 5 AAL*.

[26] ITU-T Recommendation Q.2931 (1995), *Digital subscriber signalling system No. 2 – User-network interface (UNI) layer 3 specification for basic call/connection control*.

[27] ITU-T Recommendation I.356 (2000), *B-ISDN ATM layer cell transfer performance*.

[28] ITU-T Recommendation I.371 (2000), *Traffic control and congestion control in B-ISDN*.

[29] ITU-T Recommendation I.371.1 (2000), *Guaranteed frame rate ATM transfer capability*.

[30] ITU-T Recommendation Q.2961.2 (1997), *Digital subscriber signalling system No. 2 – Additional traffic parameters: Support of ATM transfer capability in the broadband bearer capability information element*.

[31] ITU-T Recommendation H.282 (1999), *Remote device control protocol for multimedia applications*.

[32] ITU-T Recommendation H.283 (1999), *Remote device control logical channel transport*.

[33] ATM Forum AF-SAA-0124.000 (1999), *H.323 Media Transport Over ATM*.

[34] ITU-T Recommendation Q.2941.2 (1999), *Digital subscriber signalling system No. 2 – Generic identifier transport extensions*.

[35] ITU-T Recommendation H.450.2 (1998), *Call transfer supplementary service for H.323*.

[36] ITU-T Recommendation H.450.4 (1999), *Call hold supplementary service for H.323*.

[37] ITU-T Recommendation H.248 (2000), *Gateway control protocol*.

[38] ISO/IEC 11571:1998, *Information technology – Telecommunications and information exchange between systems – Private Integrated Services Networks – Addressing*.

[39] ITU-T Q.951.x family Recommendations, *Stage 3 description for number identification supplementary services using DSS 1*.

[40] ITU-T Recommendation H.450.3 (1998), *Call diversion supplementary service for H.323*.

[41] ITU-T Recommendation H.450.5 (1999), *Call park and call pickup supplementary services for H.323*.

[42] ITU-T Recommendation H.450.6 (1999), *Call waiting supplementary service for H.323*.

[43] ITU-T Recommendation H.450.7 (1999), *Message waiting indication supplementary service for H.323*.

[44] ITU-T Recommendation H.450.8 (2000), *Name identification supplementary service for H.323*.

[45] ISO/IEC 11572:2000, *Information technology – Telecommunications and information exchange between systems – Private Integrated Services Network – Circuit mode bearer services – Inter-exchange signalling procedures and protocol*.

[46] ITU-T Recommendation H.222.0 (2000), *Information technology – Generic coding of moving pictures and associated audio information: Systems*.

[47] ITU-T Recommendation H.223 (2001), *Multiplexing protocol for low bit rate multimedia communication*.

[48] IETF RFC 2068 (1997), *Hypertext Transfer Protocol – HTTP/1.1*.

[49] IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*.

[50] ITU-T Recommendation Z.100 (2002), *Specification and Description Language (SDL)*.

[51] IETF RFC 1738 (1994), *Uniform Resource Locators (URL)*.

[52] IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF*.

[53] ISO 4217:2001, *Codes for the representation of currencies and funds*.

[54] ITU-T Recommendation V.21 (1988), *300 bits per second duplex modem standardized for use in the general switched telephone network*.

[55] ITU-T Recommendation T.30 (2003), *Procedures for document facsimile transmission in the general switched telephone network*.

[56] ITU-T Recommendation T.38 (2002), *Procedures for real-time Group 3 facsimile communication over IP networks*.

[57] ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.

[58] IETF RFC 2833 (2000), *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*.

3 Definitions

For the purposes of this Recommendation the definitions given in clause 3/H.225.0 [1] and clause 3/H.245 [2] apply along with those in this clause. These definitions apply to the packet based network side only. Other terms may be appropriate when referring to the Switched Circuit Network (SCN) side. See clause 5, Conventions, for information on the use of terms in this Recommendation.

3.1 access gateway: A Gateway that connects one network to another network (such as an SS7 network to a QSIG network) and performs some interworking function between the different networks.

3.2 active MC: An MC that has won the master-slave determination procedure and is currently providing the multipoint control function for the conference.

3.3 ad hoc multipoint conference: An Ad Hoc Multipoint conference was a point-to-point conference that had been expanded into a multipoint conference at some time during the call. This can be done if one or more of the terminals in the initial point-to-point conference contains an MC, if the call is made using a Gatekeeper that includes MC functionality, or if the initial call is made through an MCU as a multipoint call between only two terminals.

3.4 addressable: An H.323 entity on the network having a Transport Address is addressable. Not the same as being callable. A terminal, Gateway, or MCU is addressable and callable. A Gatekeeper is addressable but not callable. An MC or MP is neither callable nor addressable but is contained within an endpoint or Gatekeeper that is. In a composite Gateway, both the MGC and the MG are addressable, but only the MGC is callable.

3.5 audio mute: Suppressing of the audio signal of a single or all source(s). Send muting means that the originator of an audio stream mutes its microphone and/or does not transmit any audio signal at all. Receive mute means that the receiving terminal ignores a particular incoming audio stream or mutes its loudspeaker.

3.6 broadcast conference: A Broadcast conference is one in which there is one transmitter of media streams and many receivers. There is no bidirectional transmission of control or media streams. Such conferences should be implemented using network transport multicast facilities, if available. Also see ITU-T Rec. H.332 [23].

3.7 broadcast panel conference: A Broadcast Panel conference is a combination of a Multipoint conference and a Broadcast conference. In this conference, several terminals are engaged in a multipoint conference, while many other terminals are only receiving the media streams. There is bidirectional transmission between the terminals in the multipoint portion of the conference and no bidirectional transmission between them and the listening terminals. Also see ITU-T Rec. H.332.

3.8 call: Point-to-point multimedia communication between two H.323 endpoints. The call begins with the call set-up procedure and ends with the call termination procedure. The call consists of the collection of reliable and unreliable channels between the endpoints. A call may be directly

between two endpoints or may include other H.323 entities such as a Gatekeeper or MC. In case of interworking with some SCN endpoints via a Gateway, all the channels terminate at the Gateway where they are converted to the appropriate representation for the SCN end system. Typically, a call is between two users for the purpose of communication, but may include signalling-only calls. An endpoint may be capable of supporting multiple simultaneous calls.

3.9 call signalling channel: Reliable channel used to convey the call setup and teardown messages (following ITU-T Rec. H.225.0) between two H.323 entities.

3.10 callable: Capable of being called as described in clause 8 or in the supplementary services ITU-T (H.450.x). In other words, an H.323 entity is generally considered callable if a user would specify the entity as a destination. Terminals, MCUs, Gateways, and MGCs are callable, but Gatekeepers, MCs and MGs are not.

3.11 centralized multipoint conference: A Centralized Multipoint conference is one in which all participating terminals communicate in a point-to-point fashion with an MCU. The terminals transmit their control, audio, video, and/or data streams to the MCU. The MC within the MCU centrally manages the conference. The MP within the MCU processes the audio, video and/or data streams and returns the processed streams to each terminal.

3.12 composite gateway: A Gateway that does not separate the Media Gateway Controller and Media Gateway functions.

3.13 control and indication: End-to-end signalling between terminals, consisting of Control, which causes a state change in the receiver, and Indication which provides for information as to the state or functioning of the system (see also ITU-T Rec. H.245 [2] for additional information and abbreviations).

3.14 data: Information stream other than audio, video, and control, carried in the logical data channel (see ITU-T Rec. H.225.0 [1]).

3.15 decentralized multipoint conference: A Decentralized Multipoint conference is one in which the participating terminals multicast their audio and video to all other participating terminals without using an MCU. The terminals are responsible for:

a) summing the received audio streams; and

b) selecting one or more of the received video streams for display.

No audio or video MP is required in this case. The terminals communicate on their H.245 Control Channels with an MC which manages the conference. The data stream is still centrally processed by the MCS-MCU which may be within an MP.

3.16 decomposed gateway: A Gateway that is functionally separated into a Media Gateway Controller and one or more Media Gateways.

3.17 endpoint: An H.323 terminal, Gateway, or MCU. An endpoint can call and be called. It generates and/or terminates information streams.

3.18 gatekeeper: The Gatekeeper (GK) is an H.323 entity on the network that provides address translation and controls access to the network for H.323 terminals, Gateways and MCUs. The Gatekeeper may also provide other services to the terminals, Gateways and MCUs such as bandwidth management and locating Gateways.

3.19 gateway: An H.323 Gateway (GW) is an endpoint on the network which provides for real-time, two-way communications between H.323 Terminals on the packet based network and other ITU Terminals on a switched circuit network or to another H.323 Gateway. Other ITU Terminals include those complying with ITU-T Rec. H.310 (H.320 on B-ISDN), H.320 (ISDN), H.321 (ATM), H.322 (GQOS-LAN), H.324 (GSTN), H.324M (Mobile), and V.70 (DSVD).

3.20 H.323 entity: Any H.323 component, including terminals, Gateways, Gatekeepers, MCs, MPs and MCUs.

3.21 H.245 control channel: Reliable Channel used to carry the H.245 control information messages (following ITU-T Rec. H.245) between two H.323 endpoints.

3.22 H.245 session: The part of the call that begins with the establishment of an H.245 Control Channel and ends with the receipt of the H.245 **EndSessionCommand** or termination due to failure. Not to be confused with a call, which is delineated by the H.225.0 Setup and Release Complete messages.

3.23 hybrid multipoint conference – centralized audio: A Hybrid Multipoint – Centralized Audio conference is one in which terminals multicast their video to other participating terminals and unicast their audio to the MP for mixing. The MP returns a mixed audio stream to each terminal.

3.24 hybrid multipoint conference – centralized video: A Hybrid Multipoint – Centralized Video conference is one in which terminals multicast their audio to other participating terminals and unicast their video to the MP for switching or mixing. The MP returns a video stream to each terminal.

3.25 information stream: A flow of information of a specific media type (e.g., audio) from a single source to one or more destinations.

3.26 lip synchronization: Operation to provide the feeling that speaking motion of the displayed person is synchronized with his speech.

3.27 local area network (LAN): A shared or switched medium, peer-to-peer communications network that broadcasts information for all stations to receive within a moderate-sized geographic area, such as a single office building or a campus. The network is generally owned, used and operated by a single organization. In the context of this Recommendation, LANs also include internetworks composed of several LANs that are interconnected by bridges or routers.

3.28 logical channel: Channel used to carry the information streams between two H.323 endpoints. These channels are established following the H.245 **OpenLogicalChannel** procedures. An unreliable channel is used for audio, audio control, video and video control information streams. A reliable channel is used for data and H.245 control information streams. There is no relationship between a logical channel and a physical channel.

3.29 media gateway: The Media Gateway converts media provided in one type of network to the format required in another type of network. For example, an MG could terminate bearer channels from a switched circuit network (i.e., DS0s) and media streams from a packet network (e.g., RTP streams in an IP network). This Gateway may be capable of processing audio, video and T.120 alone or in any combination, and will be capable of full duplex media translations. The MG may also play audio/video messages and perform other IVR functions or may perform media conferencing.

3.30 media gateway controller: Controls the parts of the call state that pertain to connection control for media channels in an MG.

3.31 mixed multipoint conference: A Mixed Multipoint conference (see Figure 2) has some terminals (D, E and F) participating in a centralized mode while other terminals (A, B and C) are participating in a decentralized mode. A terminal is not aware of the mixed nature of the conference, only of the type of conference it is participating in. The MCU provides the bridge between the two types of conferences.

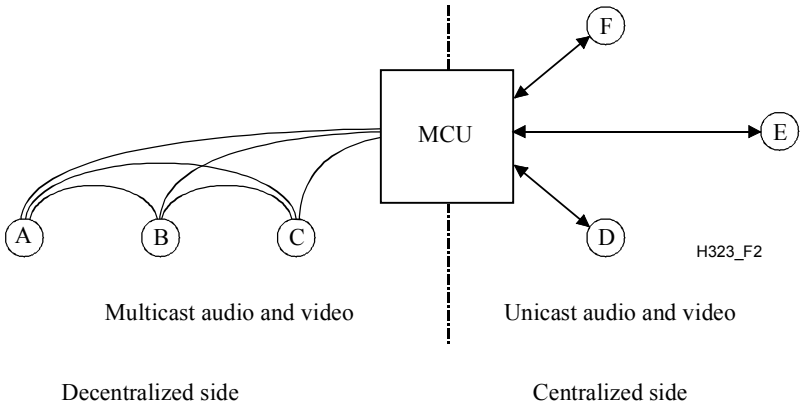


Figure 2/H.323 – Mixed multipoint conference

3.32 multicast: A process of transmitting PDUs from one source to many destinations. The actual mechanism (i.e., IP multicast, multi-unicast, etc.) for this process may be different for different network technologies.

3.33 multipoint conference: A Multipoint conference is a conference between three or more terminals. The terminals may be on the network or on the SCN. The multipoint conference shall always be controlled by an MC. Various multipoint conference types are defined in this subclause but they all require a single MC per conference. They may also involve one or more H.231 MCUs on the SCN. A terminal on the network may also participate in an SCN multipoint conference by connecting via a Gateway to an SCN-MCU. This does not require the use of an MC.

3.34 multipoint control unit: The Multipoint Control Unit (MCU) is an endpoint on the network which provides the capability for three or more terminals and Gateways to participate in a multipoint conference. It may also connect two terminals in a point-to-point conference which may later develop into a multipoint conference. The MCU generally operates in the fashion of an H.231 MCU; however, an audio processor is not mandatory. The MCU consists of two parts: a mandatory Multipoint Controller and optional Multipoint Processors. In the simplest case, an MCU may consist only of an MC with no MPs. An MCU may also be brought into a conference by the Gatekeeper without being explicitly called by one of the endpoints.

3.35 multipoint controller: The Multipoint Controller (MC) is an H.323 entity on the network which provides for the control of three or more terminals participating in a multipoint conference. It may also connect two terminals in a point-to-point conference which may later develop into a multipoint conference. The MC provides for capability negotiation with all terminals to achieve common levels of communications. It may also control conference resources such as who is multicasting video. The MC does not perform mixing or switching of audio, video and data.

3.36 multipoint processor: The Multipoint Processor (MP) is an H.323 entity on the network which provides for the centralized processing of audio, video and/or data streams in a multipoint conference. The MP provides for the mixing, switching or other processing of media streams under the control of the MC. The MP may process a single media stream or multiple media streams depending on the type of conference supported.

3.37 multi-unicast: A process of transferring PDUs where an endpoint sends more than one copy of a media stream, but to different endpoints. This may be necessary in networks which do not support multicast.

3.38 network address: The network layer address of an H.323 entity as defined by the (inter)network layer protocol in use (e.g., an IP address). This address is mapped onto the layer one address of the respective system by some means defined in the (inter)networking protocol.

3.39 packet based network (also network): Any shared, switched, or point-to-point medium which provides peer-to-peer communications between two or more endpoints using a packet based transport protocol.

3.40 point-to-point conference: A Point-to-Point conference is a conference between two terminals. It may be either directly between two H.323 terminals or between an H.323 terminal and an SCN terminal via a Gateway. A call between two terminals (see Call).

3.41 RAS channel: Unreliable channel used to convey the registration, admissions, bandwidth change, and status messages (following ITU-T Rec. H.225.0) between two H.323 entities.

3.42 reliable channel: A transport connection used for reliable transmission of an information stream from its source to one or more destinations.

3.43 reliable transmission: Transmission of messages from a sender to a receiver using connection-mode data transmission. The transmission service guarantees sequenced, error-free, flow-controlled transmission of messages to the receiver for the duration of the transport connection.

3.44 RTP session: For each participant, the session is defined by a particular pair of destination Transport Addresses (one Network Address plus a TSAP identifier pair for RTP and RTCP). The destination Transport Address pair may be common for all participants, as in the case of IP multicast, or may be different for each, as in the case of individual unicast network addresses. In a multimedia session, the media audio and video are carried in separate RTP sessions with their own RTCP packets. The multiple RTP sessions are distinguished by different Transport Addresses.

3.45 switched circuit network (SCN): A public or private switched telecommunications network such as the GSTN, N-ISDN, or B-ISDN.

NOTE – While B-ISDN is not strictly a switched circuit network, it exhibits some of the characteristics of an SCN through the use of virtual circuits.

3.46 terminal: An H.323 Terminal is an endpoint on the network which provides for real-time, two-way communications with another H.323 terminal, Gateway, or Multipoint Control Unit. This communication consists of control, indications, audio, moving colour video pictures, and/or data between the two terminals. A terminal may provide speech only, speech and data, speech and video, or speech, data and video.

3.47 transport address: The transport layer address of an addressable H.323 entity as defined by the (inter)network protocol suite in use. The Transport Address of an H.323 entity is composed of the Network Address plus the TSAP identifier of the addressable H.323 entity.

3.48 transport connection: An association established by a transport layer between two H.323 entities for the transfer of data. In the context of this Recommendation, a transport connection provides reliable transmission of information.

3.49 trunking gateway: a Gateway that connects two like networks (for example, two SS7 networks or two QSIG networks), in which tunnelling is used to create full transparency and a true tandem function.

3.50 TSAP identifier: The piece of information used to multiplex several transport connections of the same type on a single H.323 entity with all transport connections sharing the same Network Address, (e.g., the port number in a TCP/UDP/IP environment). TSAP identifiers may be (pre)assigned statically by some international authority or may be allocated dynamically during the setup of a call. Dynamically assigned TSAP identifiers are of transient nature, i.e., their values are only valid for the duration of a single call.

3.51 unicast: A process of transmitting messages from one source to one destination.

3.52 unreliable channel: A logical communication path used for unreliable transmission of an information stream from its source to one or more destinations.

3.53 unreliable transmission: Transmission of messages from a sender to one or more receivers by means of connectionless-mode data transmission. The transmission service is *best-effort* delivery of the PDU, meaning that messages transmitted by the sender may be lost, duplicated, or received out of order by (any of) the receiver(s).

3.54 well-known TSAP identifier: A TSAP identifier that has been allocated by an (international) authority that is in charge of the assignment of TSAP identifiers for a particular (inter)networking protocol and the related transport protocols (e.g., the IANA for TCP and UDP port numbers). This identifier is guaranteed to be unique in the context of the respective protocol.

3.55 zone: A Zone (see Figure 3) is the collection of all terminals (Tx), Gateways (GW) and Multipoint Control Units (MCUs) managed by a single Gatekeeper (GK). A Zone has one and only one Gatekeeper. A Zone may be independent of network topology and may be comprised of multiple network segments which are connected using routers (R) or other devices.

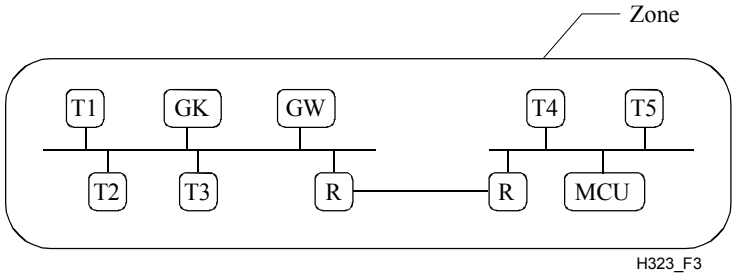


Figure 3/H.323 – Zone

4 Symbols and abbreviations

This Recommendation uses the following abbreviations:

4CIF 4 times CIF

16CIF 16 times CIF

ABNF Augmented Backus-Naur Form

ABR Available Bit Rate

ABT/DT ATM Block Transfer/Delayed Transmission

ABT/IT ATM Block Transfer/Immediate Transmission

ACF Admission Confirmation

AGW Access Gateway

APE Application Protocol Entity

ARJ Admission Reject

ARQ Admission Request

ATC ATM Transfer Capability

ATM Asynchronous Transfer Mode

BAS Bit rate Allocation Signal

BCF Bandwidth Change Confirmation

BCH Bose, Chaudhuri, and Hocquengham

B-HLI	Broadband High Layer Information
B-ISDN	Broadband Integrated Services Digital Network
BRJ	Bandwidth Change Reject
BRQ	Bandwidth Change Request
BTC	Broadband Transfer Capability
CAS	Channel Associated Signalling
CDV	Cell Delay Variation
CED	Called Terminal Identification Tone
CER	Cell Error Ratio
CID	Conference Identifier
CIF	Common Intermediate Format
CLR	Cell Loss Ratio
CMR	Cell Misinsertion Rate
CNG	Calling Tone
CTD	Cell Transfer Delay
DBR	Deterministic Bit Rate
DCF	Disengage Confirmation
DNS	Domain Name System
DRQ	Disengage Request
DSVD	Digital Simultaneous Voice and Data
DTMF	Dual-Tone MultiFrequency
FAS	Facility Associated Signalling
FIR	Full Intra Request
GCC	Generic Conference Control
GCF	Gatekeeper Confirmation
GID	Global Call Identifier
GIT	Generic Identifier Transport
GK	Gatekeeper
GQOS	Guaranteed Quality of Service
GRJ	Gatekeeper Reject
GRQ	Gatekeeper Request
GSTN	General Switched Telephone Network
GW	Gateway
HDLC	High Level Data Link Control
HTTP	Hypertext Transfer Protocol
IACK	Information Acknowledgment
IANA	Internet Assigned Numbers Authority

ID	Identifier
IE	Information Element
IMT	Inter-machine Trunk
INAK	Information Negative Acknowledgment
IP	Internet Protocol
IPX	Internetwork Protocol Exchange
IRQ	Information Request
IRR	Information Request Response
ISDN	Integrated Services Digital Network
ISUP	ISDN User Part
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
LAN	Local Area Network
LCF	Location Confirmation
LRJ	Location Reject
LRQ	Location Request
MC	Multipoint Controller
MCS	Multipoint Communications System
MCU	Multipoint Control Unit
MG	Media Gateway
MGC	Media Gateway Controller
MIME	Multipurpose Internet Mail Extensions
MP	Multipoint Processor
MTU	Maximum Transmission Unit
NACK	Negative Acknowledge
NFAS	Non-facility Associated Signalling
N-ISDN	Narrow-band Integrated Services Digital Network
NNI	Network-to-Network Interface
NSAP	Network Service Access Point
OLC	H.245 **openLogicalChannel** message
PBN	Packet Based Network
PDU	Packet Data Unit
PPP	Point-to-Point Protocol
PRI	Primary Rate Interface
QCIF	Quarter CIF
QOS	Quality of Service
QSIG	Signalling between the Q reference points defined in [45]
RAS	Registration, Admission and Status

RAST Receive and Send Terminal
RCF Registration Confirmation
RIP Request in Progress
RRJ Registration Reject
RRQ Registration Request
RTCP Real Time Control Protocol
RTP Real Time Protocol
SBE Single Byte Extension
SBR1 Statistical Bit Rate configuration 1
SBR2 Statistical Bit Rate configuration 2
SBR3 Statistical Bit Rate configuration 3
SCI Service Control Indication
SCM Selected Communications Mode
SCN Switched Circuit Network
SCR Service Control Response
SDL Specification and Description Language
SECBR Severely Errored Cell Block Ratio
SPX Sequential Protocol Exchange
SQCIF Sub QCIF
SS7 Signalling System No. 7
SSRC Synchronization Source Identifier
TCP Transport Control Protocol
TGW Trunking Gateway
TSAP Transport layer Service Access Point
UCF Unregister Confirmation
UDP User Datagram Protocol
UNI User-to-Network Interface
URJ Unregister Reject
URQ Unregister Request
VC Virtual Channel

5 Conventions

In this Recommendation, the following conventions are used:

"Shall" indicates a mandatory requirement.

"Should" indicates a suggested but optional course of action.

"May" indicates an optional course of action rather than a recommendation that something take place.

References to clauses, subclauses, annexes and appendices refer to those items within this Recommendation unless another specification is explicitly listed. For example, 1.4 refers to 1.4 of this Recommendation; 6.4/H.245 refers to 6.4 in ITU-T Rec. H.245.

Throughout this Recommendation, the term "network" is used to indicate any packet based network regardless of the underlying physical connection or the geographic scope of the network. This includes Local Area Networks, internetworks, and other packet based networks. The term "Switched Circuit Network" or "SCN" is used explicitly when referring to switched circuit networks such as GSTN and ISDN.

Where items exist on both the packet based network and the SCN, references to the SCN item will be explicit. For example, an MCU is an H.323 MCU on the packet based network, an SCN-MCU is an MCU on the SCN.

This Recommendation describes the use of three different message types: H.245, RAS and H.225.0 call signalling. To distinguish between the different message types, the following convention is followed. H.245 message and parameter names consist of multiple concatenated words highlighted in bold typeface (**maximumDelayJitter**). RAS message names are represented by three letter abbreviations (ARQ). H.225.0 call signalling message names consist of one or two words with the first letters capitalized (Call Proceeding).

6 System description

This Recommendation describes the elements of the H.323 components. These are Terminals, Gateways, Gatekeepers, MCs and MCUs. These components communicate through the transmission of Information Streams. The characteristics of these components are described in this clause.

6.1 Information streams

Visual telephone components communicate through the transmission of Information Streams. These Information Streams are classified into video, audio, data, communications control and call control as follows.

Audio signals contain digitized and coded speech. In order to reduce the average bit rate of audio signals, voice activation may be provided. The audio signal is accompanied by an audio control signal.

Video signals contain digitized and coded motion video. Video is transmitted at a rate no greater than that selected as a result of the capability exchange. The video signal is accompanied by a video control signal.

Data signals include still pictures, facsimile, documents, computer files and other data streams.

Communications control signals pass control data between remote-like functional elements and are used for capability exchange, opening and closing logical channels, mode control and other functions that are part of communications control.

Call control signals are used for call establishment, disconnect and other call control functions.

The information streams described above are formatted and sent to the network interface as described in ITU-T Rec. H.225.0.

6.2 Terminal characteristics

An example of an H.323 terminal is shown in Figure 4. The diagram shows the user equipment interfaces, video codec, audio codec, telematic equipment, H.225.0 layer, system control functions and the interface to the packet based network. All H.323 terminals shall have a System Control

Unit, H.225.0 layer, Network Interface and an Audio Codec Unit. The Video Codec Unit and User Data Applications are optional.

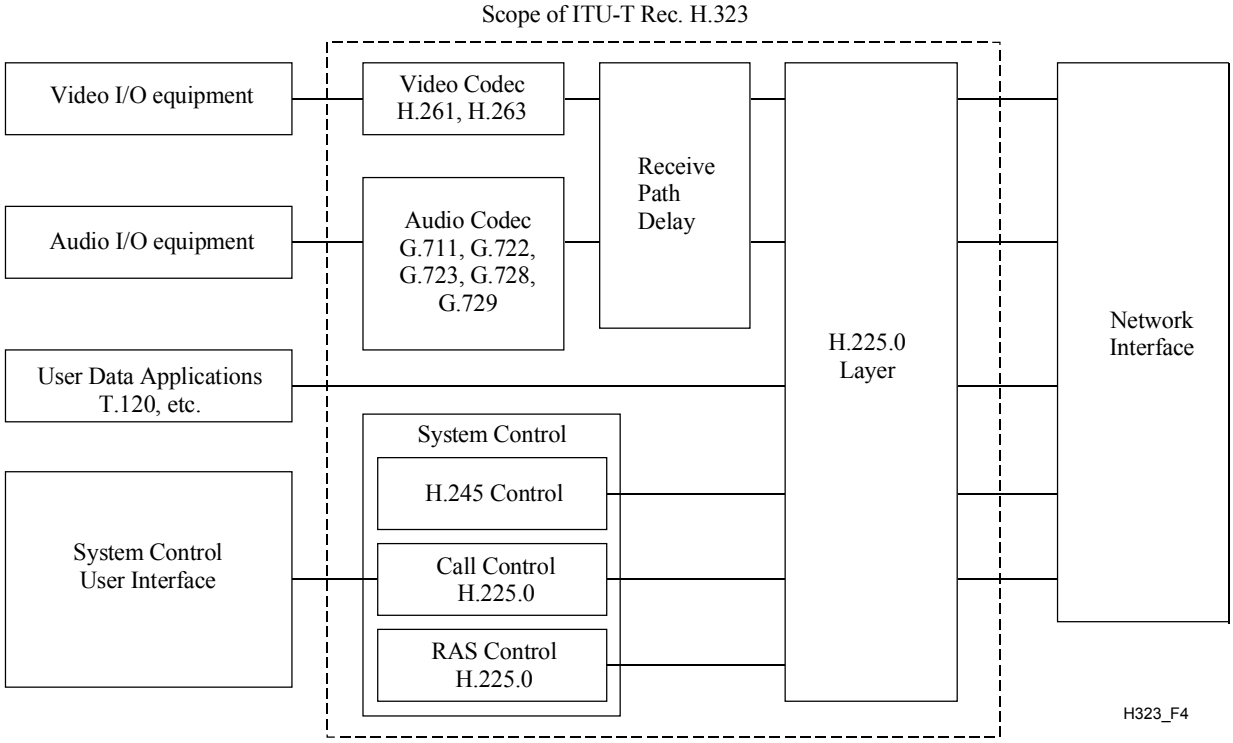


Figure 4/H.323 – H.323 terminal equipment

6.2.1 Terminal elements outside the scope of this Recommendation

The following elements are not within the scope of this Recommendation and are therefore not defined within this Recommendation:

- Attached audio devices providing voice activation sensing, microphone and loudspeaker, telephone instrument or equivalent, multiple microphones mixers, and acoustic echo cancellation.
- Attached video equipment providing cameras and monitors, and their control and selection, video processing to improve compression or provide split screen functions.
- Data applications and associated user interfaces which use T.120 or other data services over the data channel.
- Attached Network Interface, which provides the interface to the packet based network, supporting appropriate signalling and voltage levels, in accordance with national and international standards.
- Human user system control, user interface and operation.

6.2.2 Terminal elements within the scope of this Recommendation

The following elements are within the scope of this Recommendation and are therefore subject to standardization and are defined within this Recommendation:

- The Video Codec (H.261, etc.) encodes the video from the video source (i.e., camera) for transmission and decodes the received video code which is output to a video display.
- The Audio Codec (G.711, etc.) encodes the audio signal from the microphone for transmission and decodes the received audio code which is output to the loudspeaker.

- The Data Channel supports telematic applications such as electronic whiteboards, still image transfer, file exchange, database access, audiographics conferencing, etc. The standardized data application for real-time audiographics conferencing is ITU-T Rec. T.120. Other applications and protocols may also be used via H.245 negotiation as specified in 6.2.7.
- The System Control Unit (H.245, H.225.0) provides signalling for proper operation of the H.323 terminal. It provides for call control, capability exchange, signalling of commands and indications, and messages to open and fully describe the content of logical channels.
- H.225.0 Layer (H.225.0) formats the transmitted video, audio, data and control streams into messages for output to the network interface and retrieves the received video, audio, data and control streams from messages which have been input from the network interface. In addition, it performs logical framing, sequence numbering, error detection and error correction as appropriate to each media type.

6.2.3 Packet based network interface

The packet based network interface is implementation-specific and is outside the scope of this Recommendation. However, the network interface shall provide the services described in ITU-T Rec. H.225.0. This includes the following: Reliable (e.g., TCP, SPX) end-to-end service is mandatory for the H.245 Control Channel, the Data Channels, and the Call Signalling Channel. Unreliable (e.g., UDP, IPX) end-to-end service is mandatory for the Audio Channels, the Video Channels and the RAS Channel. These services may be duplex or simplex, unicast or multicast depending on the application, the capabilities of the terminals, and the configuration of the network.

6.2.4 Video codec

The video codec is optional. If video capability is provided, it shall be provided according to the requirements of this Recommendation. All H.323 terminals providing video communications shall be capable of encoding and decoding video according to H.261 QCIF. Optionally, a terminal may also be capable of encoding and decoding video according to the other modes of H.261 or H.263. If a terminal supports H.263 with CIF or higher resolution, it shall also support H.261 CIF. All terminals which support H.263 shall support H.263 QCIF. The H.261 and H.263 codecs, on the network, shall be used without BCH error correction and without error correction framing.

Other video codecs and other picture formats, may also be used via H.245 negotiation. More than one video channel may be transmitted and/or received, as negotiated via the H.245 Control Channel. The H.323 terminal may optionally send more than one video channel at the same time, for example, to convey the speaker and a second video source. The H.323 terminal may optionally receive more than one video channel at the same time, for example, to display multiple participants in a distributed multipoint conference.

The video bit rate, picture format and algorithm options that can be accepted by the decoder are defined during the capability exchange using H.245. The encoder is free to transmit anything that is within the decoder capability set. The decoder should have the possibility to generate requests via H.245 for a certain mode, but the encoder is allowed to simply ignore these requests if they are not mandatory modes. Decoders which indicate capability for a particular algorithm option shall also be capable of accepting video bit streams which do not make use of that option.

H.323 terminals shall be capable of operating in asymmetric video bit rates, frame rates, and, if more than one picture resolution is supported, picture resolutions. For example, this will allow a CIF capable terminal to transmit QCIF while receiving CIF pictures.

When each video logical channel is opened, the selected operating mode to be used on that channel is signalled to the receiver in the H.245 **openLogicalChannel** message. The header within the video logical channel indicates which mode is actually used for each picture, within the stated decoder capability.

The video stream is formatted as described in ITU-T Rec. H.225.0.

6.2.4.1 Terminal-based continuous presence

H.323 terminals may receive more than one video channel, particularly for multipoint conferencing. In these cases, the H.323 terminal may need to perform a video mixing or switching function in order to present the video signal to the user. This function may include presenting the video from more than one terminal to the user. The H.323 terminal shall use H.245 simultaneous capabilities to indicate how many simultaneous video streams it is capable of decoding. The simultaneous capability of one terminal should not limit the number of video streams which are multicast in a conference (this choice is made by the MC).

6.2.5 Audio codec

All H.323 terminals shall have an audio codec. All H.323 terminals shall be capable of encoding and decoding speech according to ITU-T Rec. G.711. All terminals shall be capable of transmitting and receiving A-law and μ-law. A terminal may optionally be capable of encoding and decoding speech using other audio codecs which can be signalled via H.245 negotiation. The audio algorithm used by the encoder shall be derived during the capability exchange using H.245. The H.323 terminal should be capable of asymmetric operation for all audio capabilities it has declared within the same capability set, e.g., it should be able to send G.711 and receive G.728 if it is capable of both.

If G.723.1 audio is provided, the audio codec shall be capable of encoding and decoding according to both the 5.3 kbit/s mode and the 6.3 kbit/s mode.

The audio stream is formatted as described in ITU-T Rec. H.225.0.

The H.323 terminal may optionally send more than one audio channel at the same time, for example, to allow two languages to be conveyed.

Audio packets should be delivered to the transport layer periodically at an interval determined by the audio codec Recommendation in use (audio frame interval). The delivery of each audio packet shall occur no later than 5 ms after a whole multiple of the audio frame interval, measured from delivery of the first audio frame (audio delay jitter). Audio coders capable of further limiting their audio delay jitter may so signal using the H.245 **maximumDelayJitter** parameter of the **h2250Capability** structure contained within a terminal capability set message, so that receivers may optionally reduce their jitter delay buffers. This is not the same as the RTCP interarrival jitter field.

NOTE – The testing point for the maximum delay jitter is at the input to network transport layer. Network stack, network, driver, and interface card jitter are not included.

6.2.5.1 Audio mixing

H.323 terminals may receive more than one audio channel, particularly for multipoint conferencing. In these cases, the H.323 terminal may need to perform an audio mixing function in order to present a composite audio signal to the user. The H.323 terminal shall use H.245 simultaneous capabilities to indicate how many simultaneous audio streams it is capable of decoding. The simultaneous capability of one terminal should not limit the number of audio streams which are multicast in a conference.

6.2.5.2 Maximum audio-video transmit skew

To allow H.323 terminals to appropriately set their receive buffer(s) size, H.323 terminals shall transmit the **h2250MaximumSkewIndication** message to indicate the maximum skew between the audio and video signals as delivered to the network transport. **h2250MaximumSkewIndication** shall be sent for each pair of associated audio and video logical channels. This is not required for

audio only or hybrid conferences. Lip synchronization, if desired, shall be achieved via use of time-stamps.

6.2.5.3 Low bit rate operation

G.711 audio cannot be used in an H.323 conference being carried over low bit rate (< 56 kbit/s) links or segments. An endpoint used for multimedia communications over such low bit rate links or segments should have an audio codec capable of encoding and decoding speech according to ITU-T Rec. G.723.1. An endpoint used for audio-only communications over such low bit rate links or segments should have an audio codec capable of encoding and decoding speech according to ITU-T Rec. G.729. An endpoint may support several audio codecs in order to provide the widest possible interoperability with those endpoints which only support one low bitrate audio codec. The endpoint shall indicate in the H.245 Capability Exchange procedures at the beginning of each call the capability to receive audio according to the available audio Recommendations which can be supported within the known bit rate limitations of the connection. An endpoint which does not have this low bit rate audio capability may not be able to operate when the end-to-end connection contains one or more low bit rate segments.

The endpoint shall also comply with the requirement of 6.2.5 to be capable of encoding and decoding speech according to ITU-T Rec. G.711. However, the endpoint need not indicate this capability if it is sure that it is communicating through a low bit rate segment. If an endpoint is unaware of the presence, in the end-to-end connection, of any links or segments with insufficient capacity to support G.711 audio (along with other intended media streams, if any), then the endpoint shall declare the capability to receive audio according to ITU-T Rec. G.711.

6.2.6 Receive path delay

Receive path delay includes delay added to a media stream in order to maintain synchronization and to account for network packet arrival jitter. Media streams may optionally be delayed in the receiver processing path to maintain synchronization with other media streams. Further, a media stream may optionally be delayed to allow for network delays which cause packet arrival jitter. An H.323 terminal shall not add delay for this purpose in its transmitting media path.

Intermediate processing points such as MCUs or Gateways may alter the video and audio time tag information and shall transmit appropriately modified audio and video time tags and sequence numbers, reflecting their transmitted signals. Receiving endpoints may add appropriate delay in the audio path to achieve lip synchronization.

6.2.7 Data channel

One or more data channels are optional. The data channel may be unidirectional or bidirectional depending on the requirements of the data application.

ITU-T Rec. T.120 is the default basis of data interoperability between an H.323 terminal and other H.323, H.324, H.320 or H.310 terminals. Where any optional data application is implemented using one or more of the ITU-T Recommendations, which can be negotiated via H.245, the equivalent T.120 application, if any, shall be one of those provided.

Note that non-standard data applications (**dataApplicationCapability.application = non-standard** application) and Transparent User Data (**dataApplicationCapability.application = userData** application, **dataProtocolCapability = transparent**) may be used whether the equivalent T.120 application is provided or not.

T.120 capability shall be signalled using **dataApplicationCapability.application = t120** application, **dataProtocolCapability = separateLANStack**.

Within the **MediaDistributionCapability**, the **distributedData** structure shall be used if multicast T.120 is available and/or the **centralizedData** structure if unicast T.120 is available. Any node that supports the T.120 data capability shall support the standard T.123 unicast stack.

In the **openLogicalChannel** message, the **distribution** choice of the **NetworkAccessParameters** structure is set to **unicast** if T.123 is to be used or **multicast** if Annex A/T.125 is to be used. The **networkAddress** choice is set to **localAreaAddress**, which should always be **unicastAddress**. Within the **iPAddress** sequence, the **network** field is set to the binary IP address and the **tsapIdentifier** is set to the dynamic port on which the T.120 stack will be calling or listening.

The Data channel is formatted as described in ITU-T Rec. H.225.0.

6.2.7.1 T.120 data channels

The T.120 connection is established during an H.323 call as an inherent part of the call. Procedures for establishing the T.120 connection prior to the H.323 connection are for further study.

The normal call setup procedures of 8.1 are followed. After the capability exchange takes place, a bidirectional logical channel shall be opened for the T.120 connection according to the normal H.245 procedures indicating that a new connection shall be created as described below.

The opening of a bidirectional logical channel for T.120 may be initiated by either entity sending an **openLogicalChannel** message and then following the bidirectional logical channel procedures of ITU-T Rec. H.245.

To actually open the logical channel, the initiating entity shall send an **openLogicalChannel** message indicating that a T.120 data channel is to be opened in the **forwardLogicalChannelParameters** as well as in the **reverseLogicalChannelParameters**. The initiator shall include a Transport Address in the **openLogicalChannel** message. The peer endpoint may choose to ignore the Transport Address. An endpoint may use a dynamic port number for the T.120 Transport Address instead of using port 1503 as specified in ITU-T Rec. T.123. If the peer (the responder) accepts this logical channel, it shall confirm the opening of the logical channel using **openLogicalChannelAck**. In the **openLogicalChannelAck**, the responder shall include a Transport Address even if it expects the initiator to originate the T.120 call. In all cases, the Transport Address for the T.120 connection shall be carried in the **separateStack** parameter and shall remain valid for the duration of the logical channel.

In the **openLogicalChannel** message, the **t120SetupProcedure** choice of the **NetworkAccessParameters** structure can optionally be set to indicate to the responder how the initiator would like to establish the T.120 call. The responder is free to override this preference. **originateCall** indicates that the initiator would like the responder to place the call. **waitForCall** indicates that the initiator would like the responder to receive the call. **issueQuery** is not used when indicating a preference.

In the **openLogicalChannelAck** message, the **t120SetupProcedure** choice of the **NetworkAccessParameters** structure should be set to indicate to the initiator how the T.120 call will be established. If neither endpoint has a preference, the T.120 call should be established in the same direction as the H.323 call. **originateCall** tells the initiator to place the call. **waitForCall** tells the initiator that it will receive the call. Whoever originates the call will issue either a join request or an invite request, depending on which endpoint won master/slave determination (the master is always hierarchically higher in the T.120 conference). **issueQuery** can be used by a Gateway to tell the initiator that it must originate the call and issue a query request to the remote endpoint. It must then set up the T.120 conference in accordance with the contents of the query response (as described in ITU-T Rec. T.124).

When possible, the T.120 call should be established in the same direction as the H.323 call. The OLC initiator should not indicate a preference unless there is a need to modify this default behaviour. When the initiator indicates a preference, the responder should not override it unless

necessary. When no preference is indicated, the responder should specify the default unless there is a need to do otherwise.

In both the **openLogicalChannel** and the **openLogicalChannelAck** messages, the **associateConference** parameter shall be set to FALSE.

ITU-T Rec. T.120 shall follow the procedures of ITU-T Rec. T.123 for the protocol stack indicated in the **dataProtocolCapability** except that the Transport Addresses as described above shall be employed for connection setup.

If an endpoint is the Active MC or master in a conference, which includes T.120, it should also be in control of the T.120 top provider node.

If an endpoint intends to create a conference, which includes audio and/or video plus T.120 data, then the H.245 Control Channel shall be established before the T.120 connection is made. This applies to conference create, join, and invite and the actions of an MC. The H.323 call setup procedures shall be used to establish the Active MC (if any), before a T.120 connection is made.

In order to establish a T.120 connection using a GCC-Join request, endpoints are required to know the T.120 conference name. If an alias exists which represents an H.323 conference name (**conferenceAlias**), then the same text which is used for the conference alias should be used as the text portion of the T.120 conference name. Likewise, the H.323 CID should be used as the numeric T.120 conference name as follows. Each byte of the H.323 CID is converted into a series of three ASCII characters, which represent the decimal value of the byte being converted. Note that this requires the value of some CID bytes to be converted such that "0" characters are used for padding. The result will be a string of 48 ASCII characters.

A T.120 MP may be queried for a list of existing conferences. The H.323 CID may be available by converting from the T.120 Numeric Conference name back into the 16-byte octet string. Likewise, the Text Conference name may be used as the H.323 conference alias. Note that a T.124 Conference Query may happen out-of-band from H.323 and prior to an endpoint setting up an H.323 call.

The termination of the associated T.120 conference does not imply the termination of the H.323 call. In other words, closing the T.120 channel shall only affect the Data stream of an H.323 call and shall not affect any other part of the H.323 call. By contrast, when an H.323 call or conference is terminated, then the associated T.120 conference shall also be terminated.

NOTE – The T.120 operation after completion of the connection setup is beyond the scope of this Recommendation.

6.2.7.2 Remote device control

H.323 endpoints may support remote device control through the H.282 protocol. The H.282 protocol shall be supported in an H.245 logical channel according to ITU-T Rec. H.283. ITU-T Rec. H.283 describes logical channel transport for the H.282 protocol in an H.323 conference.

ITU-T Rec. H.282 may also be used by T.120 systems and carried in a T.120 APE. Optionally H.323 systems may also support remote device control using ITU-T Rec. H.282 over T.120. However, this is an option and an H.323 system that supports H.282 shall support it with ITU-T Rec. H.283.

If both H.282 with H.283 and H.282 with T.120 are supported, then both may be used. Coordination of the two lower layer protocols under H.282 is a local matter. However, H.283 shall always be active to account for possible late joining nodes that support H.282 over H.283 but not H.282 over T.120.

6.2.8 H.245 control function

The H.245 Control Function uses the H.245 Control Channel to carry end-to-end control messages governing operation of the H.323 entity, including capabilities exchange, opening and closing of logical channels, mode preference requests, flow control messages, and general commands and indications.

H.245 signalling is established between two endpoints, an endpoint and an MC, or an endpoint and a Gatekeeper. The endpoint shall establish exactly one H.245 Control Channel for each call that the endpoint is participating in. This channel shall use the messages and procedures of ITU-T Rec. H.245. Note that a terminal, MCU, Gateway, or Gatekeeper may support many calls and thus many H.245 Control Channels. The H.245 Control Channel shall be carried on logical channel 0. Logical channel 0 shall be considered to be permanently open from the establishment of the H.245 Control Channel until the termination of this channel. The normal procedures for opening and closing logical channels shall not apply to the H.245 Control Channel.

ITU-T Rec. H.245 specifies a number of independent protocol entities which support endpoint-to-endpoint signalling. A protocol entity is specified by its syntax (messages), semantics, and a set of procedures which specify the exchange of messages and the interaction with the user. H.323 endpoints shall support the syntax, semantics, and procedures of the following protocol entities:

- Master/slave determination.
- Capability Exchange.
- Logical Channel Signalling.
- Bidirectional Logical Channel Signalling.
- Close Logical Channel Signalling.
- Mode Request.
- Round Trip Delay Determination.
- Maintenance Loop Signalling.

General commands and indications shall be chosen from the message set contained in ITU-T Rec. H.245. In addition, other command and indication signals may be sent which have been specifically defined to be transferred in-band within video, audio or data streams (see the appropriate Recommendation to determine if such signals have been defined).

H.245 messages fall into four categories: Request, Response, Command and Indication. Request and Response messages are used by the protocol entities. Request messages require a specific action by the receiver, including an immediate response. Response messages respond to a corresponding request. Command messages require a specific action, but do not require a response. Indication messages are informative only and do not require any action or response. H.323 terminals shall respond to all H.245 commands and requests as specified in Annex A and shall transmit indications reflecting the state of the terminal.

H.323 terminals shall be capable of parsing all H.245 **multimediaSystemControlMessage** messages and shall send and receive all messages needed to implement required functions and those optional functions which are supported by the terminal. Annex A contains a table showing which H.245 messages are mandatory, optional, or forbidden for H.323 terminals. H.323 terminals shall send the **functionNotSupported** message in response to any unrecognized request, response, or command messages that it receives.

An H.245 indication, **userInputIndication**, is available for transport of user input alphanumeric characters from a keypad or keyboard, equivalent to the DTMF signals used in analogue telephony or SBE number messages in ITU-T Rec. H.230. This may be used to manually operate remote equipment such as voice mail or video mail systems, menu-driven information services, etc.

H.323 terminals shall support the transmission of user input characters 0-9, "*", and "#". Transmission of other characters is optional.

Three H.245 request messages conflict with RTCP control packets. The H.245 **videoFastUpdatePicture**, **videoFastUpdateGOB** and **videoFastUpdateMB** requests should be used instead of the RTCP control packets Full Intra Request (FIR) and Negative Acknowledgement (NACK). The ability to accept FIR and NACK is signalled during the H.245 capability exchange.

6.2.8.1 Capabilities exchange

Capabilities exchange shall follow the procedures of ITU-T Rec. H.245, which provides for separate receive and transmit capabilities, as well as a method by which the terminal may describe its ability to operate in various combinations of modes simultaneously.

Receive capabilities describe the terminal's ability to receive and process incoming information streams. Transmitters shall limit the content of their transmitted information to that which the receiver has indicated it is capable of receiving. The absence of a receive capability indicates that the terminal cannot receive (is a transmitter only).

Transmit capabilities describe the terminal's ability to transmit information streams. Transmit capabilities serve to offer receivers a choice of possible modes of operation, so that the receiver may request the mode which it prefers to receive. The absence of a transmit capability indicates that the terminal is not offering a choice of preferred modes to the receiver (but it may still transmit anything within the capability of the receiver).

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

The transmitting terminal assigns each individual mode the terminal is capable of operating in a number in a **capabilityTable**. For example, G.723.1 audio, G.728 audio, and CIF H.263 video would each be assigned separate numbers.

These capability numbers are grouped into **alternativeCapabilitySet** structures. Each **alternativeCapabilitySet** indicates that the terminal is capable of operating in exactly one mode listed in the set. For example, an **alternativeCapabilitySet** listing {G.711, G.723.1, G.728} means that the terminal can operate in any one of those audio modes, but not more than one.

These **alternativeCapabilitySet** structures are grouped into **simultaneousCapabilities** structures. Each **simultaneousCapabilities** structure indicates a set of modes the terminal is capable of using simultaneously. For example, a **simultaneousCapabilities** structure containing the two **alternativeCapabilitySet** structures {H.261, H.263} and {G.711, G.723.1, G.728} means that the terminal can operate either of the video codecs simultaneously with any one of the audio codecs. The **simultaneousCapabilities** set { {H.261}, {H.261, H.263}, {G.711, G.723.1, G.728} } means the terminal can operate two video channels and one audio channel simultaneously: one video channel per H.261, another video channel per either H.261 or H.263, and one audio channel per either G.711, G.723.1, or G.728.

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

NOTE 1 – The master may send a **requestMode** to the slave with the proper codec before sending the **openLogicalChannelReject** to explicitly request a specific codec.

NOTE 2 – The actual capabilities stored in the **capabilityTable** are often more complex than presented here. For example, each H.263 capability indicates details including the ability to support various picture formats at given minimum picture intervals and the ability to use optional coding modes. For a complete description, see ITU-T Rec. H.245.

The terminal's total capabilities are described by a set of **capabilityDescriptor** structures, each of which is a single **simultaneousCapabilities** structure and a **capabilityDescriptorNumber**. By sending more than one **capabilityDescriptor**, the terminal may signal dependencies between operating modes by describing different sets of modes which it can simultaneously use. For example, a terminal issuing two **capabilityDescriptor** structures, one { {H.261, H.263}, {G.711, G.723.1, G.728} } as in the previous example, and the other { {H.262}, {G.711} }, means the terminal can also operate the H.262 video codec, but only with the low-complexity G.711 audio codec.

Terminals may dynamically add capabilities during a communication session by issuing additional **capabilityDescriptor** structures or remove capabilities by sending revised **capabilityDescriptor** structures. All H.323 terminals shall transmit at least one **capabilityDescriptor** structure.

Non-standard capabilities and control messages may be issued using the **nonStandardParameter** structure defined in ITU-T Rec. H.245. Note that while the meaning of non-standard messages is defined by individual organizations, equipment built by any manufacturer may signal any non-standard message, if the meaning is known.

Terminals may reissue capability sets at any time, according to the procedures of ITU-T Rec. H.245.

6.2.8.2 Logical channel signalling

Each logical channel carries information from a transmitter to one or more receivers and is identified by a logical channel number which is unique for each direction of transmission.

Logical channels are opened and closed using the **openLogicalChannel** and **closeLogicalChannel** messages and procedures of ITU-T Rec. H.245. When a logical channel is opened, the **openLogicalChannel** message fully describes the content of the logical channel, including media type, algorithm in use, any options, and all other information needed for the receiver to interpret the content of the logical channel. Logical channels may be closed when no longer needed. Open logical channels may be inactive, if the information source has nothing to send.

Most logical channels in this Recommendation are unidirectional, so asymmetrical operation, in which the number and type of information streams is different in each direction of transmission, is allowed. However, if a receiver is capable only of certain symmetrical modes of operation, it may send a receive capability set that reflects its limitations, except where noted elsewhere in this Recommendation. Terminals may also be capable of using a particular mode in only one direction of transmission. Certain media types, including data protocols such as T.120, inherently require a bidirectional channel for their operation. In such cases a single bidirectional logical channel may be opened using the bidirectional channel opening procedures of ITU-T Rec. H.245.

Logical channels shall be opened using the following procedure:

The initiating terminal shall send an **openLogicalChannel** message as described in ITU-T Rec. H.245. If the logical channel is to carry a media type using RTP (audio or video), the **openLogicalChannel** message shall include the **mediaControlChannel** parameter containing the Transport Address for the reverse RTCP channel.

The responding terminal shall respond with an **openLogicalChannelAck** message as described in ITU-T Rec. H.245. If the logical channel is to carry a media type using RTP, the **openLogicalChannelAck** message shall include both the **mediaChannel** parameter containing the

RTP Transport Address for the media channel and the **mediaControlChannel** parameter containing the Transport Address for the forward RTCP channel.

Media types (such as T.120 data) which do not use RTP/RTCP shall omit the **mediaControlChannel** parameters.

If a corresponding reverse channel is opened for a given existing RTP session (identified by the RTP **sessionID**), the **mediaControlChannel** Transport Addresses exchanged by the **openLogicalChannel** process shall be identical to those used for the forward channel. **sessionID** values of 1, 2 and 3 are pre-assigned for primary audio, video and data sessions, respectively. Even the slave endpoint can open logical channels for these primary sessions without negotiating the **sessionID** value with the master endpoint. The master endpoint can open any additional session with a particular **sessionID** value greater than 3. The slave endpoint can open a corresponding session with the given **sessionID**. Otherwise, the slave endpoint can open additional sessions with **sessionID**=0 in the **openLogicalChannel** message, but it shall acquire the actual **sessionID** value from the master endpoint's **openLogicalChannelAck** message. Should a collision occur where both ends attempt to establish conflicting RTP sessions at the same time, the master endpoint shall reject the conflicting attempt as described in ITU-T Rec. H.245. The rejected **openLogicalChannel** attempt may then be retried at a later time.

Unless specified otherwise for a particular data type, reliable data channels are bidirectional channels and, as such, shall contain both the **forwardLogicalChannelParameters** and **reverseLogicalChannelParameters** elements without the **mediaChannel** elements. The endpoint accepting the channel shall return the **mediaChannel** element in the **reverseLogicalChannelParameters** element and be prepared to accept the reliable connection from the requesting endpoint prior to returning the **OpenLogicalChannelAck** message.

An endpoint that accepts a bidirectional reliable channel shall be prepared to accept a reliable connection from the requesting endpoint prior to returning the **OpenLogicalChannelAck** message.

6.2.8.3 Mode preferences

Receivers may request transmitters to send a particular mode using the H.245 **requestMode** message, which describes the desired mode. Transmitters should comply if possible.

An endpoint receiving the **multipointModeCommand** from the MC shall then comply with all **requestMode** commands, if they are within its capability set. Note that in a decentralized conference, as in a centralized conference, all terminal **requestMode** commands are directed to the MC. The MC may grant the request or not; the basis for this decision is left to the manufacturer.

6.2.8.4 Master-slave determination

The H.245 Master-slave determination procedures are used to resolve conflicts between two endpoints which can both be the MC for a conference or between two endpoints which are attempting to open a bidirectional channel. In this procedure, two endpoints exchange random numbers in the H.245 **masterSlaveDetermination** message, to determine the master and slave endpoints. H.323 endpoints shall be capable of operating in both master and slave modes. The endpoints shall set **terminalType** to the value specified in Table 1 below and set **statusDeterminationNumber** to a random number in the range 0 to $2^{24} - 1$. Only one random number shall be chosen by the endpoint for each call, except in the case of identical random numbers, as described in ITU-T Rec. H.245.

Table 1/H.323 – H.323 terminal types for H.245 master-slave determination

TerminalType value table	H.323 entity			
Feature set	**Terminal**	**Gateway**	**Gatekeeper**	**MCU**
Entity with No MC	50	60	NA	NA
Entity contains an MC but no MP	70	80	120	160
Entity contains MC with data MP	NA	90	130	170
Entity contains MC with data and audio MP	NA	100	140	180
Entity contains MC with data, audio and video MP	NA	110	150	190

The Active MC in a conference shall use a value of 240.

If a single H.323 entity can take part in multiple calls, then the value used for **terminalType** in the master-slave determination process shall be based on the features that the H.323 entity has assigned or will assign to the call in which it is being signalled.

An MC that is already acting as an MC shall always remain the Active MC. Therefore, once an MC has been selected as the Active MC in a conference, it shall use the Active MC value for all subsequent connections to the conference.

If no MC is active and the entities are of the same type, then the H.323 entity with the highest feature set (as shown in Table 1) shall win the master-slave determination. If no MC is active and the entities are of different types, then an MC that is located in an MCU shall have priority over an MC that is located in a Gatekeeper, which shall have priority over an MC that is located in a Gateway, which in turn shall have priority over an MC located in a terminal.

If an H.323 entity can be associated with two or more of the classifications shown in Table 1, then it should use the highest value for which it qualifies.

6.2.8.5 Timer and counter values

All timers defined in ITU-T Rec. H.245 should have periods of at least as long as the maximum data delivery time allowed by the data link layer carrying the H.245 Control Channel, including any retransmissions.

The H.245 retry counter N100 should be at least 3.

Procedures relating to H.245 protocol error handling are covered in 8.6.

6.2.8.6 Multiplexed stream transmission over a single logical channel

Multiple media streams may be multiplexed over a single logical channel. A multiplexed stream is a stream that contains multiple media streams using the multiplexing protocols H.222.0 [46] or H.223 [47] and transmitted as a series of RTP packets. By using these multiplexing protocols, an H.323 endpoint may take advantage of certain benefits, such as more efficient bandwidth usage, precise media synchronization, or low delay in multimedia transmission.

There are two ways to control the configuration of a multiplexed stream. One way is to transmit the H.245 messages inside the RTP packets of multiplexed streams. In this case, H.323 endpoints first open bidirectional logical channel for the multiplexed stream transmission using H.245 logical channel signalling procedures as normal RTP media streams. Then the control for multiplexed stream is done using the H.245 messages inside RTP packets of the target multiplexed stream. The control of the multiplexed stream includes capability exchange about the media codecs available for this multiplexed stream, multiplex table exchange, and open/close logical channels. The logical channel numbers over a multiplexed stream are independent from that of the other multiplexed streams or that of H.245 logical channels.

The other way to control the configuration of a multiplexed stream is to control the logical channels on the multiplexed stream in the same way as non-multiplexed logical channels, i.e., the H.245 messages for the multiplexed stream are transmitted in the same manner as other H.245 messages. In this case, an H.323 endpoint opens a unidirectional or bidirectional logical channel for the multiplexed stream transmission using H.245 logical channel signalling procedures as normal RTP media streams. Then the logical channels over the multiplexed stream are opened using logical channel signalling with parameters of the multiplex protocol configuration and logical channel number of the multiplex stream over which the new logical channel is being opened.

6.2.8.6.1 Capability exchange related to multiplexed stream

H.323 terminals supporting multiplexed streams indicate this capability by including a **MultiplexedStreamCapability** as a part of terminal capability. The parameter **controlOnMuxStream** in **MultiplexedStreamCapability** indicates whether the terminal supports the control of the multiplexed stream using H.245 messages or on the RTP packets of the multiplexed stream itself. If **controlOnMuxStream** is TRUE, the capability of codecs on the multiplexed stream may be set to **capabilityOnMuxStream**. If **capabilityOnMuxStream** does not exist, the terminal shall perform the capability exchange procedure by sending the H.245 messages in the RTP packets over the multiplexed stream once the logical channel for the multiplexed stream is opened. If **controlOnMuxStream** is FALSE, the capability of codecs on the multiplexed stream shall be set to **capabilityOnMuxStream**.

6.2.8.6.2 Logical channel signalling to transport multiplexed stream

The logical channel for the multiplexed stream is opened by sending an **openLogicalChannel** message with the **dataType** of a **MultiplexedStreamCapability** type and **multiplexParameters** of **h2250LogicalChannelParameters**. If the **controlOnMuxStream** in **MultiplexedStreamCapability** is TRUE, the logical channel shall be opened as bidirectional logical channel, i.e., **reverseLogicalChannelParameters** shall be set. Otherwise, the logical channel may be opened as unidirectional logical channel. Note that if the logical channel is opened as unidirectional, some of the multiplex protocol function may not be used, e.g., AL3 of H.223 cannot be used over unidirectional logical channels.

Terminal shall not open more than one logical channel with **multiplexFormat** of **h223Capability** and **controlOnMuxStream** of FALSE.

6.2.8.6.3 Logical channel signalling to transport media stream over multiplexed stream

The logical channel over multiplexed stream is opened by sending an **openLogicalChannel** message with the appropriate **dataType** for the media being delivered and **multiplexParameters** of the appropriate multiplex protocol in use (i.e., **h223logicalChannelParameters**). In case of H.223, multiplex table signalling procedure shall also be performed before or after this logical channel signalling as described in 6.4.2/H.324.

If **controlOnMuxStream** is TRUE, these H.245 messages are delivered within the RTP packets for the multiplexed stream over which the new logical channel is opened. In case of H.223, H.245 **MultimediaSystemControlMessage** messages are protected with the Simple Retransmission Protocol (SRP) and delivered over logical channel number 0 of the multiplexed stream, as described in 6.5.4/H.324.

If **controlOnMuxStream** is FALSE, these H.245 messages are delivered over H.245 Control Channel as usual. In case of H.222.0, **resourceID** of **h2220LogicalChannelParameters** are set to the logical channel number for the multiplexed stream over which this new logical channel is being opened. Note that in case of H.223, no such signalling is needed due to the fact that no more than one logical channel exists.

Logical channels over the multiplexed stream are closed by sending **closeLogicalChannel** messages, which are transmitted in the same way as the **openLogicalChannel** messages for the channel.

6.2.8.6.4 Logical channel signalling to close multiplexed stream

The logical channel for the multiplexed stream which is opened with **controlOnMuxStream** set to TRUE may be closed at any time by a **closeLogicalChannel** message. The logical channel for the multiplexed stream which is opened with **controlOnMuxStream** set to FALSE shall be closed only after all logical channels on the multiplexed stream is closed.

6.2.9 RAS signalling function

The RAS signalling function uses H.225.0 messages to perform registration, admissions, bandwidth changes, status, and disengage procedures between endpoints and Gatekeepers. The RAS Signalling Channel is independent from the Call Signalling Channel and the H.245 Control Channel. H.245 open logical channel procedures are not used to establish the RAS Signalling Channel. In network environments that do not have a Gatekeeper, the RAS Signalling Channel is not used. In network environments which contain a Gatekeeper (a Zone), the RAS Signalling Channel is opened between the endpoint and the Gatekeeper. The RAS Signalling Channel is opened prior to the establishment of any other channels between H.323 endpoints. This channel is described in detail in clause 7.

6.2.10 Call signalling function

The call signalling function uses H.225.0 call signalling to establish a connection between two H.323 endpoints. The Call Signalling Channel is independent from the RAS Channel and the H.245 Control Channel. H.245 open logical channel procedures are not used to establish the Call Signalling Channel. The Call Signalling Channel is opened prior to the establishment of the H.245 Channel and any other logical channels between H.323 endpoints. In systems that do not have a Gatekeeper, the Call Signalling Channel is opened between the two endpoints involved in the call. In systems which contain a Gatekeeper, the Call Signalling Channel is opened between the endpoint and the Gatekeeper or between the endpoints themselves as chosen by the Gatekeeper. This channel is described in detail in clause 7.

6.2.11 H.225.0 layer

Logical channels of video, audio, data or control information are established according to the procedures of ITU-T Rec. H.245. Logical channels are unidirectional and are independent in each direction of transmission. Some logical channels, such as for data, may be bidirectional and are associated through the bidirectional open logical channel procedure of ITU-T Rec. H.245. Any number of logical channels of each media type may be transmitted, except for the H.245 Control Channel of which there shall be one per call. In addition to the logical channels, H.323 endpoints use two signalling channels for call control, and Gatekeeper related functions. The formatting used for these channels shall conform to ITU-T Rec. H.225.0.

6.2.11.1 Logical channel numbers

Each logical channel is identified by a Logical Channel Number, in the range 0 to 65535, which serves only to associate logical channels with the transport connection. Logical channel numbers are selected arbitrarily by the transmitter, except that logical channel 0 shall be permanently assigned to the H.245 Control Channel. The actual Transport Address that the transmitter shall transmit to shall be returned by the receiver in the **openLogicalChannelAck** message.

6.2.11.2 Logical channel bit rate limits

A logical channel's bandwidth shall have an upper limit specified by the minimum of the endpoint's transmit capability (if present) and the receiving endpoint's receive capability. Based on this limit, an endpoint shall open a logical channel at a bit rate at or below this upper limit. A transmitter shall transmit an information stream within the logical channel at any bit rate at or below the open logical channel bit rate. The limit applies to the information streams which are the content of the logical channel(s), not including RTP headers, RTP payload headers and network headers and other overhead.

H.323 endpoints shall obey the **flowControlCommand** message of H.245, which commands a limit to the bit rate of a logical channel or the aggregate bit rate of all logical channels. H.323 endpoints that want to limit the bit rate of a logical channel or the aggregate bit rate of all logical channels should send the **flowControlCommand** message to the transmitting endpoint.

When the terminal has no information to send in a given channel, the terminal shall send no information. Fill data shall not be sent on the network in order to maintain a specific data rate.

6.3 Gateway characteristics

The Gateway shall provide the appropriate translation between transmission formats (for example H.225.0 to/from H.221) and between communications procedures (for example H.245 to/from H.242). This translation is specified in ITU-T Rec. H.246. The Gateway shall also perform call setup and clearing on both the network side and the SCN side. Translation between video, audio, and data formats may also be performed in the Gateway. In general, the purpose of the Gateway (when not operating as an MCU) is to reflect the characteristics of a network endpoint to an SCN endpoint, and the reverse, in a transparent fashion.

An H.323 endpoint may communicate with another H.323 endpoint on the same network directly and without involving a Gateway. The Gateway may be omitted if communications with SCN terminals (terminals not on the network) are not required. It may also be possible for a terminal on one segment of the network to call out through one Gateway and back onto the network through another Gateway in order to bypass a router or a low bandwidth link.

The Gateway has the characteristics of an H.323 Terminal or MCU on the network and of the SCN terminal or MCU on the SCN. The choice of terminal or MCU is left to the manufacturer. The Gateway provides the necessary conversion between the different terminal types. Note that the Gateway may initially operate as a terminal, but later using H.245 signalling begin to operate as an MCU for the same call that was initially point-to-point. Gatekeepers are aware of which terminals are Gateways since this is indicated when the terminal/Gateway registers with the Gatekeeper.

A Gateway which passes T.120 data between the SCN and the network may contain a T.120 MCS Provider which connects the T.120 MCS Providers on the network to the T.120 MCS Providers on the SCN.

Four examples of an H.323 Gateway are shown in Figure 5. The diagrams show the H.323 terminal or MCU function, the SCN terminal or MCU function, and the conversion function. The H.323 terminal function has the characteristics described in 6.2. The H.323 MCU function has the characteristics described in 6.5. The Gateway appears to the other H.323 terminals on the network as one or more H.323 terminals or an H.323 MCU. It communicates with the other H.323 terminals using the procedures in this Recommendation.

The SCN terminal or MCU function has the characteristics described in the appropriate Recommendation (H.310, H.320, H.321, H.322, H.324, V.70, GSTN or ISDN speech only terminals). The Gateway appears to the terminals on the SCN as one or more of the same terminal types or MCUs. It communicates to another terminal on the SCN using the procedures described in the appropriate Recommendation for that terminal. SCN signalling procedures are beyond the scope

of this Recommendation, including such topics as whether the H.323 Gateway appears as a terminal or a network to the SCN. Note that a Gateway may convert H.323 directly to H.324 or H.310 without going to H.320.

Gateways supporting interworking with speech only terminals on GSTN or ISDN should generate and detect DTMF signals corresponding to H.245 **userInputIndications** for 0-9, *, and #. Additionally, gateways may be able to generate and detect DTMF, telephony tones and telephony signals corresponding to these events transported with a special RTP payload type, as described in 10.5.

The conversion function provides the necessary conversion of transmission format, control, audio, video, and/or data streams between the different terminal Recommendations. At a minimum, the Gateway shall provide a conversion function for the transmission format, call setup signals and procedures, and communications control signals and procedures. When required, the Gateway shall provide for H.242 to H.245 conversion. The Gateway performs the appropriate conversion between the H.225.0 Call Signalling and the SCN signalling system (Q.931, Q.2931, etc.). The conversion between H.225.0 call signalling messages on the network and Q.931 messages on the SCN is described in ITU-T Rec. H.246.

All call signalling received by the Gateway from an SCN endpoint and not applicable to the Gateway should be passed through to the network endpoint and vice versa. This signalling includes, but is not limited to, Q.932, Q.950 and H.450-series messages. This will allow H.323 endpoints to implement the Supplementary Services defined in those Recommendations. The handling of other SCN call signalling systems is for further study.

This Recommendation describes the connection of one H.323 terminal on the network to one external terminal on the SCN through the Gateway. The actual number of H.323 terminals that can communicate through the Gateway is not subject to standardization. Similarly, the number of SCN connections, number of simultaneous independent conferences, audio/video/data conversion functions, and inclusion of multipoint functions is left to the manufacturer. If the Gateway includes an MCU function on the network side, that function shall be an H.323 MCU on the network side. If the Gateway includes an MCU function on the SCN side, it may appear as an H.231/H.243 MCU or as an MCU for H.310 or H.324 systems (these MCUs are indicated as for further study in the respective Recommendations) on the SCN side.

A Gateway may be connected via the SCN to other Gateways to provide communication between H.323 terminals which are not on the same network.

Equipment which provides transparent interconnection between networks without using H-series protocols (such as routers and remote dial in units) are not Gateways as defined within the scope of this Recommendation.

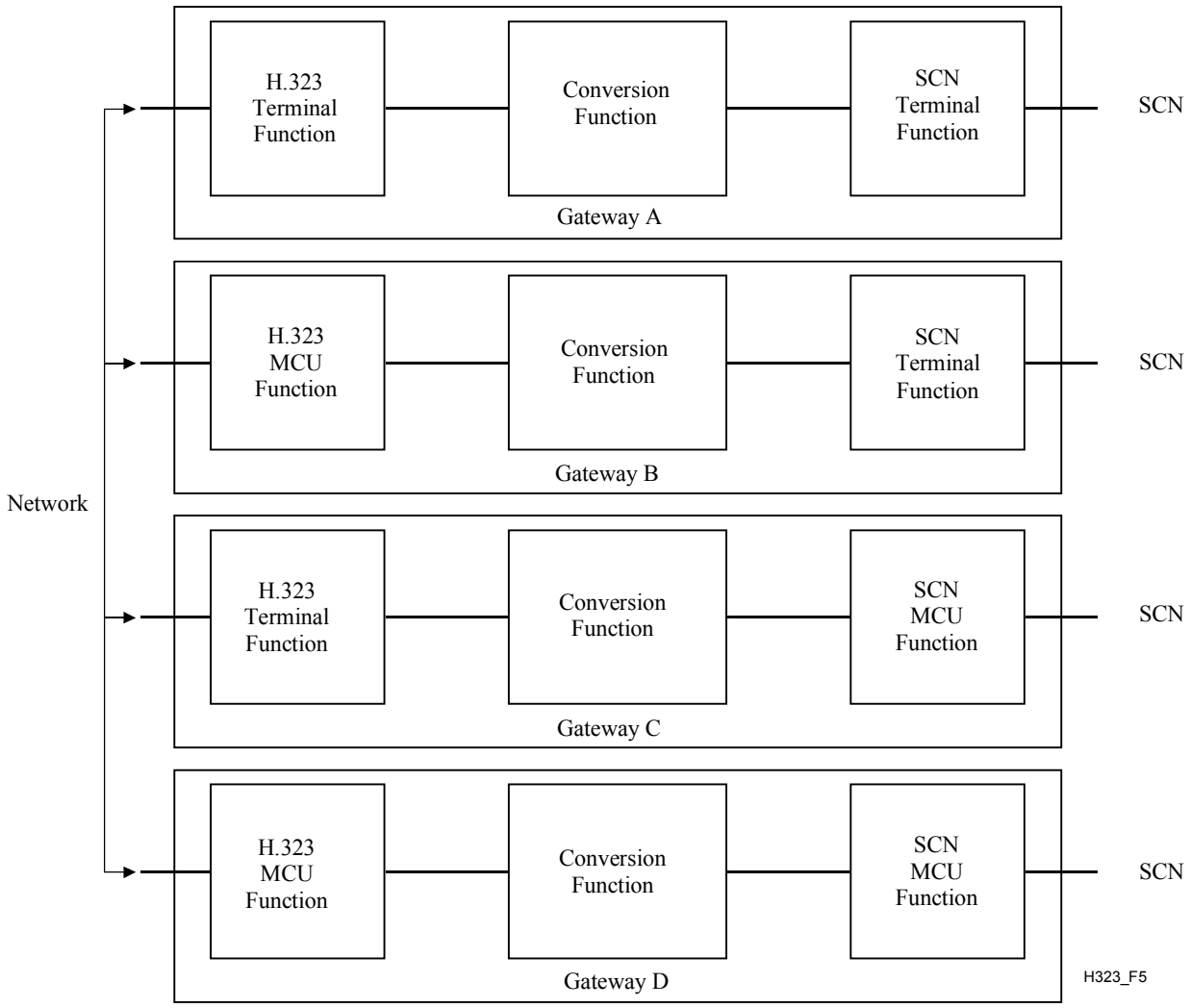


Figure 5/H.323 – H.323 gateway configurations

6.3.1 Gateway decomposition

This clause identifies a group of interfaces and functions to be used to decompose H.323 Gateways. It addresses each interface and its resulting protocol, but certain Gateway implementations may choose to group two or more functional components into a single physical device. For this reason, interfaces may provide a capability to transparently backhaul other protocols.

In Figure 6, the packet/circuit media component terminates SCN media channel and converts these streams to packet based media on the packet network interface. Interface A represents the device control protocol defined in ITU-T Rec. H.248, which is used to create, modify and delete Gateway media connections. The control logic component will accomplish signalling interworking between the SCN and H.323 sides of the Gateway.

The B interface represents the H.225.0 and H.245 protocol components that make up the H.323 signalling interfaces on the packet side of the Gateway.

Interface C will describe the ISDN type call control function between the FAS SCN services and the Gateway control logic. Interface D is a protocol that conveys the NFAS SCN signalling to the controller. This decomposition provides the flexibility to conserve SS7 code points and allows the SS7 switch to serve multiple decomposed Gateway Controllers.

The resource control elements differentiate between a high level understanding of resources in the Gateway Controller and a lower level understanding of resources in a Gateway device.

The SCN interfaces are described as a low-level interface that transports signalling and a high level SCN signalling termination that interfaces with the controller of this Gateway. This can be FAS signalling, such as ISDN PRI, or NFAS signalling, such as SS7.

Figure 6 does not represent a physical decomposition at this point. The challenge for Gateway vendors is to group these components into physical devices and implement the associated interfaces in order to produce highly scaleable, multi-vendor H.323 Gateways. The X interface is the external H.323 interface, the Y interface is the external packet media interface (i.e., RTP) and the Z interface is the external SCN interface.

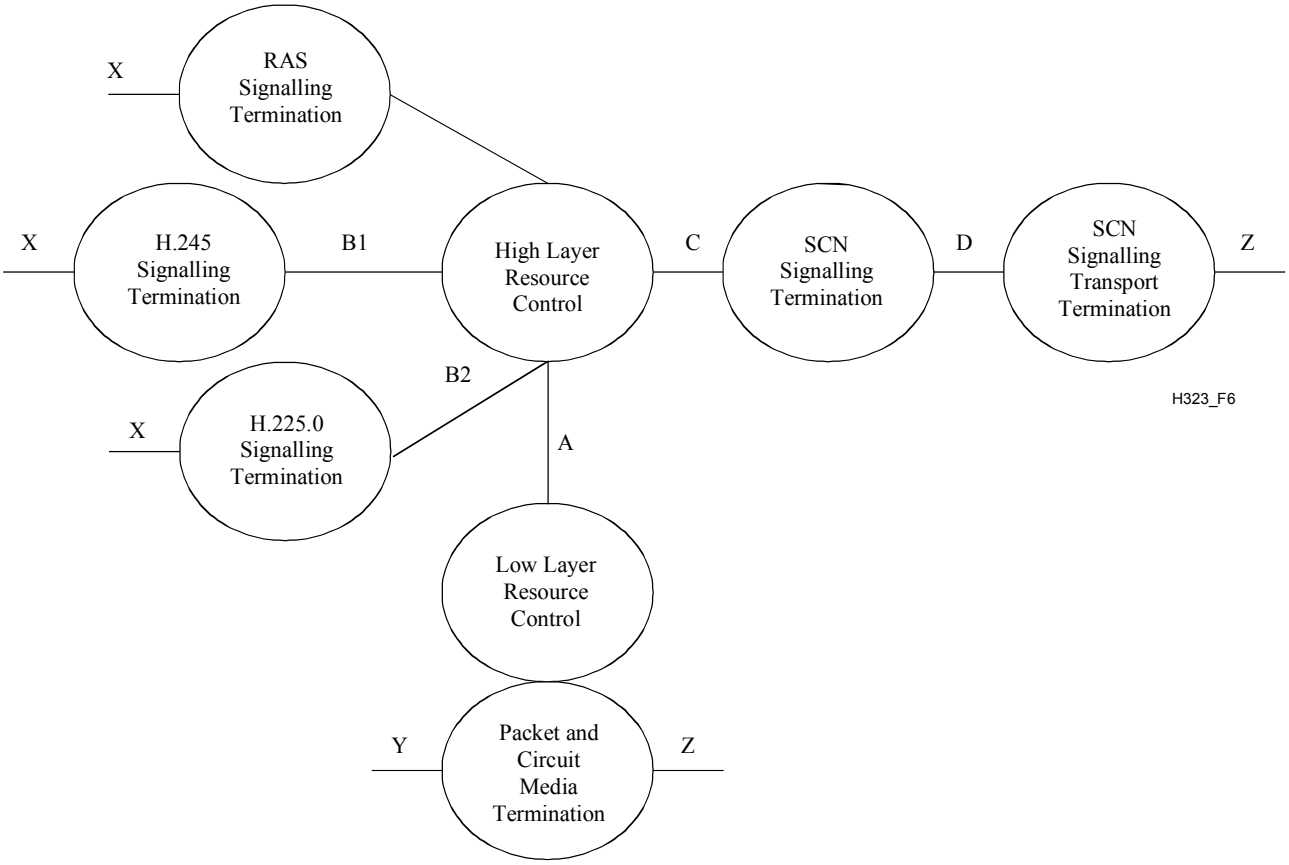


Figure 6/H.323 – Functional architecture of the decomposed gateway

6.3.1.1 Physical decompositions

This clause describes examples of possible Gateway decompositions and the internal interfaces that are required. In all cases the external interfaces, such as H.323 and SCN, remain unchanged. The controller portion of the physical Gateway is called the Media Gateway Controller (MGC). The MGC's functions are to:

- handle H.225.0 RAS messaging with an external gatekeeper;
- optionally handle the SS7 signalling interface;
- optionally handle the H.323 signalling interface.

The Media Gateway (MG) component:

- terminates the IP network interface;
- terminates the SCN network span;
- may handle H.323 signalling in some physical decompositions;
- may handle FAS SCN signalling in some physical decompositions.

Decomposed Gateways need not realize all interfaces but the MGC/MG split exposing interface A is a mandatory part of all decompositions. This will allow an MGC to control different types of MGs that may be optimized for certain applications (e.g., voice versus multimedia H.320/H.323 Gateways). The decomposition of interfaces B and C on the MG, which may require a protocol to backhaul signalling from the MG to the MGC, is for further study.

The MG terminates the IP or ATM media on the packet network side and bearer channels on the SCN network interfaces. The packet side may be IP, ATM, or an ATM network interface where audio and video packets traverse native ATM connections according to Annex C.

The MGC and MG differentiate between high-level and low-level resource management elements. The MGC is responsible for high-level resource management where it understands the availability of resources, such as echo cancellers, but does not assign specific resources to specific Gateway sessions. The MG is responsible for low-level resource allocation and management, as well as the hardware manipulations required to switch and process media streams within the Media Gateway.

6.3.1.1.1 Separate SS7 gateways

Figure 7 represents one possible Gateway decomposition for an ISUP-to-H.323 Gateway, where the SS7 Gateway, MGC, and MG functions are decomposed into separate physical devices. This arrangement exposes an ISUP signalling transport interface D and the device control interface A.

To facilitate interoperability, decomposed Gateway configurations must support interface A and contain internal H.323 and SCN signalling in the MGC.

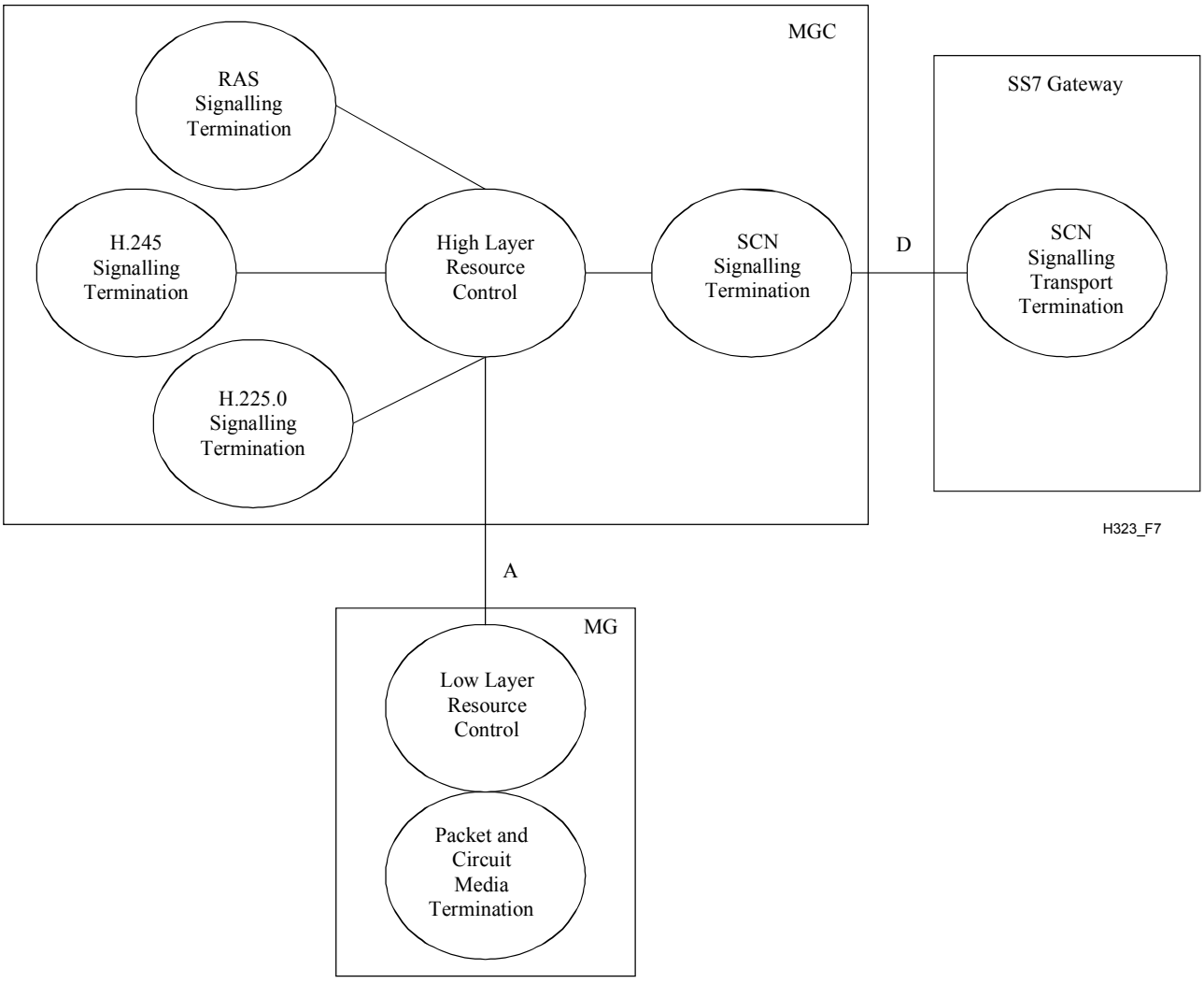


Figure 7/H.323 – SS7 gateway decomposition

6.3.1.1.2 FAS gateway decomposition

The Gateway decomposition shown in Figure 8 isolates the FAS SCN services, such as ISDN PRI on the MG, and retains the H.323 signalling on the MGC. This exposes the C and A interfaces between the MG and MGC.

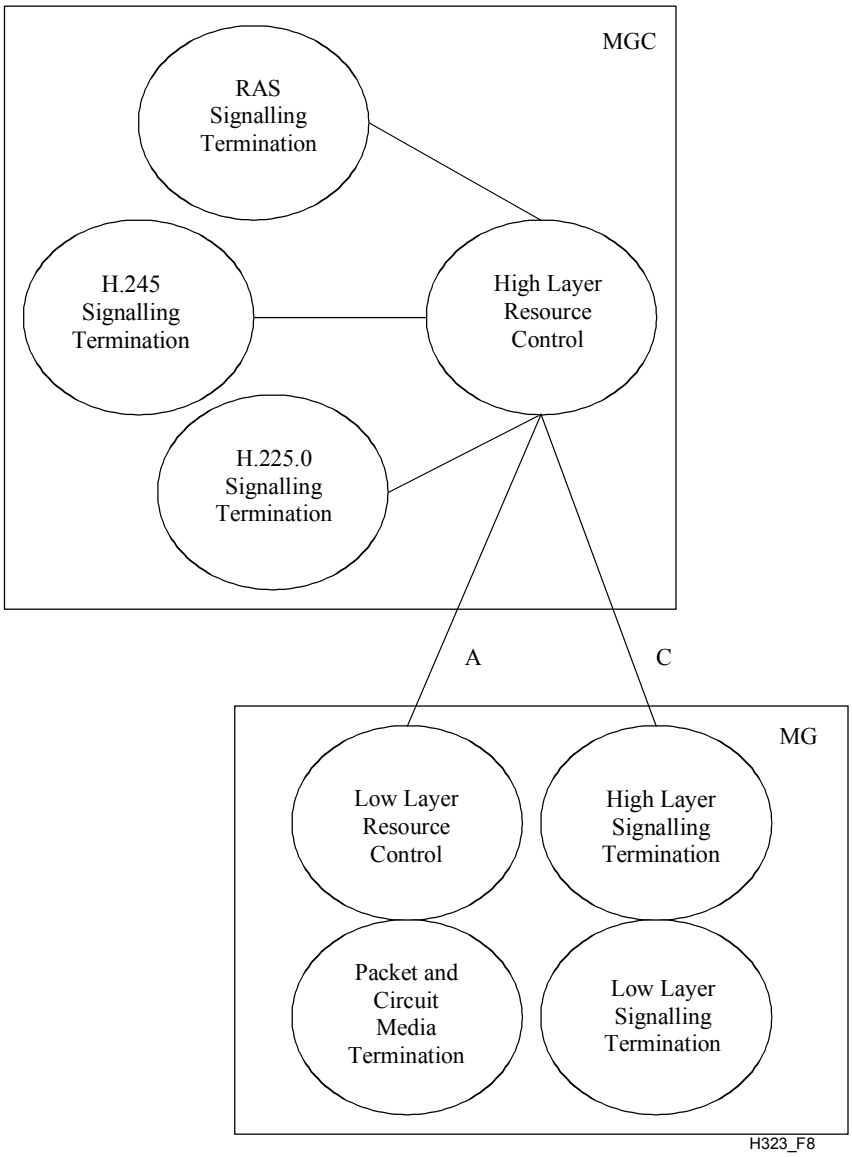


Figure 8/H.323 – FAS gateway with H.323 signalling in MG

6.3.1.1.3 SS7 gateway with H.323 signalling in the MG

The decomposition shown in Figure 9 leverages the SS7 interface of the MGC and deploys the H.323 signalling on the MG exposing interfaces D, A and B.

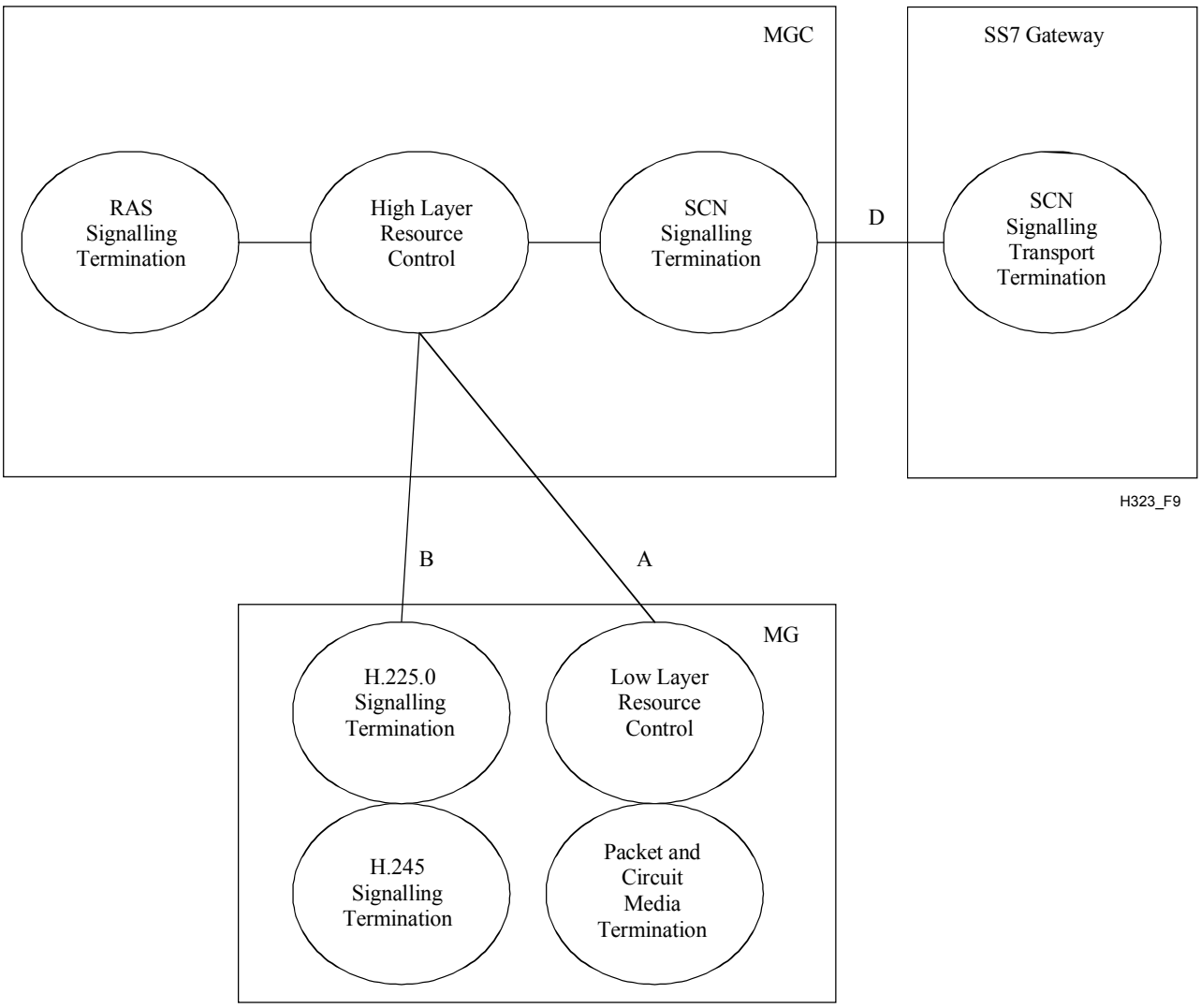


Figure 9/H.323 – SS7 terminated in the media gateway

6.3.1.1.4 FAS and H.323 signalling in the media gateway

Requirements exist for H.320 Gateways that are decomposed such that H.323 and SCN signalling are both present on the MG, along with the packet and circuit terminations. In this decomposition, signalling is handled locally by the MG and event notifications are reported to the MGC (see Figure 10).

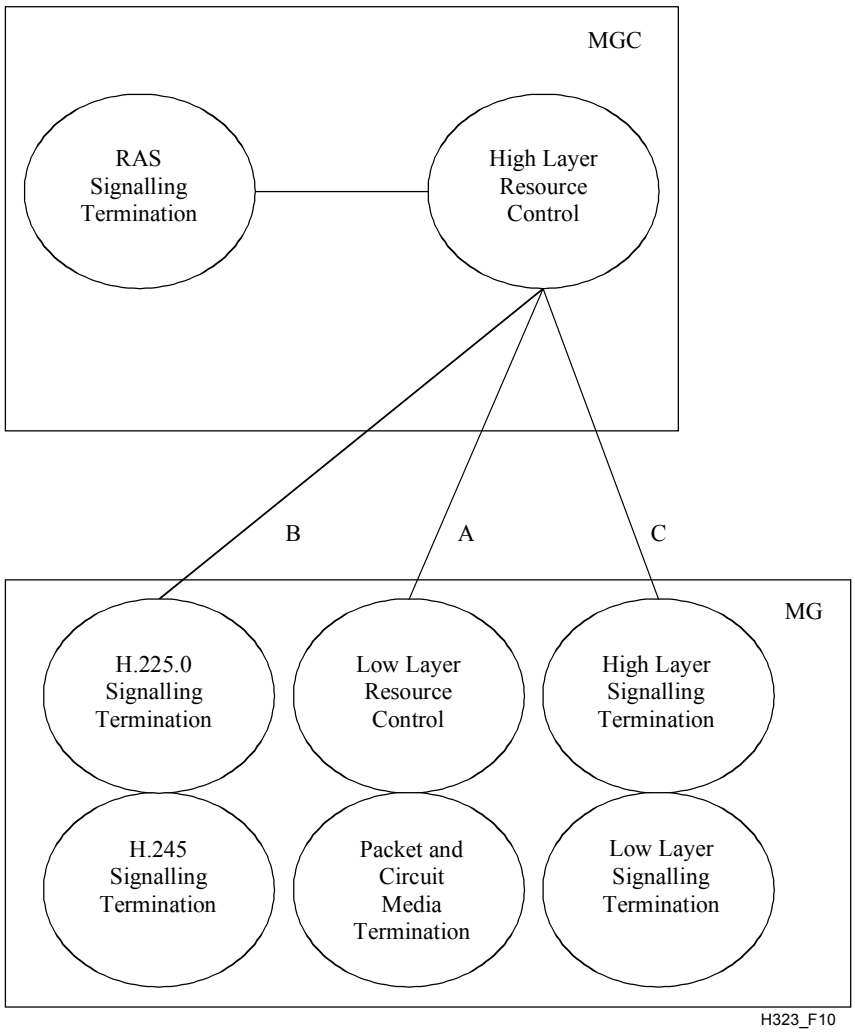


Figure 10/H.323 – FAS and H.323 signalling in the MG

6.3.1.1.5 SS7 in the media gateway

The decomposition shown in Figure 11 terminates the SS7 network in the MG and exposes the D interface between the MGC and MG.

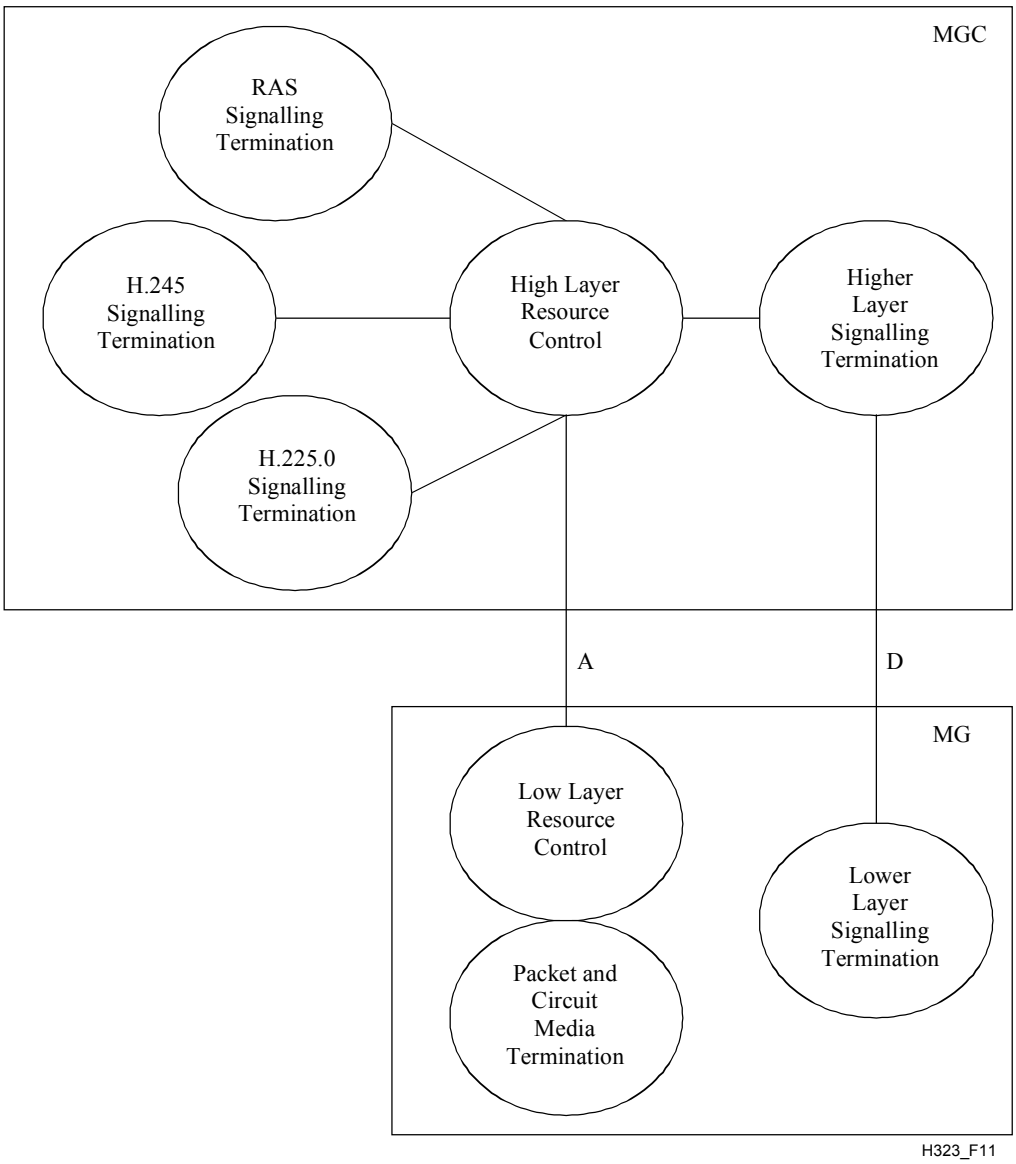


Figure 11/H.323 – SS7 terminated in the MG

6.3.2 Gateway applications

There are many applications for decomposed and composite gateways. Vendors and/or carriers may decide to use a composite or decomposed gateway depending on the application requirements. Decomposed gateways are mandated by H.248 to interwork with composite gateways.

This clause discusses some shared vocabulary between H.323 equipment, the SCN and H.248. It also provides examples of applications gateways. It is not meant to be a comprehensive list of all applications. It is also not intended to illustrate the only way such applications can be supported. In this clause the terms MG, MGC, and GW represent physical instantiations of these devices.

6.3.2.1 Overview of trunking and access gateways

The terms trunking and access gateways are used in both H.323 and H.248, and are also part of the terminology of circuit switching, where they are applied to tandem and access switches. Because the same words are used to mean different things in the context of three different architectures, this clause attempts to clarify the variations in terminology.

6.3.2.1.1 SCN terminology

In the SCN, a "tandem" or "trunking" switch refers to a switch that connects networks using an NNI protocol, such as SS7/ISUP or a CAS NNI protocol. An "access" switch refers to a switch that has user connections using BRI/PRI and is also connected via NNI protocols to a larger network. A "mixed" switch may have both functions.

6.3.2.1.2 H.323 terminology

In H.323 networks, a "trunking" gateway refers to a gateway that provides a true tandem function that is transparent to the attached networks. These attached networks might be SS7 networks, QSIG networks, or other networks. However, in all cases tunnelling is used to create full transparency and a true tandem function. Interworking between ISUP flavours is considered to take place outside of the H.323 network. Tunnelling is based on H.225.0 protocol negotiation and Annex M.

An H.323 "access" gateway provides an interworking function from another network, enterprise, or endpoint that is not fully transparent. The interworked protocols might include:

- SS7/ISUP, using Annex C/H.246;
- QSIG using H.450;
- H.320 using Annex A/H.246.

It should be noted that the H.323 "trunking" Gateway and the SCN "tandem" switch are filling the same function, but the H.323 "access gateway" and the SCN "access switch" fill very different roles. A particular point of confusion is that H.225.0 acts as both UNI and NNI signalling in the H.323 network, filling the roles of both ISUP and ISDN (BRI/PRI) in the SCN. H.323 does not make the kind of UNI/NNI signalling distinctions found in the SCN and call signalling is the same whether between endpoints directly or when mediated via network elements like an H.323 Gatekeeper or Border Element (BE).

Figure 12 summarizes the points above and also shows the relationship between H.323 domains, which have some SCN network-like characteristics. However, it is important to keep in mind that H.225.0 is also used for all call signalling, whether between terminals, zones, or domains. In addition, zones and domains are fundamentally virtual rather than physical, and switches (e.g., ATM switches used to route IP), although they may be present, are not visible from above the IP layer in the packet network.

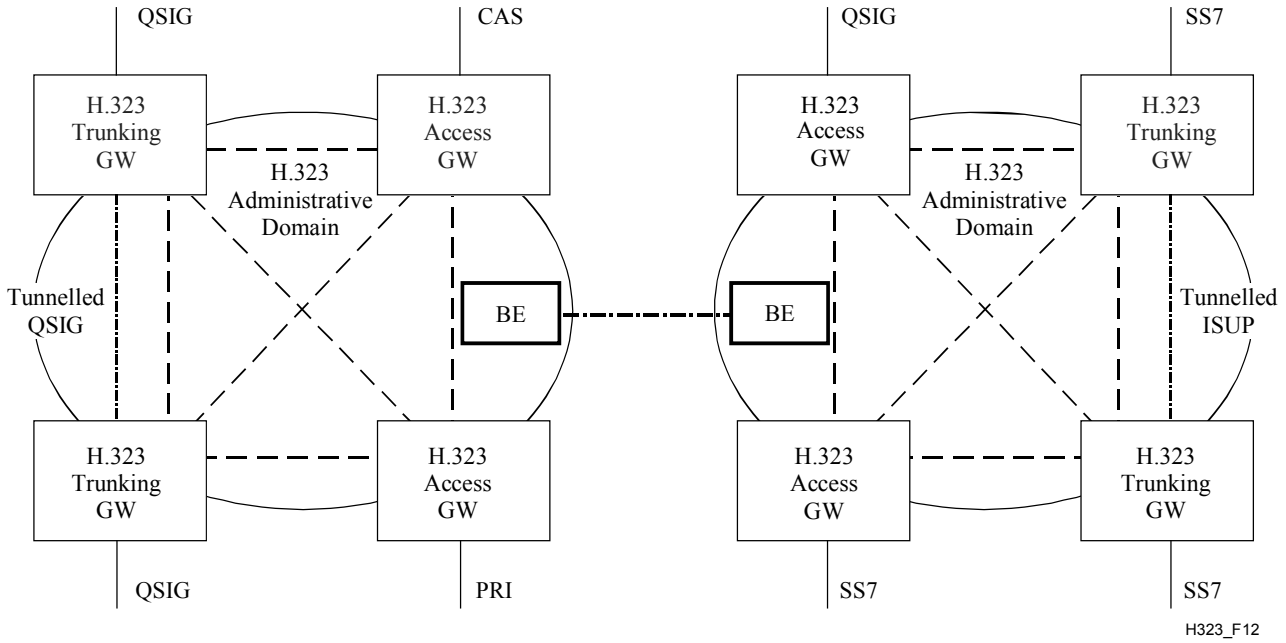


Figure 12/H.323 – Relationship of H.323, SCN and H.248 gateways

6.3.2.1.3 H.248 terminology

ITU-T Rec. H.248 also uses the terms "trunking" and "access" gateways. Noting that H.248 devices can be viewed as simply decompositions of H.323 composite gateways into MGC and MGs, it is assumed that the MGCs support H.323 and interwork using H.225.0, just as any other H.323 Gateway, including tunnelling of ISUP, etc. However, when viewed from a decomposed perspective, the terms take on slightly variant meanings. A "trunking" gateway is one in which the signalling is connected directly to the MGC, i.e., ISUP, while an "access" Gateway is one in which the signalling arrives at the MG and is then is passed via H.248 to the MGC. It is important to note that, although an "access" Gateway may support a UNI protocol, it may also support NNI CAS protocols, so that defining an H.248 "access" Gateway as a Gateway supporting a UNI interface is not accurate.

Figure 13 illustrates the architecture of ITU-T Rec. H.248. It should be noted that composed H.323 gateways are often used as "access" gateways in H.248 systems as illustrated. The diagram shows a collocated H.248 MGC and H.323 Gatekeeper.

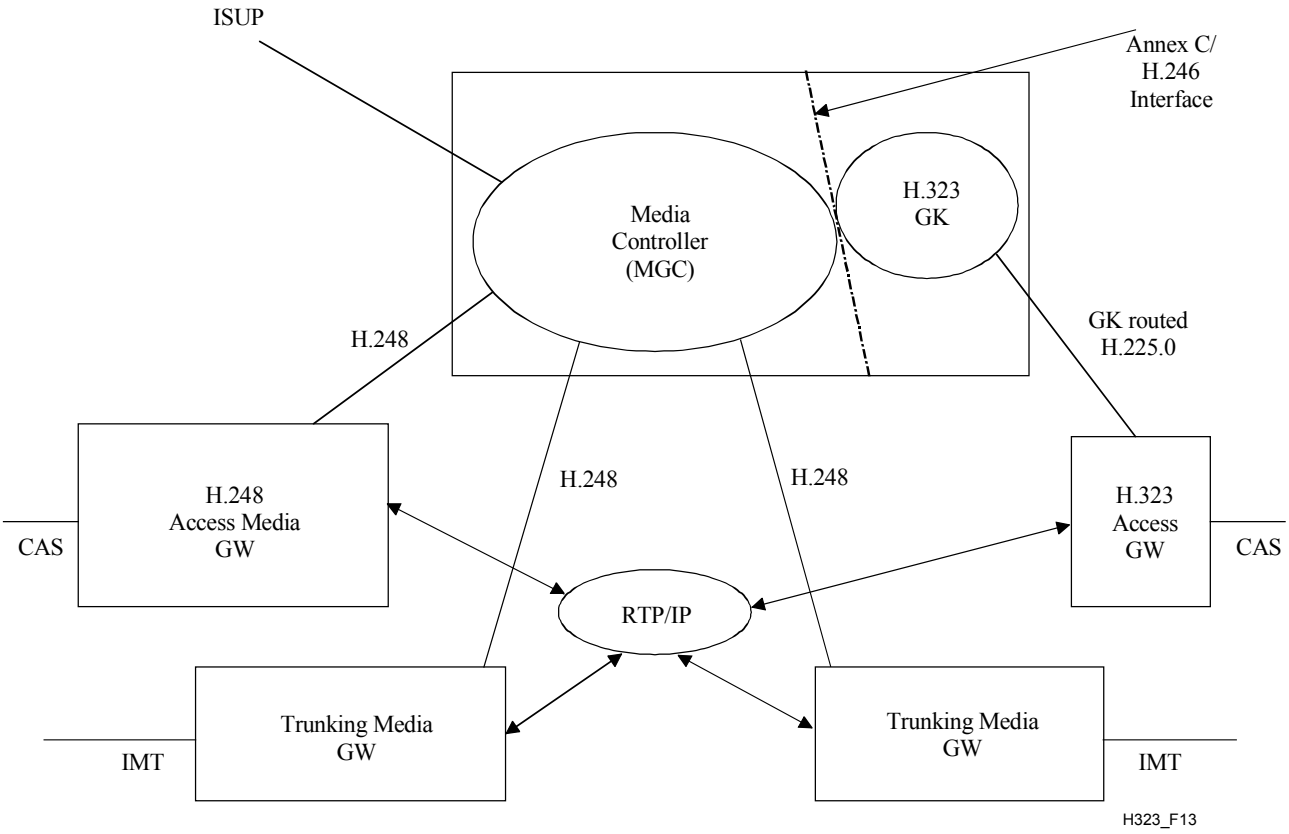


Figure 13/H.323 – Relationship of H.323 and H.248

6.3.2.2 Service provider trunking gateways

Figure 14 shows an example of a call routed across a packet switched network between two service provider trunking Gateways. In this application, the packet network acts as a tandem voice network for the service provider. For this application, interface A is used to control the Media Gateways. The packet network connects to the Switched Circuit Network via SS7 signalling and inter-machine trunks. Figure 14 illustrates the case in which SS7 A-links are used to connect to the SS7 network. In this case, the MGC terminates the signalling links directly instead of via a signalling Gateway. The MGCs pass signalling information between each other using interface X (for example, by tunnelling ISUP in an H.225.0 connection). The voice traffic flows between the two Gateways.

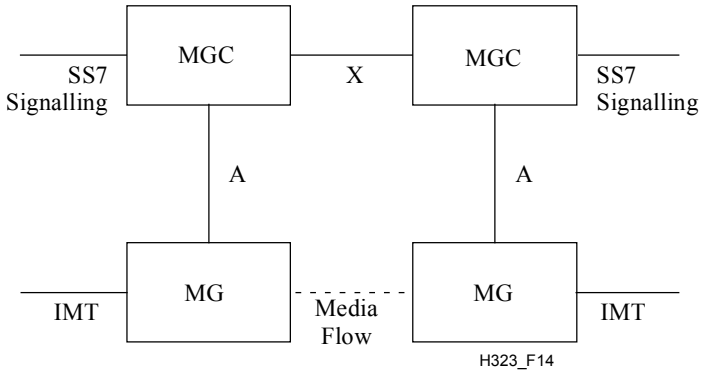


Figure 14/H.323 – Two decomposed service provider trunking gateways

6.3.2.3 Service provider access gateways

Figure 15 represents an example of a call routed across a packet switched network between a composite H.323 Service Provider Access Gateway and a decomposed service provider trunking Gateway. In this application, the service provider is providing a channel associated signalling interface to an enterprise PBX system for carrying voice calls over the provider's network. H.225.0 call signalling is used between the composite Gateway and the decomposed Gateway. The MGC performs the appropriate SS7 signalling to communicate with the service provider's SS7 network and SCN. In this example, X is H.225.0 and the MGC implements an Annex E/H.246 interworking function.

Although Recommendations exist which describe the interworking between various protocols, such as ISUP and H.323, service providers and manufacturers should carefully consider when it is appropriate to perform such interworking and the number of such interworking points. Interworking may not result in a perfect translation between two protocols and multiple translations may lead to a higher loss of transparency.

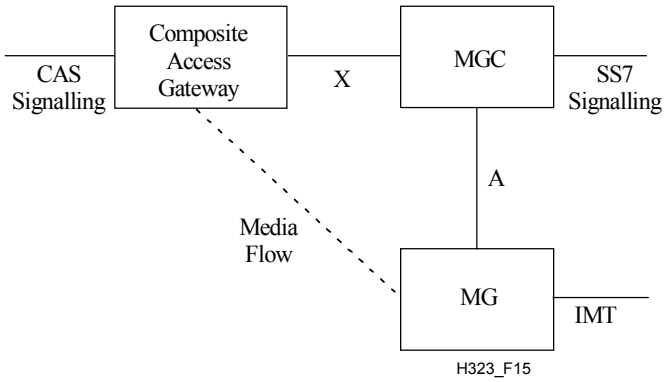


Figure 15/H.323 – A composite access gateway and decomposed trunking gateway

Figure 16 illustrates the same application in which the service provider access Gateway is also decomposed. In this case, interface A is used to control the channel associated signalling. The MGCs communicate with each other using interface X. In this particular case, if there is no signalling backhaul between the MG and the MGC, the amount of information on the call available to the MGC will be limited to what is defined by ITU-T Rec. H.248. In this example, X is H.225.0 and the MGC on the right is performing Annex E/H.246 ISUP interworking.

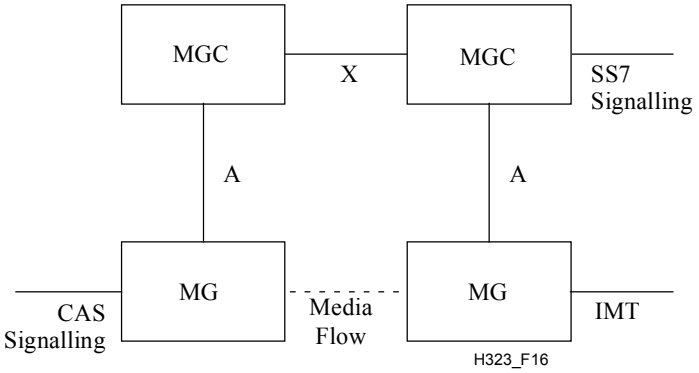


Figure 16/H.323 – Decomposed service provider access and trunking gateways

In considering which of these approaches might be best for a particular application, the following factors should be considered:

- Number of lines to be connected.
- Cost of trunks.
- Homologation issues.
- Capacity of the MGC.
- Number of access Gateways relative to trunking Gateways.
- Type of CAS protocols to be supported.
- Service provider call processing architecture.
- Network design.

For access Gateways, the application environment will determine whether a decomposed Gateway, an H.323 terminal using H.450.x, an Annex L stimulus terminal, or a composite Gateway is the most appropriate.

6.3.2.4 Enterprise trunking gateways

Figure 17 illustrates an enterprise gateway that is used between PBXs in a private voice network. The packet network is used instead of leased lines to connect the PBXs. In this case, QSIG is used for signalling between the PBXs. Since QSIG is a facility associated signalling type, the signalling may be backhauled from the Media Gateway to the Media Gateway Controller via interface C. Interface A is used between the MGC and MG for gateway control. MGCs communicate between each other over interface X, which may be H.225.0 tunnelling QSIG according to Annex M1.

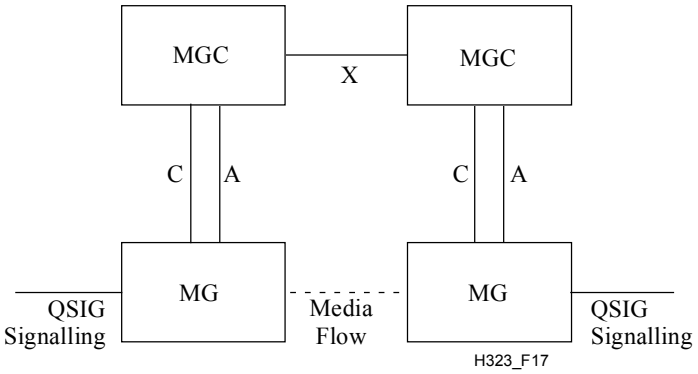


Figure 17/H.323 – Decomposed enterprise trunking gateways

Figure 18 illustrates Gateways that are used between PBXs in a private voice network. The packet network is used instead of leased lines to connect the PBXs. In this case, QSIG is also used for signalling between the PBXs. However, QSIG tunnelling over interface X is used to carry QSIG signalling between a composite Gateway and a decomposed Gateway. Other combinations, such as composite-composite and decomposed-decomposed, could also be used.

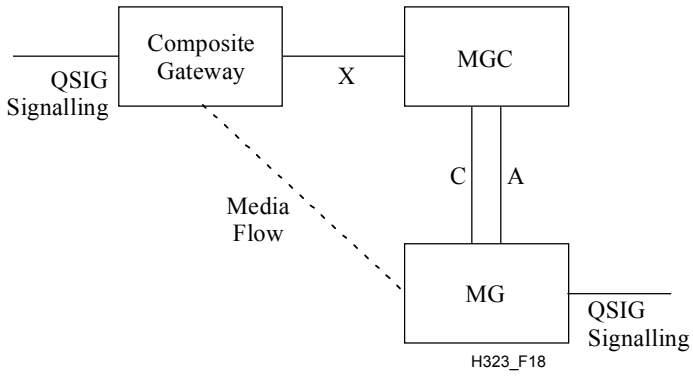


Figure 18/H.323 – QSIG tunnelling example

6.3.2.5 Enterprise to service provider access gateways

In some cases, an enterprise H.323 network will communicate with the PSTN via a decomposed gateway. This is illustrated in Figure 19. In this case, the decomposed gateway communicates to the H.323 endpoints via H.323 signalling (H.225, H.245, etc.). The decomposed gateway connects to the PSTN via ISDN PRI. The D-channel signalling can be backhauled via interface C.

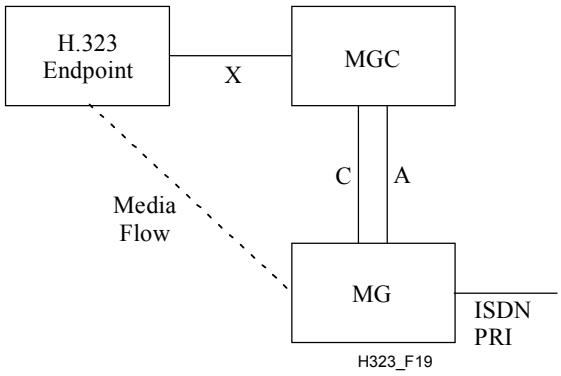


Figure 19/H.323 – Decomposed gateway and H.323 endpoint

Another enterprise access application uses H.248 to manage terminals, but appears as a composite Gateway to composite Gateways on other premises as shown in Figure 20. In this example, H.450.x is used to provide supplementary services interworking.

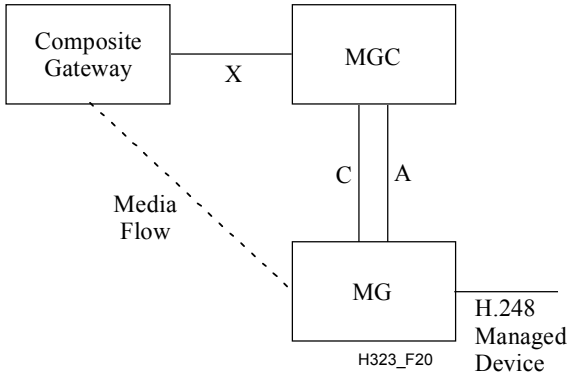


Figure 20/H.323 – Composite gateway and H.248 managed devices

An additional enterprise access application uses Annex L to manage terminals, but appears as a composite Gateway to composite Gateways on other premises as shown in Figure 21. In this example, H.450.x may be used to provide supplementary services interworking. In this example, X1 is H.225.0 with H.450, while X2 is H.225.0 with Annex L stimulus signalling.

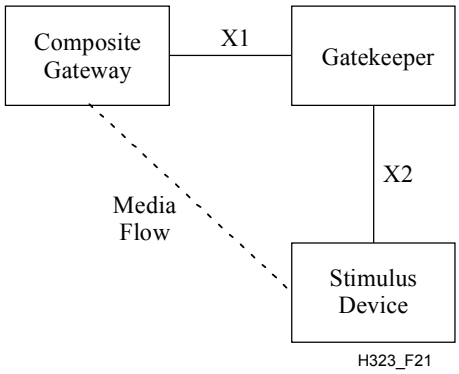


Figure 21/H.323 – Composite gateway and Annex L device

It should be noted that the Annex L terminals of Figure 21 may interwork with the H.248 managed terminals of Figure 20 using H.450.x. These configurations allow extensive feature innovation on the enterprise while supporting inter-enterprise interoperability using H.450.x. Note that Gatekeeper routed call signalling is used in Figure 21 in the enterprise Gatekeeper managing the Annex L terminals, although the other enterprise Gateways may use the direct call model and have a different Gatekeeper.

6.4 Gatekeeper characteristics

The Gatekeeper, which is optional in an H.323 system, provides call control services to the H.323 endpoints. More than one Gatekeeper may be present and communicate with each other in an unspecified fashion. The Gatekeeper is logically separate from the endpoints; however, its physical implementation may coexist with a terminal, MCU, Gateway, MC, or other non-H.323 network device.

There is one and only one Gatekeeper in a Zone at any given time, although multiple distinct devices may provide the Gatekeeper function in a Zone. Multiple devices that provide the RAS signalling function for the Gatekeeper are referred to as Alternate Gatekeepers. Each Alternate Gatekeeper may appear to endpoints as a distinct Gatekeeper. Communication between Alternate Gatekeepers and other devices that provide the Gatekeeper function for the Zone is outside the scope of the Recommendation.

When it is present in a system, the Gatekeeper shall provide the following services:

- Address Translation – The Gatekeeper shall perform alias address to Transport Address translation. This should be done using a translation table which is updated using the Registration messages described in clause 7. Other methods of updating the translation table are also allowed.
- Admissions Control – The Gatekeeper shall authorize network access using ARQ/ACF/ARJ H.225.0 messages. This may be based on call authorization, bandwidth, or some other criteria which is left to the manufacturer. It may also be a null function which admits all requests.
- Bandwidth Control – The Gatekeeper shall support BRQ/BRJ/BCF messages. This may be based on bandwidth management. It may also be a null function which accepts all requests for bandwidth changes.

- Zone Management – The Gatekeeper shall provide the above functions for terminals, MCUs, and Gateways which have registered with it as described in 7.2.

The Gatekeeper may also perform other optional functions such as:

- Call Control Signalling – The Gatekeeper may choose to complete the call signalling with the endpoints and may process the call signalling itself. Alternatively, the Gatekeeper may direct the endpoints to connect the Call Signalling Channel directly to each other. In this manner, the Gatekeeper can avoid handling the H.225.0 call control signals. The Gatekeeper may have to act as the network as defined in ITU-T Rec. Q.931 in order to support supplementary services. This operation is for further study.
- Call Authorization – Through the use of the H.225.0 signalling, the Gatekeeper may reject calls from a terminal due to authorization failure. The reasons for rejection may include, but are not limited to, restricted access to/from particular terminals or Gateways and restricted access during certain periods of time. The criteria for determining if authorization passes or fails is outside the scope of this Recommendation.
- Bandwidth Management – Control of the number of H.323 terminals permitted simultaneous access to the network. Through the use of the H.225.0 signalling, the Gatekeeper may reject calls from a terminal due to bandwidth limitations. This may occur if the Gatekeeper determines that there is not sufficient bandwidth available on the network to support the call. The criteria for determining if bandwidth is available is outside the scope of this Recommendation. Note that this may be a null function, i.e., all terminals are granted access. This function also operates during an active call when a terminal requests additional bandwidth.
- Call Management – For example, the Gatekeeper may maintain a list of ongoing H.323 calls. This information may be necessary to indicate that a called terminal is busy and to provide information for the Bandwidth Management function.
- Alias Address Modification – The Gatekeeper may return a modified Alias Address. If the Gatekeeper returns an alias address in an ACF, the endpoint shall use the Alias Address in establishing the connection.
- Dialled Digit Translation – The Gatekeeper may translate dialled digits into an E.164 number or a Private Network number.
- Gatekeeper management information data structure – For further study.
- Bandwidth reservation for terminals not capable of this function – For further study.
- Directory services – For further study.

In order to support ad hoc Multipoint Conferences, the Gatekeeper may choose to receive the H.245 Control Channels from the two terminals in a point-to-point conference. When the conference switches to a multipoint conference, the Gatekeeper can redirect the H.245 Control Channel to an MC. The Gatekeeper need not process the H.245 signalling; it only needs to pass it between the terminals or the terminals and the MC.

Networks which contain Gateways should also contain a Gatekeeper in order to translate incoming **dialledDigits** or **partyNumber** (including **e164Number** and **privateNumber**) addresses into Transport Addresses.

H.323 entities that contain a Gatekeeper shall have a mechanism to disable the internal Gatekeeper so that when there are multiple H.323 entities that contain a Gatekeeper on a network, the H.323 entities can be configured into the same Zone.

6.5 Multipoint controller characteristics

The MC provides control functions to support conferences between three or more endpoints in a multipoint conference. The MC carries out the capabilities exchange with each endpoint in a multipoint conference. The MC sends a capability set to the endpoints in the conference indicating the operating modes in which they may transmit. The MC may revise the capability set that it sends to the terminals as a result of terminals joining or leaving the conference or for other reasons.

In this manner, the MC determines the Selected Communications Mode (SCM) for the conference. The SCM may be common for all endpoints in the conference. Alternatively, some endpoints may have a different SCM than other endpoints in the conference. The manner in which the MC determines an SCM is not within the scope of this Recommendation.

As part of multipoint conference setup, an endpoint will become connected to an MC on its H.245 Control Channel. This connection may occur:

– via an explicit connection with an MCU;

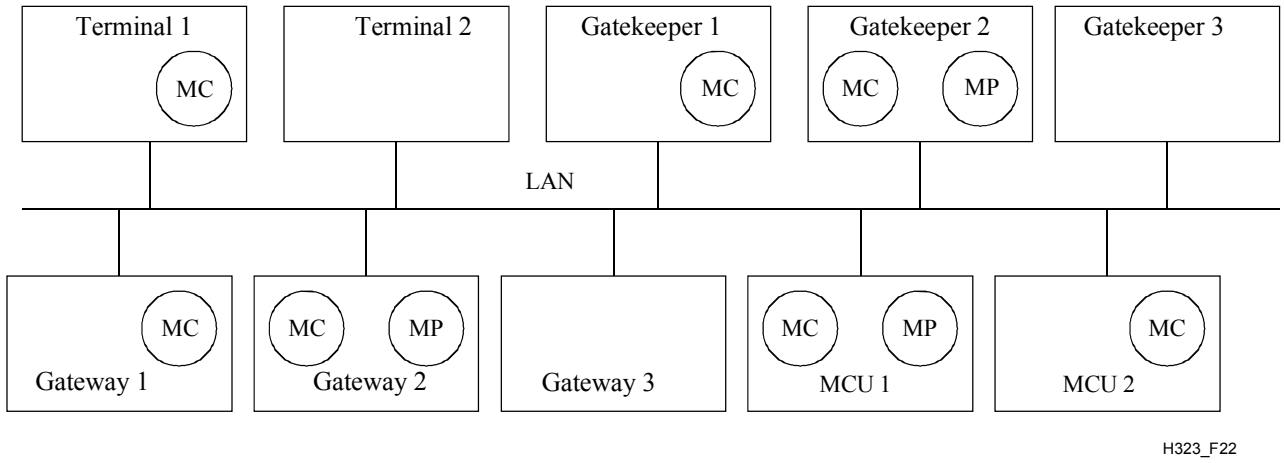
– via an implicit connection to the MC within a Gatekeeper;

– via an implicit connection to the MC within another terminal or Gateway in the multipoint conference;

– via an implicit connection through a Gatekeeper to an MCU.

The choice of conference mode (e.g., decentralized or centralized) occurs after connection with the MC using H.245 signalling. The choice of conference mode may be limited by the capability of the endpoints or the MC.

The MC may be located within a Gatekeeper, Gateway, terminal, or MCU. See Figure 22.



NOTE – Gateway, Gatekeeper and MCU can be a single device.

Figure 22/H.323 – Possible locations of MC and MP in H.323 system

An MC within a terminal is not callable. It can be included in the call in order to process the H.245 signalling to support ad hoc multipoint conferences. In this case, there may be no distinction between the MC and the H.245 Control Function (see 6.2.8) of the terminal. Communications between them are outside the scope of this Recommendation.

An MC located with the Gatekeeper is not callable; however, an MCU located with a Gatekeeper may be callable. An MCU located with a Gatekeeper may function as an independent MCU. An MC located with a Gatekeeper may be used to support ad hoc multipoint conferences when the Gatekeeper receives the H.245 Control Channels from the endpoints. In this manner, the Gatekeeper

can route the H.245 Control Channels to the MC at the start of the call or when the conference switches to multipoint.

The Gateway can function as a terminal or an MCU. When functioning as a terminal, the Gateway may contain an MC. This has the same characteristics as described above for an MC within a terminal.

An MCU always contains an MC. The MCU is callable and the MC processes the H.245 Control Channel from all of the endpoints.

When two or more endpoints are in a conference, the endpoints shall use the master slave resolution procedure of ITU-T Rec. H.245 to determine the MC that will control the conference.

After the capability exchange and master/slave determination, the MC may first assign a terminal number to a new endpoint using the **terminalNumberAssign**. The MC shall then notify the other endpoints of the new endpoint in the conference using **terminalJoinedConference**. The new endpoint may request a list of other endpoints in the conference using the **terminalListRequest**.

6.6 Multipoint processor characteristics

The MP receives audio, video and/or data streams from the endpoints involved in a centralized or hybrid multipoint conference. The MP processes these media streams and returns them to the endpoints.

Communications between the MC and the MP are not subject to standardization.

The MP may process one or more media stream types. When the MP processes video, it shall process the video algorithms and formats as described in 6.2.4. When the MP processes audio, it shall process the audio algorithms as described in 6.2.5. When the MP processes data, it shall process data streams as described in 6.2.7.

An MP which processes video shall provide either video switching or video mixing. Video switching is the process of selecting the video that the MP outputs to the terminals from one source to another. The criteria used to make the switch may be determined through detection of a change in speaker (sensed by the associated audio level) or through H.245 control. Video mixing is the process of formatting more than one video source into the video stream that the MP outputs to the terminals. An example of video mixing is combining four source pictures into a two-by-two array in the video output picture. The criteria for which sources and how many are mixed is determined by the MC until other controls are defined. The use of the T.120-series Recommendations for these control functions is for further study.

An MP which processes audio shall prepare N-audio outputs from M-audio inputs by switching, mixing, or a combination of these. Audio mixing requires decoding the input audio to linear signals (PCM or analogue), performing a linear combination of the signals and recoding the result to the appropriate audio format. The MP may eliminate or attenuate some of the input signals in order to reduce noise and other unwanted signals. Each audio output may have a different mix of input signals providing for private conversations. The terminals shall assume that their audio is not present in the audio stream returned to them. Terminal removal of its own audio from the MP audio output is for further study.

An MP which processes T.120 data shall be capable of acting as a non-leaf MCS provider and should be capable of acting as the Top MCS Provider. An MP may also process non-standard data, transparent user data and/or other types of data.

The MP may provide algorithm and format conversion, allowing terminals to participate in a conference at different SCMs.

The MP is not callable, the MCU which it is a part of is callable. The MP terminates and sources the media channels.

6.7 Multipoint control unit characteristics

The MCU is an endpoint which provides support for multipoint conferences. The MCU shall consist of an MC and zero or more MPs. The MCU uses H.245 messages and procedures to implement features similar to those found in ITU-T Rec. H.243.

A typical MCU that supports centralized multipoint conferences consists of an MC and an audio, video and data MP. A typical MCU that supports decentralized multipoint conferences consists of an MC and a data MP supporting ITU-T Rec. T.120. It relies on decentralized audio and video processing.

The network side of a Gateway may be an MCU. A Gatekeeper may also include an MCU. In either case, they are independent functions that happen to be colocated.

The MCU shall be callable by other endpoints using the procedures of clause 8.

6.8 Multipoint capability

6.8.1 Centralized multipoint capability

All endpoints shall have centralized multipoint capability. In this mode of operation they communicate with the MC of the MCU in a point-to-point manner on the control channel and with the MP on the audio, video and data channels. In this mode, the MC performs H.245 multipoint control functions, while the MP performs video switching or mixing, audio mixing, and T.120 multipoint data distribution. The MP transmits the resulting video, audio and data streams back to the endpoints. The MP may have the capability to convert between different audio, video and data formats and bit rates, allowing the endpoints to participate in the conference using different communications modes.

The MCU may use multicast to distribute the processed media streams if the endpoints in the conference can receive multicast transmissions. Multicast distribution of data is for further study.

This mode is signalled by the following H.245 capabilities: **centralizedControl**, **centralizedAudio**, **centralizedVideo** and **centralizedData**. Optionally, **distributedAudio** and **distributedVideo** may be used to indicate multicast distribution of media streams.

6.8.2 Decentralized multipoint capability

If the endpoints have decentralized multipoint capability, they communicate with the MC of an MCU, Gateway, Gatekeeper, or endpoint in a point-to-point mode on the H.245 Control Channel and optionally with an MP on data channels. The endpoints shall have the capability to multicast their audio and video channels to all other endpoints in the conference. The MC may control which endpoint or endpoints are actively multicasting audio and/or video (for example by using the **flowControlCommand** on either channel).

The endpoints receive multicast video channels and select one or more of the available channels for display to the user. The endpoints receive the multicast audio channels and perform an audio mixing function in order to present a composite audio signal to the user.

The MC may provide conference control functions such as chair control, video broadcast and video selection. This shall be done by receiving H.245 from an endpoint and then sending the appropriate control to other endpoints to enable or disable their video multicast. T.120 commands may optionally provide the same functions.

This mode is signalled by the following H.245 capabilities: **centralizedControl**, **distributedAudio**, **distributedVideo** and **centralizedData**.

6.8.3 Hybrid multipoint – Centralized audio

If the endpoints and MCU have hybrid multipoint-centralized audio capability, they may use distributed multipoint for video and centralized multipoint for audio. In this mode, the endpoints communicate with the MC in a point-to-point mode on the H.245 Control Channel and optionally with an MP on data channels.

The endpoints shall have the capability to multicast their video channels to all other endpoints in the conference. The MC may control which endpoint or endpoints are actively multicasting video. The endpoints receive multicast video channels and select one or more of the available channels for display to the user.

All of the endpoints in the conference transmit their audio channels to the MP. The MP performs the audio mixing function and outputs the resulting audio streams to the endpoints. The MP may produce an exclusive audio sum for each endpoint in the conference. Multicast distribution of processed audio is for further study.

This mode is signalled by the following H.245 capabilities: **centralizedControl**, **centralizedAudio**, **distributedVideo** and **centralizedData**.

6.8.4 Hybrid multipoint – Centralized video

If the endpoints and MCU have hybrid multipoint-centralized video capability, they may use distributed multipoint for audio and centralized multipoint for video. In this mode, the endpoints communicate with the MC in a point-to-point mode on the H.245 Control Channel and optionally with an MP on data channels.

The endpoints shall have the capability to multicast their audio channels to all other endpoints in the conference. The MC may control which endpoint or endpoints are actively multicasting audio. The endpoints receive multicast audio channels and perform a mixing function in order to present a composite audio signal to the user.

All of the endpoints in the conference transmit their video channels to the MP. The MP performs the video switching, mixing, or format conversion functions and outputs the resulting video streams to the endpoints. The MP may produce an exclusive video stream for each endpoint in the conference, or it may multicast a video stream to all participating endpoints, in order to minimize the bandwidth used on the network.

This mode is signalled by the following H.245 capabilities: **centralizedControl**, **distributedAudio**, **centralizedVideo** and **centralizedData**.

6.8.5 Establishment of common mode

The MC shall coordinate a common communications mode between the endpoints in the multipoint conference. The MC may force endpoints into a particular common mode of transmission (as allowed by their capability sets) by sending to the endpoint a receive capability set listing only the desired mode of transmission, or the MC may rely on **multipointModeCommand** and mode preference commands to enforce mode symmetry. The latter approach should be used since it allows the endpoints to know the full range of conference capabilities available that can be requested.

If the MCU has the capability to convert audio and/or video formats, it may not be necessary to force all endpoints into the same communications mode.

6.8.6 Multipoint rate matching

Since the endpoints on each link in a multipoint configuration may attempt to operate at different bit rates, the MC shall send H.245 **flowControlCommand** messages to limit the transmitted bit rates to those which can be sent to receivers.

6.8.7 Multipoint lip synchronization

An MP which is providing audio mixing in either the Centralized or Hybrid multipoint conferences shall modify the time tags of the audio and video streams, taking into account its own time base, in order to maintain audio and video synchronization. Further, when the MP processes the audio and/or video to generate a new stream sourced from the MP, the MP shall generate its own sequence numbers in the audio and video packets.

When mixing audio, the MP should synchronize each of the incoming audio streams to its own timing, mix the audio streams, and then shall generate a new audio stream based on its own timing with its own sequence numbers. If the MP is also switching video, the switched stream shall have its original time-stamp replaced with the MP time base to synchronize it with the mixed audio stream and shall have a new sequence number representing the stream from the MP.

In the case of distributed multipoint conferences, the receiving endpoint may be able to maintain lip synchronization by aligning the selected video stream and its associated audio by using the RTP time tags. Alignment of the other audio streams may not be necessary. If multiple video streams are displayed, the associated audio streams should be aligned.

It may not be possible to guarantee lip synchronization in hybrid multipoint conferences.

6.8.8 Multipoint encryption

In a centralized multipoint configuration, the MP is considered to be a trusted entity. Each port of the MP decrypts the information streams from each of the H.323 endpoints and encrypts the information streams to each endpoint in accordance with 10.1. Operation of an untrusted MCU is for further study.

6.8.9 Cascading multipoint control units

The multipoint control function may be distributed between several MCs. This is called cascading. Cascading allows two or more MCs to communicate with each other in order to control a multipoint conference. Cascading MCs consists of establishing an H.245 Control Channel between the MCs. One MC is defined as the Master MC while the other MCs are defined as Slave MCs.

The procedures for cascading MCs are defined in 8.4.5.

6.9 Models for supplementary services

The ability to support a large variety of supplementary services and features is a requirement for many telephony solutions, regardless of the underlying technologies.

For many such services, an associated requirement is that a high level of interoperability exists between equipment provided by different vendors. This requirement leads to standards-based solutions.

At the same time, equipment suppliers require the ability to provide services that highlight their own products. This can be achieved using proprietary means, but interoperability is compromised. In some cases, such a penalty may be acceptable or desirable, but often this is not so.

The goal, therefore, is to define a standard that is sufficiently flexible that it can support all (or most of) the services that a vendor may wish to supply.

Within the H.323 environment, there are several different methods by which services can be provided: the H.450.x series of Recommendations, ITU-T Rec. H.248 in association with its packages, Annex L and Annex K. Although there is commonality of certain design goals for each of these solutions, the emphasis varies and each is more appropriate for certain circumstances. These solutions represent a spectrum of options for system and feature implementation, from purely peer-peer (functional) control to purely master/slave (stimulus) control, using either first or third party control. Rather than competing, they are complementary, allowing for freedom of choice to the system developer.

The H.450.x series of Recommendations is designed for interoperability of services at a functional level. Its derivation from QSIG ensures interworking with many private networking systems. Services are defined for peer-peer relationships, with feature intelligence typically resident in the endpoint. An H.450-based service must normally be explicitly supported by each affected endpoint in the system. This distribution of service control allows endpoints to be more self-supporting and self-contained and is ideally supported by higher-end endpoints.

The other protocols provide for stimulus level control, where a full understanding of a service is normally only required by a single entity, typically in a master-slave relationship. Such stimulus-based methods use a set of well-defined atomic functions, which, in various combinations, provide any number of services.

Stimulus protocols simplify the introduction of new services. However, different implementations of the same service may differ sufficiently to complicate interoperability, even within the same network type.

Annex L, like H.450, builds on H.323 and all Annex L endpoints are H.323 compliant by definition. It allows standard H.323 procedures to be used for call signalling and media control. Feature intelligence beyond basic call control is implemented in a centralized Feature Server (associated with a Gatekeeper or H.323 endpoint). The protocol allows services to be provided by one or more Feature Servers. Thus Annex L represents a hybrid of peer control and master/slave control models, where intelligence is split between the endpoint and the Feature Server.

Annex K allows third party control of an H.323 call based on a separate control channel (using HTTP [48]) for user interaction. There is no fixed set of capabilities for the user interface, as various types of text formats, images and sounds may be utilized dynamically as registered MIME [49] types. The service provider (the HTTP server) is responsible for the mapping between HTTP events and call control actions (H.450 or other messages) for supplementary services, so the H.323 endpoint is unaware of the HTTP application. The service provider may be associated with the local Gatekeeper, the remote endpoint, or remote Gatekeeper within a call.

H.248 is a generic gateway "device control" protocol, based entirely on a master/slave (stimulus) control model wherein all control intelligence is maintained in a central entity (the Media Gateway Controller, or MGC) and the endpoint (the Media Gateway, or MG) is a slave. H.248 is designed to be independent of the call control protocol and therefore does not require that endpoints be H.323 compliant. H.248 was developed for control of media gateways and it implies a tight relationship between the MGC and the MG, where a user can subscribe to features from only one MGC at a time. H.248 is designed to be easily extensible by the use of packages to define specific support, so that the services that an H.248-based system can support are limited only by the packages supported by the MGC and the MG.

7 Call signalling

Call signalling is the messages and procedures used to establish a call, request changes in bandwidth of the call, get status of the endpoints in the call, and disconnect the call. Call signalling uses messages defined in ITU-T Rec. H.225.0 and the procedures described in clause 8. This clause describes some call signalling concepts.

7.1 Addresses

7.1.1 Network address

Each H.323 entity shall have at least one Network Address. This address uniquely identifies the H.323 entity on the network. Some entities may share a Network Address (i.e., a terminal and a colocated MC). This address is specific to the network environment in which the endpoint is located. Different network environments may have different Network Address formats.

An endpoint may use different Network addresses for different channels within the same call.

7.1.2 TSAP identifier

For each Network Address, each H.323 entity may have several TSAP Identifiers. These TSAP Identifiers allow multiplexing of several channels sharing the same Network Address.

Endpoints have one well-known TSAP Identifier defined: the Call Signalling Channel TSAP Identifier. Gatekeepers have one well-known TSAP Identifier defined: the RAS Channel TSAP Identifier and one well-known multicast address defined: Discovery Multicast Address. These are defined in Appendix IV/H.225.0.

Endpoints and H.323 entities should use dynamic TSAP Identifiers for the H.245 Control Channel, Audio Channels, Video Channels, and Data Channels. The Gatekeeper should use a dynamic TSAP Identifier for Call Signalling Channels. The RAS Channels and Signalling Channels may be redirected to dynamic TSAP Identifiers during the registration procedure.

7.1.3 Alias address

An endpoint may also have one or more alias addresses associated with it. An alias address may represent the endpoint or it may represent conferences that the endpoint is hosting. The alias addresses provide an alternate method of addressing the endpoint. These address include **dialledDigits** or **partyNumber** addresses (including private telephone numbers and public E.164 numbers), H.323 IDs (alphanumeric strings representing names, e-mail like addresses, etc.), and any others defined in ITU-T Rec. H.225.0. Alias addresses shall be unique within a Zone. Gatekeepers, MCs, and MPs shall not have alias addresses.

NOTE – Versions 1, 2 and 3 of ITU-T Recs H.323 and H.225.0 referred to dialled digits in general as E.164 addresses (and **dialledDigits** was **e164**), which they were not. Also, those versions of ITU-T Recs H.323 and H.225.0 referred to E.164 addresses as Public Party Numbers (**e164Number** was **publicPartyNumber**): nowhere was it made clear that public party numbers were E.164 numbers. This terminology change does not affect backward compatibility in any way. Refer to Appendix V for a detailed discussion on the usage of E.164 numbers.

When there is no Gatekeeper in the system, the calling endpoint shall address the called endpoint directly using the Call Signalling Channel Transport Address of the called endpoint. When there is a Gatekeeper in the system, the calling endpoint may address the called endpoint by its Call Signalling Channel Transport Address, or alias address. The Gatekeeper shall translate the latter into a Call Signalling Channel Transport Address.

The called endpoint's **dialledDigits** address may consist of an optional access code followed by a telephone number specific to the service provider's numbering plan. The access code consists of n digits from the set of 0 to 9, *, and #. The number of digits and their meaning is left to the discretion of the manufacturer. One purpose of such an access code might be to request access to a Gateway. The Gatekeeper may alter this address prior to sending it to the destination. The Gatekeeper may also provide a **partyNumber** to use in place of the **dialledDigits**.

The H.323 ID consists of a string of ISO/IEC 10646-1 characters as defined in ITU-T Rec. H.225.0. It may be a user name, conference name, e-mail name, or other identifier.

An endpoint may have more than one alias address (including more than one of the same type) which is translated to the same Transport Address.

7.1.4 H.323 URL scheme

One of the alias types defined by ITU-T Rec. H.323 is the **url-ID**, which is intended to contain standard URL schemes that may be used to reach resources. An H.323 entity may accept any valid URL that it understands, but should support the H.323 URL as defined in this clause.

The H.323 URL is intended to help an entity resolve the address of another H.323 entity. It is composed of two parts: the *user* and the *hostport*. The *user* specifies an alias for the entity, such as a user or a service, without carrying any information about the location of the entity. The *hostport*, on the other hand, is the domain name of the Endpoint, Gatekeeper, or Border Element.

The H.323 URL is defined in ABNF as shown below. Note that it utilizes the Core Rules specified in 6.1 of [52].

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

The *host* is case insensitive.

The *user* is a Unicode [19] string that shall be UTF-8 [57] encoded and then escaped as necessary. Except for characters with a numeric value below 0x80, the *user* is case sensitive. The characters with a numeric value below 0x80 are case insensitive.

The character set and case sensitivity of the *url-parameter* is specified in each parameter definition.

If an endpoint registers with a Gatekeeper and does not provide a *hostport* string, the Gatekeeper may append a *hostport* string to the URL when it returns the endpoint's aliases in an RCF message. The endpoint shall accept the modified alias and use it when sending subsequent requests to the Gatekeeper, including URQ messages to unregister the alias.

7.2 Registration, Admission and Status (RAS) channel

The RAS Channel shall be used to carry messages used in the Gatekeeper discovery and endpoint registration processes which associate an endpoint's alias address with its Call Signalling Channel Transport Address. The RAS Channel shall be an unreliable channel.

Since the RAS messages are transmitted on an unreliable channel, H.225.0 recommends timeouts and retry counts for various messages. An endpoint or Gatekeeper which cannot respond to a request within the specified timeout may use the Request in Progress (RIP) message to indicate that it is still processing the request. An endpoint or Gatekeeper receiving the RIP shall reset its timeout timer and retry counter.

7.2.1 Gatekeeper discovery

Gatekeeper discovery is the process an endpoint uses to determine which Gatekeeper to register with. This may be done manually or automatically. Manual discovery relies on methods outside the scope of this Recommendation to determine which Gatekeeper an endpoint is associated with. The endpoint is configured with the Transport Address of the associated Gatekeeper. For example, it may be entered at endpoint configuration, or it may be entered into an initialization file. In this way, the endpoint knows *a priori* which Gatekeeper it is associated with. The endpoint can now register with that Gatekeeper.

The Automatic method allows the endpoint-Gatekeeper association to change over time. The endpoint may not know who its Gatekeeper is or may need to identify another Gatekeeper due to a failure. This may be done through auto discovery. Auto discovery allows for lower administrative overhead in configuring individual endpoints and additionally allows replacement of an existing Gatekeeper without manually reconfiguring all of the affected endpoints.

The endpoint may multicast (or use other methods as described in Appendix IV/H.225.0) a Gatekeeper Request (GRQ) message, asking "Who is my Gatekeeper?". This is sent to the Gatekeeper's well-known Discovery Multicast Address. One or more Gatekeepers may respond with the Gatekeeper Confirmation (GCF) message indicating "I can be your Gatekeeper" and containing the Transport Address of the Gatekeeper's RAS Channel. If a Gatekeeper does not want the endpoint to register to it, it shall return Gatekeeper Reject (GRJ). See Figure 23. If more than one Gatekeeper responds, the endpoint may choose the Gatekeeper it wants to use. At this point, the endpoint knows which Gatekeeper to register with. The endpoint can now register with that Gatekeeper.

In the event that the endpoint knows the location of the Gatekeeper by some *a priori* means, the endpoint may still choose to unicast the GRQ to the Gatekeeper for the purpose of H.225.0 cryptological exchange.

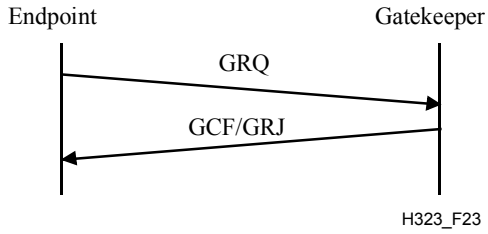


Figure 23/H.323 – Auto discovery

In order to provide redundancy in systems which use a Gatekeeper, the Gatekeeper may indicate alternate Gatekeepers that may be used in the event of a primary Gatekeeper failure. This list of alternate Gatekeepers is provided in the **alternateGatekeeper** field of the GCF and RCF messages.

If no Gatekeeper responds within a timeout, the endpoint may retry the GRQ. An endpoint shall not send a GRQ within 5 s after sending a previous one. If no response is received, the endpoint may use the manual discovery method.

If at any time an endpoint determines it has an invalid registration with its Gatekeeper, it must rediscover its Gatekeeper. The endpoint may assume a registration is invalid if an RRJ is return by a Gatekeeper in response to an RRQ or if no response is received for an RRQ within a timeout.

The GRQ may be repeated periodically (i.e., at endpoint power-up), so the Gatekeeper shall be able to handle multiple requests from the same endpoint.

7.2.2 Endpoint registration

Registration is the process by which an endpoint joins a Zone and informs the Gatekeeper of its Transport Addresses and alias addresses. As part of their configuration process, all endpoints shall register with the Gatekeeper identified through the discovery process. Registration shall occur before any calls are attempted and may occur periodically as necessary (for example, at endpoint power-up).

A Gateway or MCU may register a single Transport Address or multiple Transport Addresses as its call signalling address and may register a single Transport Address or multiple Transport Addresses as its RAS address. The use of multiple Transport Addresses shall indicate a prioritised list of addresses to try when communicating with a given endpoint through either its RAS or Call Signalling Channel.

An endpoint shall send a Registration Request (RRQ) to a Gatekeeper. This is sent to the Gatekeeper's RAS Channel Transport Address. The endpoint has the Network Address of the Gatekeeper from the Gatekeeper discovery process and uses the well-known RAS Channel TSAP Identifier. The Gatekeeper shall respond with either a Registration Confirmation (RCF) or a Registration Reject (RRJ). See Figure 24. An endpoint shall only register with a single Gatekeeper.

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

An endpoint may indicate a backup, redundant, or alternate Transport Addresses using the **alternateEndpoint** structure within the RAS messages. This allows an endpoint to have a secondary network interface or a secondary H.323 endpoint as a backup. The Gatekeeper shall reject ambiguous registrations. The Gatekeeper may reject the registration for other reasons, such as changes in discovery or security issues.

If the endpoint does not include an alias address in the RRQ message, the Gatekeeper may assign one. The Gatekeeper shall return the assigned alias address to the terminal in the RCF message.

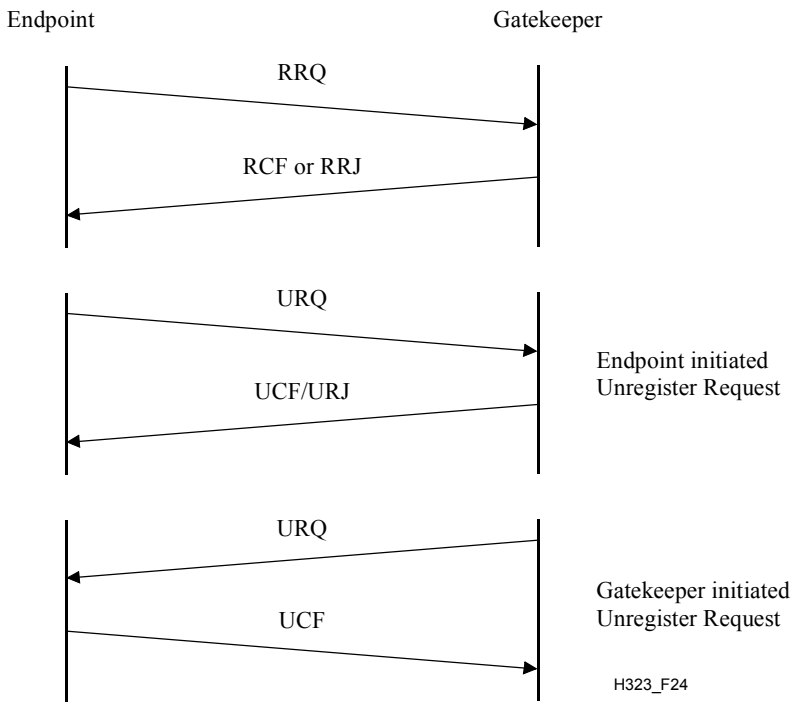


Figure 24/H.323 – Registration

An endpoint may cancel its registration by sending an Unregister Request (URQ) message to the Gatekeeper. This allows an endpoint to change the alias address associated with its Transport Address or vice versa. The Gatekeeper shall respond with either an Unregister Confirmation (UCF) message or an Unregister Reject (URJ) message according to Gatekeeper policy.

If the endpoint sends a URQ message containing a list of alias addresses, the Gatekeeper shall only unregister the listed aliases if it chooses to accept the request. If the endpoint sends a URQ message that does not contain any alias addresses, the Gatekeeper shall unregister all aliases, if any, for the endpoint if it chooses to accept the request.

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

If the Gatekeeper sends a URQ message containing a list of alias addresses, the endpoint shall assume that only those alias addresses are unregistered. A URQ that contains no aliases shall indicate a request to unregister the endpoint.

An endpoint which is not registered with a Gatekeeper is called an unregistered endpoint. This type of endpoint does not request admission permission from a Gatekeeper and so cannot participate in admissions control, bandwidth control, address translation and other functions performed by the Gatekeeper.

7.2.2.1 Use of lightweight RRQ

An endpoint's registration with a Gatekeeper may have a finite life. An endpoint may request a **timeToLive** in the RRQ message to the Gatekeeper. The Gatekeeper may respond with an RCF containing the same **timeToLive**, a longer **timeToLive**, or a shorter **timeToLive**. If the endpoint cannot accommodate a larger **timeToLive** proposed by the Gatekeeper, the endpoint shall use the largest **timeToLive** value that it can support and that is less than the **timeToLive** proposed by the Gatekeeper. After this time, the registration shall be expired. The **timeToLive** is expressed in

seconds. Prior to the expiration time, the endpoint may send an RRQ message having the **keepAlive** bit set. The keep-alive RRQ may include a minimum amount of information as described in ITU-T Rec. H.225.0. The keep-alive RRQ shall reset the time to live timer in the Gatekeeper, allowing the registration to be extended. After the expiration time, the endpoint must re-register with a Gatekeeper using a full RRQ message.

If the Gatekeeper does not include a **timeToLive** value in the RCF, the registered endpoint shall consider that the Gatekeeper is not supporting the keep-alive mechanism. Endpoints shall not send RRQs with the **keepAlive** field set to Gatekeepers which have indicated that they are not supporting the keep-alive mechanism. A Gatekeeper should not assume that an endpoint supports the keep-alive mechanism if the endpoint does not provide a **timeToLive** value in the RRQ.

Gatekeepers should not treat an RRQ with the **keepAlive** field set as a full registration (i.e., for updating or initializing its translation tables).

Endpoints should consider messaging and processing delays when determining when their registration will expire (i.e., the duration of their own time-to-live timer) at the Gatekeeper.

Expiration of the time-to-live timer in the Gatekeeper results in the expiration of the registration of the endpoint. A Gatekeeper may send a URQ to the endpoint as a notification of such expiration. This allows for loss of synchronization between the time-to-live timers of the Gatekeeper and the endpoint. It also indicates a need for re-registration to endpoints which do not support the keep-alive mechanism.

An endpoint which sends a lightweight RRQ to its Gatekeeper after the time-to-live timer has expired in the Gatekeeper will receive an RRJ response with **rejectReason** of either **fullRegistrationRequired** or **discoveryRequired**, depending on Gatekeeper requirements.

An endpoint which sends an ARQ to its Gatekeeper after the time-to-live timer has expired in the Gatekeeper will receive an ARJ with **rejectReason** of either **callerNotRegistered** or **calledPartyNotRegistered**. An endpoint which initiates a new call through its Gatekeeper after expiration of the Gatekeeper's time-to-live timer will receive a Release Complete message with a reason of **callerNotRegistered** or **calledPartyNotRegistered**.

Disposition of existing calls upon expiration of the time-to-live timer is implementation dependent.

7.2.2.2 Use of additive registrations

Support for additive registrations is optional in both the Gatekeeper and the endpoint. A Gatekeeper that supports additive registrations shall indicate support by including the **supportsAdditiveRegistration** field in the RCF message and shall comply with the procedures set forth in this clause. Additionally, an endpoint shall not use the additive registration procedure described in this clause if the **supportsAdditiveRegistration** field of the RCF is missing.

If the Gatekeeper receives an RRQ with the **additiveRegistration** field included, it shall treat the RRQ as an addition of information to an existing registration for the endpoint specified in the **endpointIdentifier** field. Upon receiving an additive RRQ, the Gatekeeper shall add the alias (or list of aliases) from the **terminalAlias** and **terminalAliasPattern** fields to the existing translation table entries for the endpoint. Also, the Gatekeeper shall add the supported prefixes from the **supportedPrefixes** field of the **terminalType** field to the existing translation table entries for the endpoint. Any previously registered alias addresses or supported prefixes for the endpoint shall remain registered. The Gatekeeper shall replace the endpoint's Call Signalling Addresses and RAS addresses with the values specified in the **callSignalAddress** and **rasAddress** fields, if any are present, and shall replace the endpoint's alternate endpoints with values specified in the **alternateEndpoints** field, if present. The **keepAlive** shall be FALSE if the **additiveRegistration** field is included in the RRQ. However, the receipt of an additive RRQ shall cause the Gatekeeper to restart the endpoint's time to live counter if one is currently running.

An endpoint that sends an additive RRQ to its Gatekeeper when the endpoint is not registered will receive an RRJ response with **rejectReason** of either **fullRegistrationRequired** or **discoveryRequired**, depending on Gatekeeper requirements.

NOTE – As the additive RRQ is not a full registration, the Gatekeeper may ignore fields in the additive RRQ not specifically referenced in this clause.

7.2.3 Endpoint location

An endpoint or Gatekeeper which has an alias address for an endpoint and would like to determine its contact information may issue a Location Request (LRQ) message. This message may be sent to a specific Gatekeeper's RAS Channel TSAP Identifier or may be multicast like the GRQ message to the Gatekeeper's well-known Discovery Multicast Address. The Gatekeeper with which the requested endpoint is registered shall respond with the Location Confirmation (LCF) message containing the contact information of the endpoint or the endpoint's Gatekeeper. Contact information shall include the Call Signalling Channel and RAS Channel addresses to be used to reach the endpoint and optionally additional destination information which can provide dialling information and extension information concerning the requested endpoint.

All Gatekeepers with which the requested endpoint is not registered shall return Location Reject (LRJ) if they received the LRQ on the RAS Channel. Any Gatekeeper with which the requested endpoint is not registered shall not respond to the LRQ, if it received the LRQ on the Discovery Multicast address.

An endpoint or Gatekeeper may include one or more **dialledDigits** or **partyNumber** extensions to which it wishes to connect in the **destinationInfo** field of the LRQ to attempt to locate an available Gateway outside of its zone. A Gatekeeper which receives an LRQ requesting an available Gateway is not obligated to make its Gateways available to such a request.

A Gatekeeper may be aware of the alias address and connection information of endpoints on the SCN. This Gatekeeper could respond to an LRQ requesting information on the SCN endpoint with the connection information necessary to reach that endpoint. This would include the information necessary to address the Gateway as well as the SCN endpoint. Note that the SCN endpoint is not registered with the Gatekeeper in the sense that it exchanges RRQ/RCF messages with the Gatekeeper. The method by which a Gatekeeper becomes aware of the SCN endpoint information is outside the scope of this Recommendation.

7.2.4 Admissions, bandwidth change, status and disengage

The RAS Channel is also used for the transmission of Admissions, Bandwidth Change, Status and Disengage messages. These messages take place between an endpoint and a Gatekeeper and are used to provide admissions control and bandwidth management functions. The detailed use of these messages is described in clause 8.

The Admission Request (ARQ) message specifies the requested Call Bandwidth. This is an upper limit on the aggregate bit rate for all transmitted and received, audio and video channels excluding any RTP headers, RTP payload headers, network headers, and other overhead. Data and control channels are not included in this limit. The Gatekeeper may reduce the requested Call Bandwidth in the Admission Confirm (ACF) message. An endpoint shall assure that the aggregate bit rate, averaged over one second, for all transmitted and received, audio and video channels is at or below the Call Bandwidth. An endpoint or the Gatekeeper may attempt to modify the Call Bandwidth during a call using the Bandwidth Change Request (BRQ) message.

The Admission Confirm Sequence message allows the Gatekeeper to provide a single reply to an ARQ containing alternate routing information, different source information, different tokens, etc. When an endpoint receives an Admission Confirm Sequence message containing more than one ACF inside, it shall process the first ACF in the sequence by attempting to establish the call as described in this Recommendation. In the event that the endpoint is unable to establish the call due

to some unexpected failure, the endpoint may then select the next ACF message in the sequence and re-attempt the call establishment without first consulting the Gatekeeper. Without limiting the definition, "unexpected failures" may include busy circuits; transport routing problems (e.g., "no route to host"); or exhausted gateway resources. It is the endpoint's decision as to whether it wishes to make call attempts to alternate routes in the face of a routing failure.

Endpoints that choose to support the Admission Confirm Sequence message shall indicate this capability by setting the **acfSequences** field in the RRQ message to TRUE. The Gatekeeper shall consider absence of this field as a FALSE value. Gatekeepers shall not send the Admission Confirm Sequence message to an endpoint that has not indicated support for this message in the RRQ. An endpoint may change the value of the **acfSequences** field in subsequent RRQ messages. In the event that the endpoint changes this value from TRUE to FALSE, the endpoint shall be prepared to receive Admission Confirm Sequence messages that might be in transit as a result of having previously advertised support of Admission Confirm Sequence messages.

As it is that the Admission Confirm Sequence is merely a means of providing alternate routing information that could not be provided in an Admission Confirm message, this Recommendation makes no further distinction elsewhere as to the semantic difference between the Admission Confirm message and the Admission Confirm Sequence message. Throughout this Recommendation, "Admission Confirm" or "ACF" refers to either a single Admission Confirm message or an Admission Confirm Sequence message.

7.2.5 Access tokens

An Access Token is a string passed in some RAS messages and the Setup message. The Access Tokens have two uses. First, they can provide privacy by shielding an endpoint's Transport Address and Alias Address information from a calling party. A user may give out only the Access Token for a calling party to use in reaching the endpoint. The Gatekeeper will know the endpoint related to the Access Token from the registration process, so that calls using the Access Token can be routed through the Gatekeeper to the called endpoint. The use of the access token only applies to the Gatekeeper routed call model when attempting to hide the Transport Address from the endpoint.

The second use of the Access Token is in ensuring that calls are routed properly through H.323 entities. An Access Token returned by a Gatekeeper shall be used in any subsequent setup messages sent by the endpoint. This Access Token may be used by a Gateway to assure that the endpoint has permission to use the Gateway resources, or it may be used by a called endpoint to assure that the calling endpoint can signal it directly.

The Access Token may also be distributed by out-of-band methods to assure proper access to Gateways and endpoints in systems that do not have Gatekeepers.

7.2.6 Alternate gatekeeper procedures

For the purposes of ensuring system availability, redundancy, and scalability, the Gatekeeper may provide the RAS signalling function by utilizing multiple physical or logical devices, referred to as Alternate Gatekeepers. If the endpoint supports the Alternate Gatekeeper procedures defined in this clause, it should include the **supportsAltGK** field in the GRQ and RRQ messages.

When an endpoint initiates communication with the Gatekeeper, it may be provided with a list of Alternate Gatekeepers via the GCF message. If the Gatekeeper does not respond to the subsequent RRQ, the endpoint shall attempt to register with the Gatekeeper using the list of Alternate Gatekeepers provided in the GCF. If no Alternate Gatekeeper responds, the endpoint shall reinitiate the Gatekeeper discovery process.

If the endpoint receives a GRJ message containing Alternate Gatekeeper information and does not receive a GCF message, the endpoint shall send GRQ messages to one or more Alternate Gatekeepers in the list of Alternate Gatekeepers received in the GRJ. If multiple GRJ messages are received, the endpoint may select any one GRJ message from which to extract Alternate Gatekeeper

information. If no Alternate Gatekeeper sends a GCF message, the endpoint may attempt to use any new Alternate Gatekeeper lists received for the purpose of Gatekeeper discovery or it may reinitiate the Gatekeeper discovery process.

If the endpoint has not yet registered with the Gatekeeper or has reinitiated the Gatekeeper discovery process, it shall ignore the **needToRegister** field in the Alternate Gatekeeper list and assume that the value is TRUE.

If the endpoint is registered with the Gatekeeper and the Gatekeeper becomes unresponsive, the endpoint shall attempt to communicate with an Alternate Gatekeeper.

The Gatekeeper may explicitly redirect an endpoint to an Alternate Gatekeeper by returning a RAS rejection message with a list of Alternate Gatekeepers. If the **altGKisPermanent** field is set to FALSE in such a redirection, the redirection is considered temporary, as it only applies to a single RAS message.

A Gatekeeper may send a URQ to an endpoint with a list of Alternate Gatekeepers, in which case the endpoint shall respond with a UCF and attempt to communicate with an Alternate Gatekeeper. An endpoint shall not include a list of Alternate Gatekeepers in any URQ message that it sends.

An endpoint shall keep only one list of Alternate Gatekeepers. That list shall be taken from the most recently received list of Alternate Gatekeepers received in any RAS message, with one exception: if the endpoint is temporarily redirected to an Alternate Gatekeeper and the Alternate Gatekeeper returns a rejection message with a list of Alternate Gatekeepers (even if the list is empty), the endpoint shall interpret the rejection as a redirection. The endpoint may ignore the list of Alternate Gatekeepers provided in such a redirection and continue using the list of Alternate Gatekeepers received in the original rejection message.

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

The endpoint shall use the **priority** field to indicate the order in which to communicate with Alternate Gatekeepers. If multiple Alternate Gatekeepers are specified to have the same **priority**, the endpoint may order the Alternate Gatekeepers with the same **priority** value as it chooses.

When an endpoint is redirected to a temporary Alternate Gatekeeper, it shall ignore the **needToRegister** field and assume the value is FALSE and retransmit only the redirected RAS message to a temporary Alternate Gatekeeper. All other RAS messages shall continue to be sent to the Gatekeeper as usual. Note that this does not preclude the Gatekeeper from temporarily redirecting an endpoint to an Alternate Gatekeeper by returning an RRJ to either an RRQ or a lightweight RRQ.

If distinct RAS requests are redirected to temporary Alternate Gatekeepers, each distinct message shall be sent to one and only one temporary Alternate Gatekeeper at a time, although different RAS messages may be sent to different temporary Alternate Gatekeepers simultaneously. If the endpoint determines that a temporary Alternate Gatekeeper is unresponsive, it shall attempt to retransmit the RAS request to another Alternate Gatekeeper. If all Alternate Gatekeepers are unresponsive to a RAS request, the endpoint shall assume that the RAS request is rejected. If the request was an RRQ, the endpoint shall reinitiate the Gatekeeper discovery process.

If the Gatekeeper becomes unresponsive or if the Gatekeeper redirects the endpoint by returning a list of Alternate Gatekeepers with the **altGKisPermanent** field set to TRUE, the endpoint shall attempt to communicate with an Alternate Gatekeeper. The endpoint shall attempt communication with only one Alternate Gatekeeper. Only after the endpoint determines that an Alternate Gatekeeper is unresponsive shall it attempt to communicate with the next Alternate Gatekeeper. If all Alternate Gatekeepers are unresponsive, the endpoint shall reinitiate the Gatekeeper discovery process. If registration is required with an Alternate Gatekeeper, the endpoint shall first attempt to

send an RRQ to the Alternate Gatekeeper, rather than a GRQ. Only if the Gatekeeper returns an RRJ with the reason **discoveryRequired** shall the endpoint send a GRQ to the Alternate Gatekeeper. When permanently transitioning to an Alternate Gatekeeper, the endpoint shall send all further RAS messages to the Alternate Gatekeeper, including outstanding RAS requests that timeout. The endpoint should reset the retry counters for any outstanding RAS messages before transmitting them to the Alternate Gatekeeper for the first time.

If an Alternate Gatekeeper to which an endpoint is redirected returns a rejection message without a list of Alternate Gatekeepers, the endpoint shall accept the message as a rejection to the original request. If the rejection was to an RRQ, the endpoint shall reinitiate the Gatekeeper discovery process. If the Alternate Gatekeeper redirects the endpoint by returning a rejection message with a list of Alternate Gatekeepers, the endpoint shall attempt to send the request to another Alternate Gatekeeper. If all Alternate Gatekeepers redirect the endpoint, the endpoint shall ultimately assume that the request is rejected.

An endpoint shall not send a URQ message when transitioning between Alternate Gatekeepers, even if the **needToRegister** field is TRUE, except in the case where the Gatekeeper sends a URQ with a list of Alternate Gatekeepers.

If an endpoint is redirected to an Alternate Gatekeeper that is specified to be permanent (i.e., the **altGKisPermanent** field is TRUE) or was forced to begin communicating with an Alternate Gatekeeper after its Gatekeeper became unresponsive, it shall assume that the Alternate Gatekeeper is prepared to accept requests relating to existing calls. It shall send all subsequent BRQ, DRQ, and IRR messages relating to existing calls to the Alternate Gatekeeper. Likewise, the Alternate Gatekeeper shall be prepared to handle such messages.

If an endpoint begins communicating with an Alternate Gatekeeper with which registration was not required, including temporary Alternate Gatekeepers, the **gatekeeperIdentifier** field of the URQ, ARQ, BRQ, LRQ, and DRQ messages shall contain the **gatekeeperIdentifier** of the Alternate Gatekeeper from the Alternate Gatekeeper list. This field may not be present when registration is required.

7.2.7 Usage information reporting

An endpoint may have the ability to collect and report call usage information, which may be useful for accounting or billing purposes. A Gatekeeper may request that an endpoint report this information. This feature is intended to interwork with the usage information reporting features of systems that implement Annex G/H.225.0.

Note that this feature is intended for scenarios in which the endpoint from which the usage information is requested is trusted, such as when a gateway and Gatekeeper are administered by the same service provider. That is, it is assumed that the endpoint will accurately report its usage information.

7.2.7.1 Advertising usage information reporting capabilities

An endpoint may advertise to a Gatekeeper its ability to collect and report usage information. It specifies these capabilities in the **usageReportingCapability** field of the RRQ message. If the endpoint has reported its capabilities and these capabilities subsequently change, the endpoint shall send another RRQ specifying its capabilities. Absence of a **usageReportingCapability** field in an RRQ indicates that the endpoint cannot report usage information.

7.2.7.2 Requesting usage information reports

A Gatekeeper may request usage information from an endpoint via the RCF, ACF and IRQ messages. A Gatekeeper should assume that an endpoint that has not advertised the ability to report a particular type of usage information will not report that information, and it should not request that information from the endpoint.

A Gatekeeper may request usage information via the **usageSpec** field of the RCF message. This request is referred to as the "default" **usageSpec**. By including this field, the Gatekeeper is requesting that the endpoint collect and report the specified usage information for all new calls. This request does not apply to calls that are already in progress.

Once a Gatekeeper has delivered a default **usageSpec** via the RCF, it assumes that this request remains in effect until it delivers another default **usageSpec**. If the Gatekeeper does not wish to change a previously delivered default **usageSpec**, it may indicate this by not including the **usageSpec** in when sending an RCF message. In order to change a previously delivered default request for usage information, a Gatekeeper shall send a new **usageSpec** in its next RCF message. In order to request that an endpoint stop reporting usage information, a Gatekeeper shall send a **usageSpec** with no options selected in either the **when** or **required** fields.

A Gatekeeper may request usage information for a particular call via the **usageSpec** field of the ACF message for that call. This request is referred to as the "per-call" **usageSpec**. If provided, this request overrides, for that call, any default usage specification that the Gatekeeper may have provided in an RCF message.

A Gatekeeper may also request usage information for a particular call via the **usageInfoRequested** field of an IRQ message. The response to this request should immediately follow in an IRR message. This request does not affect either the default usage specification sent via the RCF or the per-call usage specification sent via the ACF.

A Gatekeeper that wishes an endpoint to report usage information periodically in unsolicited IRR messages shall indicate this request by selecting the **inIrr** option of the **when** field of the **usageSpec**. It shall also specify either the **irrFrequencyInCall** in the **preGrantedARQ** field of the RCF message, or the **irrFrequency** in the ACF message, as appropriate for a particular call.

A Gatekeeper that requests that usage information be reported at the start of a call or in unsolicited IRR messages (i.e., that selects the **start** or **inIrr** options in the **when** field of the **usageSpec**) should acknowledge IRR messages in order to ensure that the requested usage information is reliably delivered. To indicate that it will acknowledge IRR messages, the Gatekeeper sets the **willRespondToIRR** field of the RCF or ACF message to TRUE.

7.2.7.3 Sending usage information reports

An endpoint may report usage information to a Gatekeeper via the BRQ, IRR and, DRQ and DCF messages. An endpoint may send usage information to a Gatekeeper that has not requested that information. If an endpoint advertises the ability to collect and report a particular type of usage information, and a Gatekeeper requests that information, then the endpoint shall report the requested information. An endpoint shall ignore requests for usage information that are erroneous (such as a request to provide the call end time at the start of a call). An endpoint may ignore a request for usage information that does not fall within the endpoint's advertised reporting capabilities.

If a Gatekeeper sends an endpoint a default **usageSpec** in an RCF message, the endpoint shall set the usage information reporting parameters for all new calls based on this template, unless the Gatekeeper supplies a per-call **usageSpec** for a particular call in an ACF message. If provided, the per-call **usageSpec** overrides the default **usageSpec** for that call. An endpoint may apply an updated default **usageSpec** to existing calls for which no per-call **usageSpec** was provided.

An endpoint shall interpret a **usageSpec** with no options selected in either the **when** or the **required** fields as a request not to report usage information.

When reporting usage information via an IRR message, and the Gatekeeper has indicated via the **willRespondToIRR** field of either the RCF or ACF that it will acknowledge IRRs, an endpoint shall set the **needResponse** field to TRUE and retransmit the information if an acknowledgement is not received. This rule shall apply whether the IRR is solicited or unsolicited.

If the Gatekeeper has requested that usage information be reported at the start of the call (i.e., it selected **start** in the **when** field of the **usageSpec**), and the requested information is within the endpoint's advertised capabilities to report, then the endpoint shall report the requested information immediately after the start of the call. If the endpoint sends a BRQ at this point in time, then it may include the requested usage information in the **usageInformation** field of the BRQ message. Otherwise, the endpoint shall send an unsolicited IRR message with the requested usage information in the per-call **usageInformation** field.

If the Gatekeeper has requested that usage information be reported at the end of the call (i.e., it selected **end** in the **when** field of the **usageSpec**), and the requested information is within the endpoint's advertised capabilities to report, then the endpoint shall report the requested information immediately after the end of the call in the DRQ message (or in the DCF if the call is terminated by the Gatekeeper).

If the Gatekeeper has requested that usage information be reported in unsolicited IRR messages (i.e., it selected **inIrr** in the **when** field of the **usageSpec**), and the requested information is within the endpoint's advertised capabilities to report, then the endpoint shall report the requested information in every unsolicited IRR that it sends.

The endpoint shall apply neither the default nor the per-call **usageSpec** when sending solicited IRR messages (i.e., responses to IRQs). If the Gatekeeper requests usage information via the **usageInfoRequested** field of the IRQ, and it is within the endpoint's advertised capabilities to report this information, then the endpoint shall report the requested information in the per-call **usageInformation** field of the IRR. If the Gatekeeper does not request usage information in the IRR, the endpoint should not include a **usageInformation** field in the response.

7.2.8 Call credit-related capabilities

By utilizing the optional credit-related capabilities, an endpoint can receive a user's credit or debit information from the Gatekeeper before and after the user establishes a call. In turn, the endpoint may relay this information to the end user via an announcement. The endpoint also has the option to limit the user's call duration to an amount of time specified by the Gatekeeper. For instance, the endpoint may disengage the call when the time or money on the user's account is exhausted.

In addition, the Gatekeeper may send balance-related announcements to the endpoint and may indicate a call duration limit to the endpoint.

7.2.8.1 Endpoint advertisement of credit-related capabilities

The endpoint indicates its support for the call credit features via the RRQ. The ability to play or display announcements regarding a caller's balance may be advertised via a new **supportedH248Packages** field. The **supportedH248Packages** field consists of an optional list of **H248PackagesDescriptors** in binary format.

To send a text announcement, the endpoint and Gatekeeper may use the "Display" package (**PackageID** dis, 0x0014), defined in Annex G/H.248. Annex G/H.248 includes facilities to control the location of the text on a terminal display and other functions.

To send the index of either a fixed or a parameterized voice announcement that is locally stored at the endpoint, the endpoint and Gatekeeper may use the "Generic Announcement" package (**PackageID** an, 0x001D) defined in Annex K/H.248.

As an alternative to the use of H.248 packages, the endpoint may indicate via H.225.0 call signalling that it is capable of including the user's balance in a text announcement that it constructs itself. This capability may be indicated via the **canDisplayAmountString** flag.

The endpoint may indicate via the **canEnforceDurationLimit** flag whether it can perform its own call timing.

7.2.8.2 Balance information sent by the gatekeeper to the endpoint

The Gatekeeper may send announcements (which could be either voice or text) to the endpoint via an H.248 "signal" in the **ServiceControlDescriptor** structure in the ACF, SCI, and/or DRQ messages. Alternatively, the Gatekeeper may send a text string to the endpoint in the **amountString** field that indicates the account balance, for example "$10.50", in the appropriate currency. In this case, the endpoint is responsible for embedding the amount string in an announcement (for example, "Current debit card balance: $10.50") that is appropriate for that particular endpoint. Note that ISO 4217 defines standard abbreviations for currency types, such as "USD" for United States dollars. The **amountString** field shall be encoded in Unicode.

A **billingMode** field is also added to allow the Gatekeeper to indicate the billing mode for the call. A mode of **debit** indicates that the call will result in charges against the amount of money available in a user's account. A mode of **credit** indicates that the call will result in charges to be paid by the user at a later time. An endpoint may use this information, for example, to determine the type of announcement to play or display.

The **callDurationLimit** field of the **CallCreditServiceControl** structure indicates the remaining amount of time allowed for a particular call. The **enforceCallDurationLimit** flag indicates whether timing enforcement shall be performed by the endpoint. The **callStartingPoint** field indicates the point in the call that timing shall begin if call duration enforcement is provided by the endpoint.

If the endpoint has advertised that it is capable of enforcing the time limit and the Gatekeeper requests that the endpoint enforce the limit, then the endpoint shall disengage the call when the time limit expires. Timing of the call duration shall begin upon transmission or reception of the Connect message or the Alerting message as indicated by the **callStartingPoint field**.

7.2.9 Alternate transport addresses

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

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

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

7.3 Call signalling channel

The Call Signalling Channel shall be used to carry H.225.0 call control messages. The Call Signalling channel shall be a reliable channel.

In networks that do not contain a Gatekeeper, call signalling messages are passed directly between the calling and called endpoints using the Call Signalling Transport Addresses. In these networks, it is assumed that the calling endpoint knows the Call Signalling Transport Address of the called endpoint and thus can communicate directly.

In networks that do contain a Gatekeeper, the initial admission message exchange takes place between the calling endpoint and the Gatekeeper using the Gatekeeper's RAS Channel Transport Address. Within the initial admissions message exchange, the Gatekeeper indicates in the ACF message whether to send the call signalling directly to the other endpoint or to route it through the Gatekeeper. The call signalling messages are sent to either the endpoint's Call Signalling Transport Address or the Gatekeeper's Call Signalling Transport Address.

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

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.

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

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

ITU-T Rec. H.225.0 specifies the mandatory Q.931 messages that are used for call signalling in this Recommendation. Clause 8 specifies the procedures for using them.

7.3.1 Call signalling channel routing

Call signalling messages may be passed in two ways. The first method is Gatekeeper routed call signalling (see Figure 25). In this method, call signalling messages are routed through the Gatekeeper between the endpoints. The second method is Direct Endpoint Call Signalling (see Figure 26). In this method, the call signalling messages are passed directly between the endpoints. The choice of which methods is used is made by the Gatekeeper.

Both methods use the same kinds of connections for the same purposes and the same messages. Admission messages are exchanged on RAS channels with the Gatekeeper, followed by an exchange of call signalling messages on a Call Signalling channel. This is then followed by the establishment of the H.245 Control Channel. The actions of the Gatekeeper in response to the

admission messages determine which call model is used; this is not under the control of the endpoint, although the endpoint can specify a preference.

The symmetrical signalling method of Annex D/Q.931 shall be used for all mandatory call signalling procedures. This does not address the role that a Gateway might play on the SCN side using Q.931 or other call signalling protocols.

The Gatekeeper Clouds in Figures 25 through 28 contain one or more Gatekeepers which may or may not communicate with each other. The endpoints may be connected to the same Gatekeeper or to different Gatekeepers.

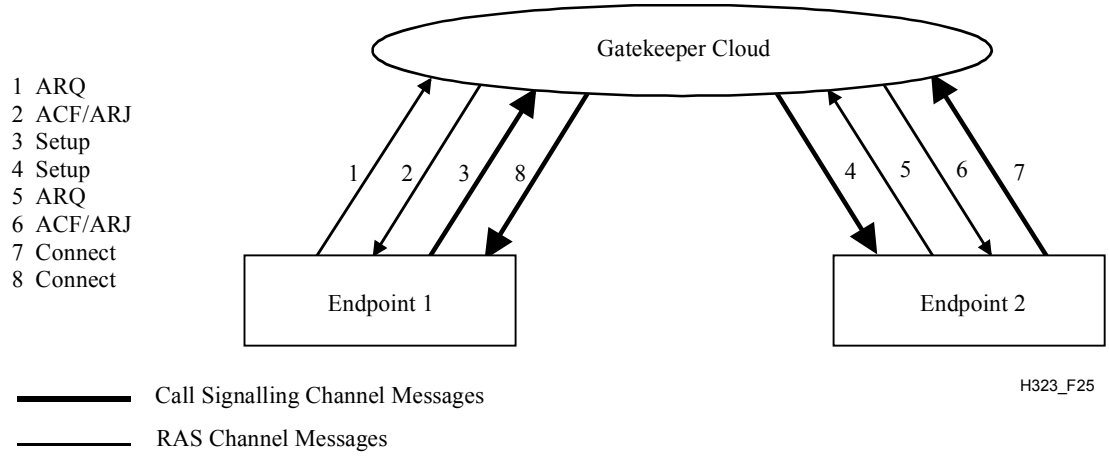


Figure 25/H.323 – Gatekeeper routed call signalling

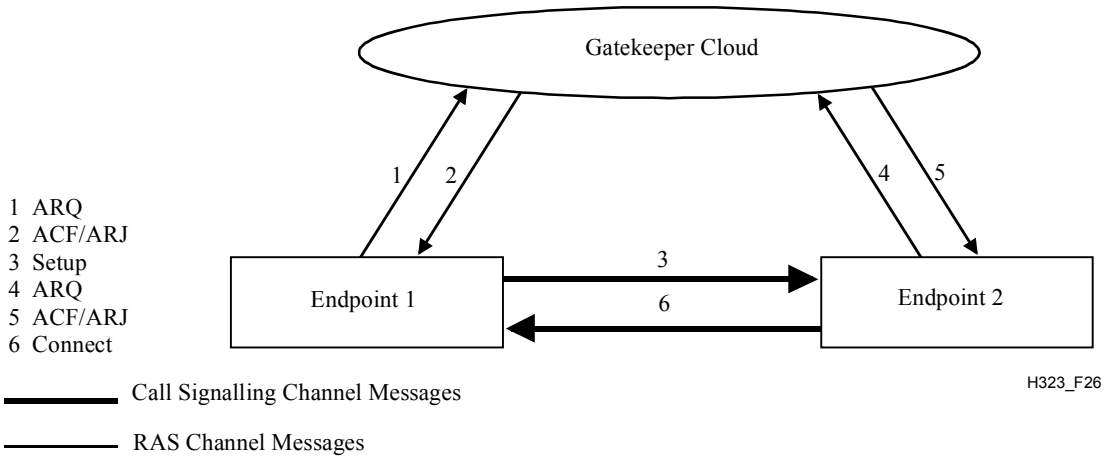


Figure 26/H.323 – Direct endpoint call signalling

7.3.2 Control channel routing

When Gatekeeper routed call signalling is used, there are two methods to route the H.245 Control Channel. In the first method, the H.245 Control Channel is established directly between the endpoints. See Figure 27. This method is for further study. In the second method, the H.245 Control Channel is routed between the endpoints through the Gatekeeper. See Figure 28. This method allows the Gatekeeper to redirect the H.245 Control Channel to an MC when an ad hoc multipoint conference switches from a point-to-point conference to a multipoint conference. The Gatekeeper makes this choice. When Direct Endpoint call signalling is used, the H.245 Control Channel can only be connected directly between the endpoints.

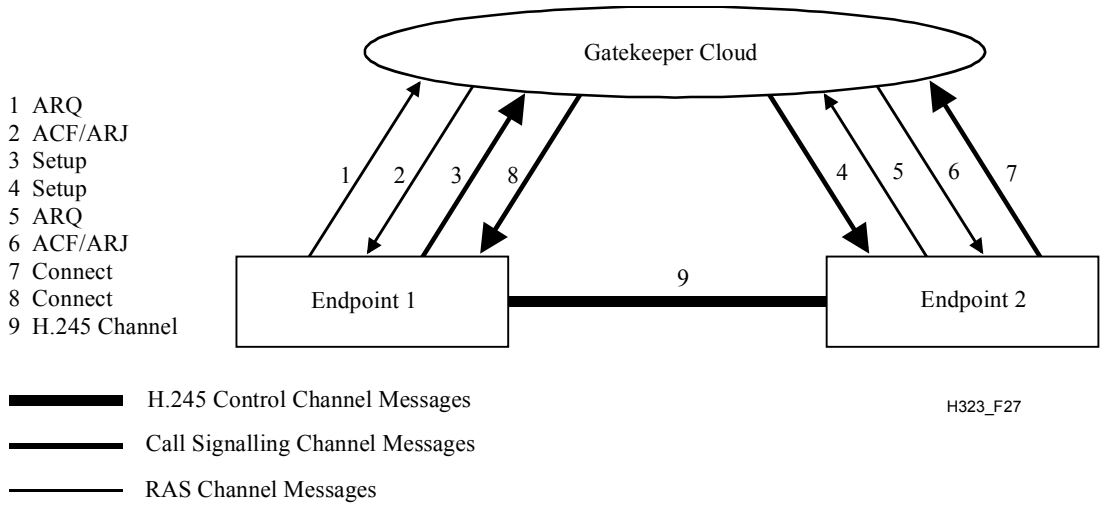


Figure 27/H.323 – Direct H.245 control channel connection between endpoints

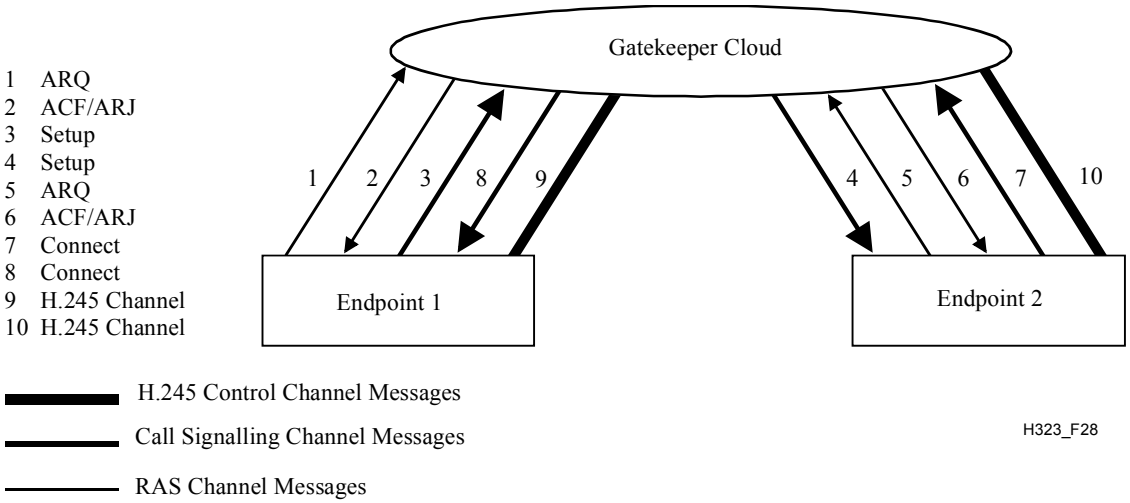


Figure 28/H.323 – Gatekeeper routed H.245 control

7.3.3 Call signalling and control protocol revisions

When a call is routed through a Gatekeeper, Gatekeepers shall use the following rules to determine the H.225.0 or H.245 version number to be indicated in messages originated by an endpoint and routed or forwarded by the Gatekeeper:

a) If the originating endpoint's H.225.0 or H.245 version number is less than or equal to the Gatekeeper's version number, and the Gatekeeper chooses to proxy the functions of an equal or later version number on behalf of the originating endpoint, the routed messages shall reflect the version number of the gatekeeper. Otherwise they shall reflect the version number of the originating endpoint.

b) If the originating endpoint's version number is greater than Gatekeeper's, the routed messages shall reflect the version number of the Gatekeeper.

In all cases, the Gatekeeper may use a single ASN.1 encoding specified by the most recent H.225.0 or H.245 version understood by the Gatekeeper according to these rules.

Since some features in the H.323, such as third party pause and re-routing, require that the signalling entities know exactly what version of the protocol is being used by the other entities in a call and because the **protocolIdentifier** may change after receiving the first call signalling message

and at other times during the call, such as when a call is re-routed to a different entity, entities that rely on version-specific features should determine the version of the other entities in a call by examining the **protocolIdentifier** in the Setup and Connect message at the very least. During a call, a call may be re-routed to a different entity that uses a different version of the protocol. In such a case, entities that rely on version-specific features should again determine the version of the entity to which the call may have been switched. If H.245 signalling is tunnelled, the endpoint may use the call signalling message containing the tunnelled non-empty terminal capability set message in order to determine the version of the remote endpoint. If a separate H.245 Channel is used, an entity may send a Status Inquiry message and determine the protocol version by examining the **protocolIdentifier** in the resulting Status message. In either case, the version of H.245 used by the other entity is signalled in the non-empty capability set message.

It should be noted that H.323 entities prior to version 4 may not provide the **protocolIdentifier** in the Status message, so H.323 entities shall assume that the absence of the **protocolIdentifier** indicates only that the entity is older than version 4.

NOTE – A Gatekeeper may signal its own protocol version when replying to a Setup message (e.g., to send a Call Proceeding message prior to establishing communication with the called party) or when initiating an outbound connection independent of an existing call. Therefore, it is important that an endpoint not rely on the initial message(s) to determine the protocol revision of the remote endpoint.

7.4 Call reference value

All call signalling and RAS messages contain a Call Reference Value (CRV). Refer to ITU-T Rec. H.225.0. There is one CRV for the Call Signalling Channel and an independent CRV for the RAS channel. One CRV is used to associate the call signalling messages. This CRV shall be used in all call signalling messages between two entities (endpoint to Gatekeeper, endpoint-to-endpoint, etc.) related to the same call. A second CRV is used to associate the RAS messages. This CRV shall be used in all RAS messages between two entities related to the same call. New CRVs shall be used for new calls. A second call from an endpoint to invite another endpoint into the same conference shall use new CRVs. The CRV is not the same as the Call ID or the Conference ID (CID). The CRV associates call signalling or RAS messages between two entities within the same call, the Call ID associates all messages between all entities within the same call, and the CID associates all messages between all entities within all calls in the same conference.

The Global Call Reference, as shown Figure 4-5/Q.931 and having the numeric value 0, is used to refer to all calls on the Call Signalling Channel or the RAS channel. When initiating or accepting calls, H.323 entities shall select a CRV value other than Global Call Reference value; the Global Call Reference is reserved for messages which do not pertain to a particular call.

When placing a new call, the calling endpoint shall select a new CRV for the call. The calling endpoint shall use the same CRV on both the RAS channel and the H.225.0 Call Signalling Channel. The called endpoint, however, shall not use the CRV value received in the Setup when communicating on its RAS channel. Instead, the called endpoint shall select a new CRV for use the RAS channel that is unique on that channel without regard to the CRV received in the Setup, though they may happen to be numerically equivalent as a matter of course.

7.5 Call ID

The Call ID is a globally unique non-zero value created by the calling endpoint and passed in various H.225.0 messages. The Call ID identifies the call with which the message is associated. It is used to associate all RAS and Call Signalling messages related to the same call. Unlike CRV, the Call ID does not change within a call. All messages from the calling endpoint to its Gatekeeper, the calling endpoint to the called endpoint, and the called endpoint to its Gatekeeper related to the same call shall contain the same Call ID. The Call ID is encoded as described in ITU-T Rec. H.225.0. In

reference to Figures 29 through 39 in clause 8, all messages within a figure shall have the same Call ID.

When a Version 1 endpoint calls a Version 2 endpoint, it is the responsibility of the Version 2 endpoint to generate a Call ID prior to sending ARQ to its Gatekeeper.

7.6 Conference ID and conference goal

The Conference ID (CID) is a unique non-zero value created by the calling endpoint and passed in various H.225.0 messages. The CID identifies the conference with which the message is associated. Therefore, messages from all endpoints within the same conference will have the same CID. The CID is encoded as specified in ITU-T Rec. H.225.0.

The **conferenceGoal** indicates the intention of the call. Choices are: **create** – to create a new conference, **join** – to join an existing conference, **invite** – to invite a new endpoint into an existing conference, **capability-negotiation** – negotiate capabilities for a later H.332 conference, and **callIndependentSupplementaryService** – transport of supplementary services APDUs.

7.7 Endpoint call capacity

Call capacity indicates an endpoint's acceptance capacity for each type of call the endpoint supports (e.g., voice, T.120 data, H.320, etc.). While any endpoint type may report call capacity through various H.225.0 messages in order to assist a Gatekeeper in routing calls, call capacity information should be reported by Gateways to assist the Gatekeeper with load balancing across Gateways and to help reduce the number of failed call attempts.

The endpoint's maximum and current capacity may be indicated at registration. In addition, the current capacity may also be indicated on a per-call basis. Representing this dynamic capacity requires consideration of these call models:

- Direct call model with per-call admission – In this case, the endpoint may indicate capacity remaining in the ARQ, DRQ, or BRQ messages.
- Direct call model with pre-granted admission – In this case, the endpoint may indicate capacity in RRQ or RAI messages (in the case that the endpoint is a Gateway).
- Gatekeeper routed call model with per-call admission – The endpoint may provide capacity information in an ARQ, DRQ, or BRQ messages.
- Gatekeeper routed call model with pre-granted admission – The endpoint may include capacity information in the call signalling messages, such as Setup or Release Complete. In this case, the originating endpoint may provide its capacity information in a Setup, while the terminating endpoint may provide its capacity information in an Alerting or Connect. Each endpoint may provide updated capacity information using the Release Complete message.

In any case, a Gatekeeper may use the IRQ/IRR exchange to audit an endpoint to potentially discover the call capacity of the endpoint. It should be noted that including capacity information in messages that are already required to be sent to a Gatekeeper, such as an ARQ when not using pre-granted admission or a Setup in a Gatekeeper routed call, rather than sending additional messages for this purpose is preferred. However, if a Gateway receives a Release Complete and is operating in a pre-granted admission mode, it should send an IRR to the Gatekeeper to enable it to maintain more accurate capacity information.

If an endpoint provides call capacity information, it should provide capacity information in an RRQ and should indicate its call capacity reporting capabilities in the RRQ. A Gatekeeper may request via the RCF and IRQ messages that an endpoint provide call capacity information. An endpoint that has indicated the ability to report call capacity shall report its capacity as requested by the gatekeeper. Other than in the initial RRQ, an endpoint should not report maximum call capacity

unless its gatekeeper requests call capacity information in an IRQ message. An endpoint may use capacity information in a BRQ, IRR, or RAI to inform the Gatekeeper of sudden changes, such as that caused by a hardware failure.

An endpoint may signal that it has different call capacities for different supported protocols (i.e., T.120, H.320, H.321, voice, etc.). However, since equipment manufacturers may utilize the same resources for multiple protocols, the Gatekeeper should make no assumptions about how the endpoint's call capacity for one supported protocol may change when the endpoint engages in a call utilizing a different protocol.

A Gateway may signal the call capacity by **group** where the **group** could represent a set of circuits associated with a particular interface or a carrier, for example. This feature allows the Gatekeeper to track the call capacity separately for each group. The **group** may be the same as that reported in the **circuitID** for a particular call.

NOTE – The capacity information reported in any message is of an advisory nature and, due to race conditions, sudden changes in the endpoint, or local allocation of resources, may not be absolutely accurate.

7.8 Caller identification services

7.8.1 Description of services

This clause describes the caller identification services, which includes:

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

7.8.1.1 Calling party address presentation

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

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

7.8.1.2 Calling party address restriction

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

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

7.8.1.3 Connected party address presentation

Connected party address presentation is a feature which provides the alias address of the connected or answering party to the calling party. The connected party address may be provided by the connected endpoint or by the Gatekeeper for Gatekeeper routed calls. When the call is routed

through the Gatekeeper with which the connected endpoint is registered, the Gatekeeper may provide a screening service that assures the address provided is actually that of the connected party. The Gatekeeper may also provide the connected party address when no address is provided by the connected party or when the connected party provides an address other than an address with which the connected party registered.

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

7.8.1.4 Connected party address restriction

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

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

7.8.1.5 Called (alerting) party address presentation

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

7.8.1.6 Called (alerting) party address restriction

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

7.8.1.7 Busy party address presentation

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

7.8.1.8 Busy party address restriction

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

7.8.2 Messages and information elements

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

7.8.2.1 Calling party address information

Calling party address information appears in the Setup message.

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

Alternatively, calling party information may appear in the **sourceAddress**, **presentationIndicator**, and **screeningIndicator** fields of the Setup message. This mode of operation is required when the **sourceAddress** is not in any form of telephone number (i.e., **sourceAddress** is not a type of **dialledDigits** or **partyNumber**). In accordance with 7.2.2.6/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.

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

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

7.8.2.2 Connected party address information

Connected party address information appears in the Connect message.

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

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

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

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

7.8.2.3 Called (alerting) party address information

Alerting party address information appears in the Alerting message.

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

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

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

7.8.2.4 Busy party address information

Busy party address information appears in the Release Complete message.

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

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

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

7.8.3 Actions at the originating endpoint

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

7.8.3.1 Gateway as originating endpoint

In the case of a Setup message received by a Gateway from the ISDN, the caller's number and presentation information reside in the Calling Party Number IE. The Gateway shall send a Setup message on the packet network with the Calling Party Number IE containing the same information as was found in the Setup message from the SCN 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/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).

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

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

7.8.3.2 Terminal or MCU as originating endpoint

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

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

7.8.4 Actions at the terminating endpoint

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

7.8.4.1 Gateway as terminating endpoint

A PSTN Gateway in receipt of a Setup message from the packet network shall copy the information found in the Calling Party Number IE from the Setup message to the 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.

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

7.8.4.2 Terminal or MCU as terminating endpoint

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

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

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

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

7.8.5 Actions at a gatekeeper

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

7.8.5.1 Gateway as originating endpoint

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

7.8.5.2 Terminal or MCU as originating endpoint

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

7.8.5.3 Gateway as terminating endpoint

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

7.8.5.4 Terminal or MCU as terminating endpoint

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

7.9 Generic extensible framework

The generic extensibility framework allows new features to be readily added to the protocol without affecting the underlying H.225.0 core specification. The extensible framework consists of two parts:

- Carriage of opaque data within H.225.0 messages.
- Negotiation of supported features.

Support of the **generic extensibility framework** is optional.

7.9.1 Format of a GenericData structure

Opaque data may be carried in a sub-set of RAS messages and H.225.0 call signalling messages in the **genericData** field.

The **GenericData** structure primarily consists of an identifier and zero or more parameters, which allows flexible definition of opaque data and features. The **GenericData** structure consists of an **id** to identify the generic data and the **parameters** field to convey the actual parameters.

Each parameter also contains an identifying **id** and a **content** field. The **content** field supports a number of different data types, including **raw**, **text**, **unicode**, **bool**, **number8**, **number16**, **number32**, **id**, **compound**, and **nested**. This allows for flexible definition of generic data and eases implementation. However, it is expected that for generic data that contain a large number of parameters, the **raw** form of **content** shall be used, which will contain ASN.1 data.

7.9.2 Negotiation using the extensible framework – General

The extensible framework provides a common method for feature negotiation that operates over multiple domains and may be managed and configured by different operational entities. Hence, entities do not require *a priori* knowledge of other entities' feature sets to operate successfully.

The mechanism used to negotiate features in both RAS and call signalling uses the **FeatureDescriptor**, which is an alias of a **GenericData** structure as described above. This allows a feature to be identified and have parameters associated with it.

Intermediate signalling entities may – subject to security issues – add their needed, desired and supported features to messages that pass through them. Intermediate entities may remove desired and supported features specified in messages before passing them on. Intermediate entities shall not remove needed feature fields unless they intend to support the features that they are removing. If the intermediate entity does not wish to allow a needed feature, then it shall reject the transaction.

If an intermediate entity elects to support a requested feature signalled in a message, then it should remove the feature request from the message before passing it on. By some means, the intermediate entity should signal back to the requesting entity that the feature is supported. This may be achieved by modifying the response from the remote entity, or by generating its own message.

7.9.3 Negotiation using the extensible framework – RAS

RAS feature negotiation applies to the discovery, registration, and call setup phase. In particular, it applies to the exchange of discovery messages (GRQ, GCF, GRJ), registration messages (RRQ, RCF, RRJ), admission request messages (ARQ, ACF, ARJ), location request messages (LRQ, LCF, LRJ), service control messages (SCI/SCR), and the NonStandardMessage.

In RAS negotiation, entities may specify the set of features that they need for a transaction to be successful, the set of features they desire, and the set of features they support.

7.9.3.1 Processing by the requesting entity

A requesting entity (usually an endpoint) uses the elements in the **FeatureSet** structure to specify the various types of features it requires. It specifies the set of features that it needs using the **neededFeatures** field, the set of features that it desires using the **desiredFeatures** field, and the set of features that it supports in the **supportedFeatures** field. All three of these fields are in the **FeatureSet** structure.

In response to its request, a requesting entity should receive either a confirm or reject message.

If the request is rejected, the responding entity may have included a set of **neededFeatures** that the requesting entity must support in order for the request to be successful. If this is the case and the requesting entity supports the needed features, the requesting entity may reissue a request specifying support for the features needed by the responding entity.

If the request is accepted, special procedures need to be applied to ensure that the negotiation operates in a backwards-compatible manner. This is done by the requesting entity checking that the features that it specified as needed are listed as **supportedFeatures** in the response. If a requesting entity does not observe the features it needs in the response message's **supportedFeatures** field, then it shall assume that the responding entity does not support the features that it needs. If the requesting entity determines that it cannot continue under these circumstances, then it shall undo the operation it was trying to perform (i.e., send a DRQ if it originally sent an ARQ and so forth), so that the state in the responding entity is rolled back.

7.9.3.2 Processing by the responding entity

The responding entity (typically a Gatekeeper) looks at the features specified in the **neededFeatures** field of the request to determine if it can accept the request. It also looks in the **neededFeatures**, **desiredFeatures** and **supportedFeatures** fields to determine whether the features needed by it are supported by the requesting entity.

If the responding entity is a Gatekeeper that sends an LRQ in response to receiving an ARQ, the Gatekeeper shall copy any features that are not provided by the Gatekeeper into the LRQ. In trying to determine whether the necessary set of features are supported, the Gatekeeper shall examine the supported features of the endpoint to which the ARQ may resolve, either locally or in response to an LCF, and the features supported by the Gatekeeper.

If the responding entity determines that the necessary sets of features are supported by both entities, then the responding entity may acknowledge the request. The responding entity lists the set of features that it chooses to support in the **supportedFeatures** field of it reply. If the request is accepted, then all of the **neededFeatures** from the request must be included in the **supportedFeatures** field of the reply. The responding entity may also include **desiredFeatures**.

If the responding entity needs additional features to be supported by the requesting entity, it shall reject the request. If it wishes to declare which features must be supported for the request to be successful, this should be specified using the **neededFeatures** field of the reject message. The responding entity may also include any **desiredFeatures** and **supportedFeatures** in the reject message.

7.9.4 Negotiation using the extensible framework – Call signalling

The following describes the negotiation process for the call signalling channel.

7.9.4.1 Processing by the initiating endpoint

An initiating endpoint may specify the features it needs for a call, the features it desires, and the features it supports. It specifies the set of features that it needs using the **neededFeatures** field in Setup. It also specifies the set of features that it desires using the **desiredFeatures** field and the set of features that it supports using the **supportedFeatures** field.

If the call is rejected, one or more responding entities may have included a set of **neededFeatures** that the initiating endpoint must support in order for the call to be successful. If this is the case, and the initiating endpoint supports the needed features, the initiating endpoint may reinitiate a call specifying support for the features needed by the various entities along the call signalling path.

If the call is accepted, the initiating endpoint shall check that the features that it specified as needed are listed as **supportedFeatures** in the Alerting or Connect message. If an initiating endpoint does not observe the features it needs in the message's **supportedFeatures** field, then it shall assume that the entities along the call signalling path do not support the features that it needs. If the initiating entity determines that it cannot continue under these circumstances, then it shall clear the call using Release Complete.

When an initiating endpoint receives an empty capability set as a result of third-party pause and re-routing, it shall remove any knowledge it has of any remote entities capabilities. When the endpoint receives a non-empty capability set, it shall send its feature set using the **featureSet** field in a Facility message with the **reason** field set to **featureSetUpdate**. In this message, the **replacementFeatureSet** field shall be set to TRUE. When the feature set from the remote endpoint is received in a Facility message, the contents may be interpreted in the same way as above.

7.9.4.2 Processing by intermediate entities

Intermediate entities along the call signalling path, such as Gatekeepers and Border Elements, may also interact with the negotiation process.

Intermediate entities along the signalling path may – subject to security issues – add their needed, desired and supported features to the call signalling messages that pass through them. Intermediate entities may remove desired and supported features specified in messages (including Setup, Alerting and Connect) before passing them on. Intermediate entities shall not remove needed feature fields from a Setup message or a Facility message unless they intend to support the features that they remove. If the intermediate entity does not wish to allow a needed feature, then it shall reject or terminate the call.

If an intermediate entity elects to support a requested feature signalled in a Setup message, then it should remove the feature request from the Setup message before passing it on. The intermediate entity should signal supported features it supports in the Alerting (if sent) or Connect messages along with the destinations supported feature set.

When an intermediate entity receives a **featureSet** parameter in a Facility message with the **replacementFeatureSet** field set to TRUE, it shall modify the features indicated according to its requirements in a similar fashion to how it modifies the features signalled in the Setup, Alerting or Connect messages. It should then pass the message on.

7.9.4.3 Processing by the called endpoint

The called endpoint looks at the features specified in the **neededFeatures** field of the Setup message to determine if it can accept the call. It also looks in the **neededFeatures**, **desiredFeatures** and **supportedFeatures** fields to determine whether the features needed by it are supported by the various entities along the call signalling path.

If the called endpoint determines that the necessary sets of features are supported by the appropriate entities, then the called endpoint may accept the call. The called endpoint lists the set of features that it chooses to support in the **supportedFeatures** field of the Alerting (if sent) and Connect messages. If the call is accepted, then all of the **neededFeatures** from the Setup message must be declared in the **supportedFeatures** field of the Alerting (if sent) or Connect call signalling messages. The called endpoint may also include **desiredFeatures** within the message.

If the called endpoint needs additional features to be supported by the various entities along the call signalling path, it shall reject the call by sending a Release Complete. If it wishes to declare which features must be supported for the call to be successful, this should be specified using the **neededFeatures** field in the Release Complete message. The called endpoint may also include any **desiredFeatures** and **supportedFeatures** in the Release Complete message.

When a called endpoint receives an empty capability set as a result of third-party pause and re-routing, it shall act in the same way as if it had initiated the call. That is, it shall remove any knowledge it has of any remote entities capabilities. When the endpoint later receives a non-empty capability set, it shall send its feature set using the **featureSet** field in a Facility message with the **reason** field set to **featureSetUpdate**. In this message, the **replacementFeatureSet** field shall be set to TRUE. When the feature set from the remote endpoint is received in a Facility message, the contents may be interpreted in the same way as above.

8 Call signalling procedures

The provision of the communication is made in the following steps:

– Phase A: Call setup (see 8.1).

– Phase B: Initial communication and capability exchange (see 8.2).

– Phase C: Establishment of audiovisual communication (see 8.3).

– Phase D: Call services (see 8.4).

– Phase E: Call termination (see 8.5).

8.1 Phase A – Call setup

Call setup takes place using the call control messages defined in ITU-T Rec. H.225.0 according to the call control procedures defined below. Requests for bandwidth reservation should take place at the earliest possible phase.

If both the alias address and the Transport Address are specified, preference shall be given to the alias address.

There is no explicit synchronization or locking between two endpoints during the call setup procedure. This implies that endpoint A can send a Setup message to endpoint B at exactly the same time that endpoint B sends a Setup message to endpoint A. It is up to the application to determine if only one call is desired and to take the appropriate action. This action may be for an endpoint to indicate that it is busy whenever it has an outstanding Setup message. If an endpoint can support more than one simultaneous call, it should indicate that it is busy whenever it receives a Setup message from the same endpoint to which it has an outstanding Setup message.

An endpoint shall be capable of sending the Alerting message. Alerting has the meaning that the called party (user) has been alerted of an incoming call. Alerting shall only be originated by the ultimate called endpoint and then only when it has alerted the user. In the case of interworking through a Gateway, the Gateway shall send Alerting when it receives a ring indication from the SCN. If an endpoint can respond to a Setup message with a Connect, Call Proceeding, or Release Complete within 4 seconds, it is not required to send the Alerting message. An endpoint sending the Setup message can expect to receive either an Alerting, Connect, Call Proceeding, or Release Complete message within 4 seconds after successful transmission.

The Connect message should be sent only if it is certain that the H.245 capability exchange will conclude successfully and a minimum level of communications can take place. This is to maintain the consistency of the meaning of the Connect message between packet based networks and circuit switched networks.

8.1.1 Basic call setup – neither endpoint registered

In the scenario shown in Figure 29 neither endpoint is registered to a Gatekeeper. The two endpoints communicate directly. Endpoint 1 (calling endpoint) sends the Setup (1) message to the well-known Call Signalling Channel TSAP Identifier of Endpoint 2. Endpoint 2 responds with the Connect (4) message which contains an H.245 Control Channel Transport Address for use in H.245 signalling.

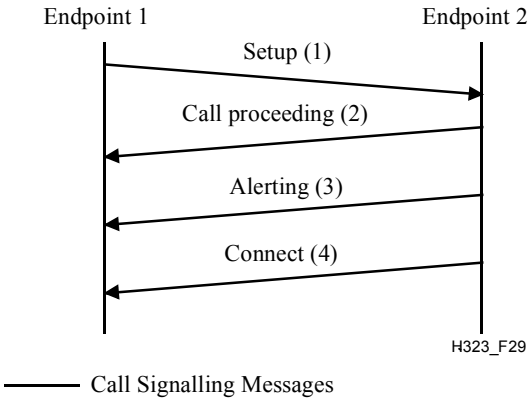


Figure 29/H.323 – Basic call setup, no Gatekeepers

8.1.2 Both endpoints registered to the same gatekeeper

In the scenario shown in Figure 30, both endpoints are registered to the same Gatekeeper, and the Gatekeeper has chosen direct call signalling. Endpoint 1 (calling endpoint) initiates the ARQ (1)/ACF (2) exchange with that Gatekeeper. The Gatekeeper shall return the Call Signalling Channel Transport Address of Endpoint 2 (called endpoint) in the ACF. Endpoint 1 then sends the Setup (3) message to Endpoint 2 using that Transport Address. If Endpoint 2 wishes to accept the call, it initiates an ARQ (5)/ACF (6) exchange with the Gatekeeper. It is possible that an ARJ (6) is received by Endpoint 2, in which case it sends Release Complete to Endpoint 1. Endpoint 2 responds with the Connect (8) message which contains an H.245 Control Channel Transport Address for use in H.245 signalling.

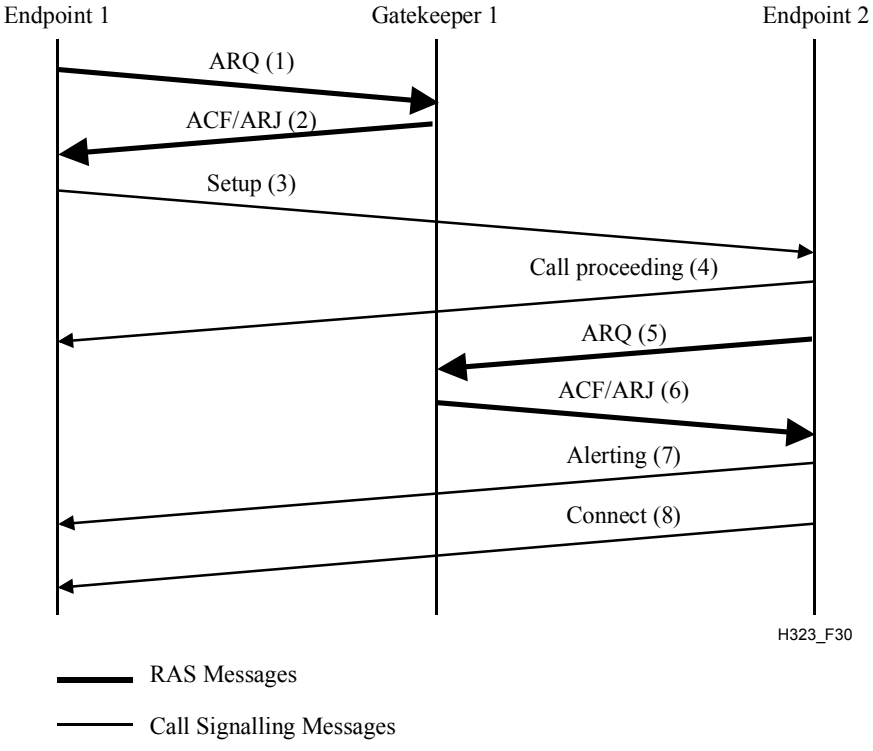


Figure 30/H.323 – Both endpoints registered, same gatekeeper – Direct call signalling

In the scenario shown in Figure 31, both endpoints are registered to the same Gatekeeper, and the Gatekeeper has chosen to route the call signalling. Endpoint 1 (calling endpoint) initiates the ARQ (1)/ACF (2) exchange with that Gatekeeper. The Gatekeeper shall return a Call Signalling Channel Transport Address of itself in the ACF. Endpoint 1 then sends the Setup (3) message using that Transport Address. The Gatekeeper then sends the Setup (4) message to Endpoint 2. If Endpoint 2 wishes to accept the call, it initiates an ARQ (6)/ACF (7) exchange with the Gatekeeper. It is possible that an ARJ (7) is received by Endpoint 2, in which case it sends Release Complete to the Gatekeeper. Endpoint 2 responds with the Connect (9) message which contains an H.245 Control Channel Transport Address for use in H.245 signalling. The Gatekeeper sends the Connect (10) message to Endpoint 1 which may contain the Endpoint 2 H.245 Control Channel Transport Address or a Gatekeeper H.245 Control Channel Transport Address, based on whether the Gatekeeper chooses to route the H.245 Control Channel or not.

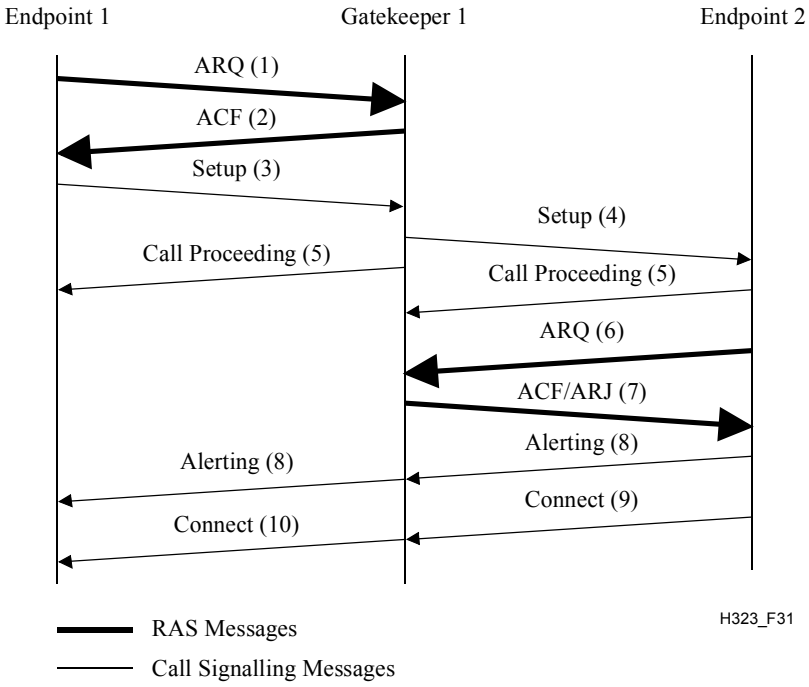


Figure 31/H.323 – Both endpoints registered, same Gatekeeper – Gatekeeper routed call signalling

8.1.3 Only calling endpoint has gatekeeper

In the scenario shown in Figure 32, Endpoint 1 (calling endpoint) is registered with a Gatekeeper, Endpoint 2 (called endpoint) is not registered with a Gatekeeper, and the Gatekeeper has chosen direct call signalling. Endpoint 1 initiates the ARQ (1)/ACF (2) exchange with the Gatekeeper. Endpoint 1 then sends the Setup (3) message to Endpoint 2 using the well-known Call Signalling Channel Transport Address. If Endpoint 2 wishes to accept the call, it responds with the Connect (6) message which contains an H.245 Control Channel Transport Address for use in H.245 signalling.

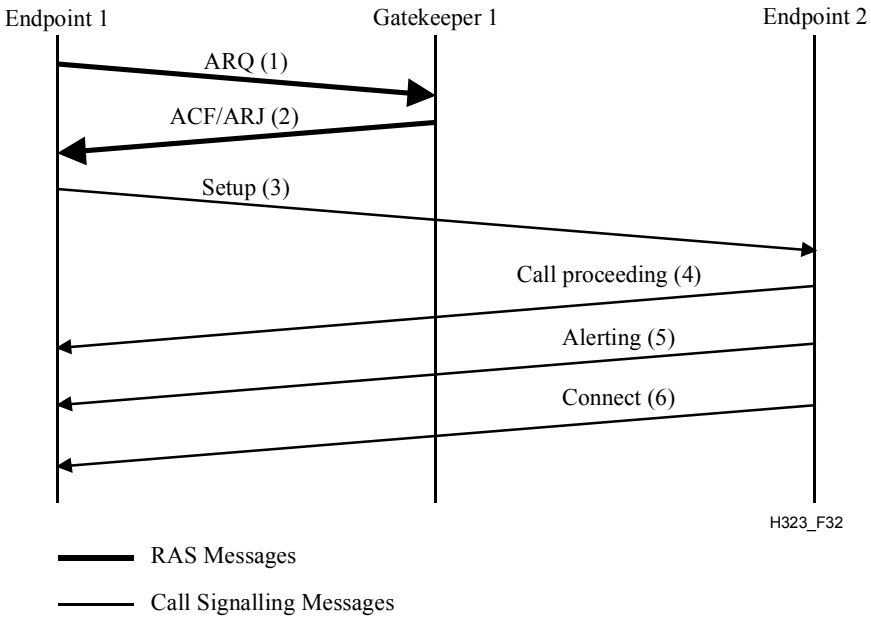


Figure 32/H.323 – Only calling endpoint registered – Direct call signalling

In the scenario shown in Figure 33, Endpoint 1 (calling endpoint) is registered with a Gatekeeper, Endpoint 2 (called endpoint) is not registered with a Gatekeeper, and the Gatekeeper has chosen to route the call signalling. Endpoint 1 (calling endpoint) initiates the ARQ (1)/ACF (2) exchange with that Gatekeeper. The Gatekeeper shall return a Call Signalling Channel Transport Address of itself in the ACF (2). Endpoint 1 then sends the Setup (3) message using that Transport Address. The Gatekeeper then sends the Setup (4) message to the well-known Call Signalling Channel Transport Address of Endpoint 2. If Endpoint 2 wishes to accept the call, it responds with the Connect (7) message which contains an H.245 Control Channel Transport Address for use in H.245 signalling. The Gatekeeper sends the Connect (8) message to Endpoint 1 which may contain the Endpoint 2 H.245 Control Channel Transport Address or a Gatekeeper H.245 Control Channel Transport Address, based on whether the Gatekeeper chooses to route the H.245 Control Channel or not.

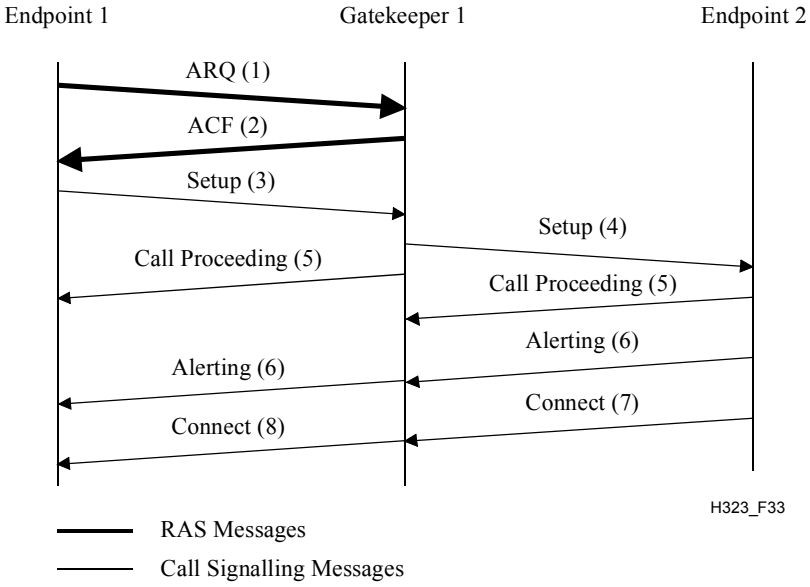


Figure 33/H.323 – Only calling endpoint registered – Gatekeeper routed call signalling

8.1.4 Only called endpoint has gatekeeper

In the scenario shown in Figure 34, Endpoint 1 (calling endpoint) is not registered with a Gatekeeper, Endpoint 2 (called endpoint) is registered with a Gatekeeper, and the Gatekeeper has chosen direct call signalling. Endpoint 1 sends the Setup (1) message to Endpoint 2 using the well-known Call Signalling Channel Transport Address. If Endpoint 2 wishes to accept the call, it initiates an ARQ (3)/ACF (4) exchange with the Gatekeeper. It is possible that an ARJ (4) is received by Endpoint 2, in which case it sends Release Complete to Endpoint 1. Endpoint 2 responds with the Connect (6) message which contains an H.245 Control Channel Transport Address for use in H.245 signalling.

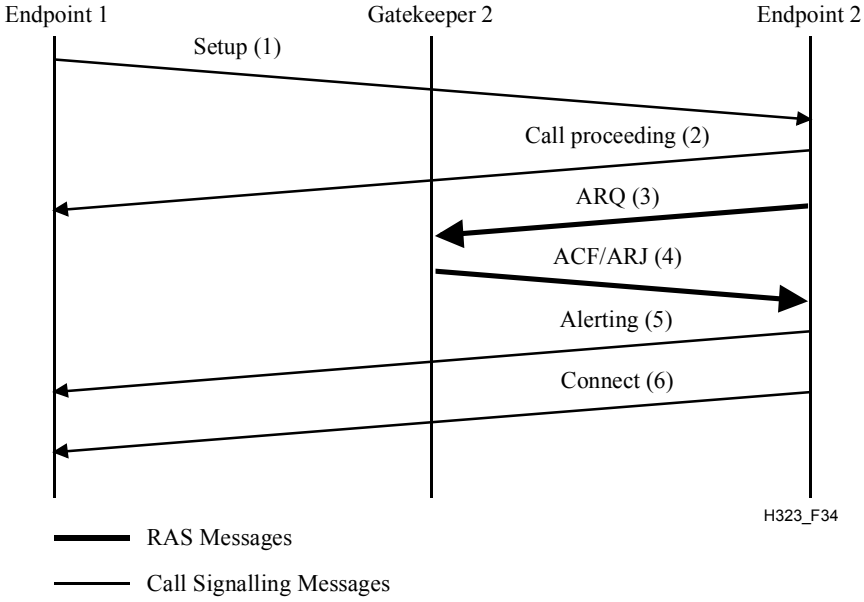


Figure 34/H.323 – Only called endpoint registered – Direct call signalling

In the scenario shown in Figure 35, Endpoint 1 (calling endpoint) is not registered with a Gatekeeper, Endpoint 2 (called endpoint) is registered with a Gatekeeper, and the Gatekeeper has chosen to route the call signalling. Endpoint 1 (calling endpoint) sends a Setup (1) message to the well-known Call Signalling Channel Transport Address of Endpoint 2. If Endpoint 2 wishes to accept the call, it initiates the ARQ (3)/ACF (4) exchange with that Gatekeeper. If acceptable, the Gatekeeper shall return a Call Signalling Channel Transport Address of itself in the ARJ (4) with a cause code of **routeCallToGatekeeper**. Endpoint 2 replies to Endpoint 1 with a Facility (5) message containing the Call Signalling Transport Address of its Gatekeeper. Endpoint 1 then sends the Release Complete (6) message to Endpoint 2. Endpoint 1 sends a Setup (7) message to the Gatekeeper's Call Signalling Channel Transport Address. The Gatekeeper sends the Setup (8) message to Endpoint 2. Endpoint 2 initiates the ARQ (9)/ACF (10) exchange with that Gatekeeper. Endpoint 2 then responds with the Connect (12) message which contains its H.245 Control Channel Transport Address for use in H.245 signalling. The Gatekeeper sends the Connect (13) message to Endpoint 1 which may contain the Endpoint 2 H.245 Control Channel Transport Address or a Gatekeeper H.245 Control Channel Transport Address, based on whether the Gatekeeper chooses to route the H.245 Control Channel or not.

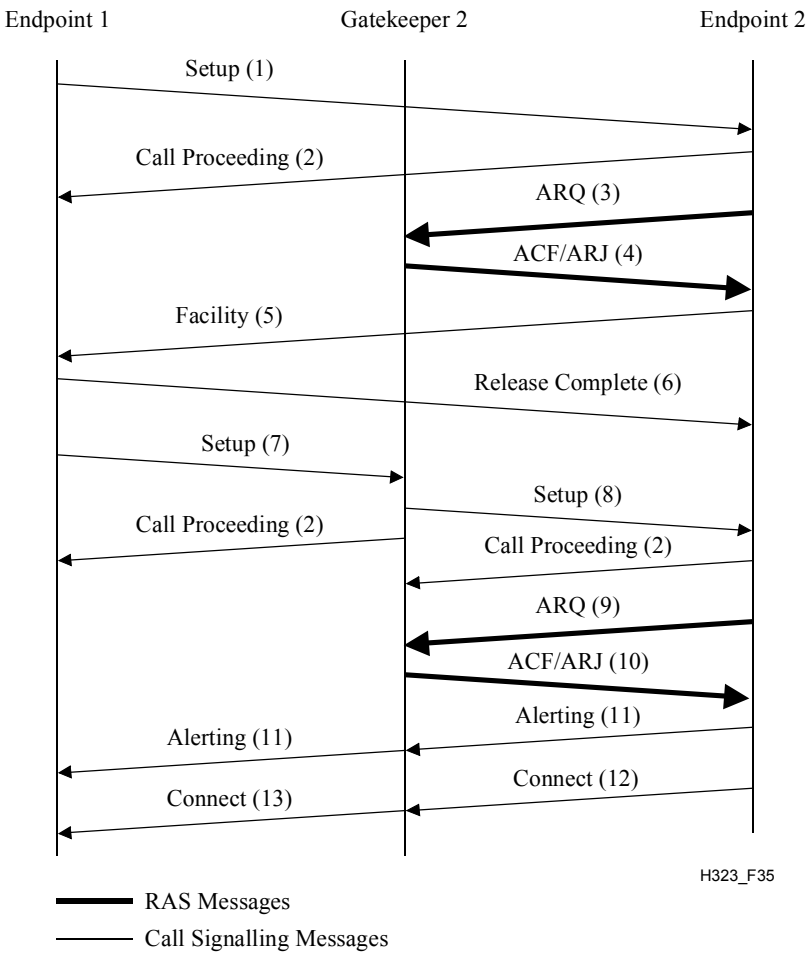


Figure 35/H.323 – Only called endpoint registered – Gatekeeper routed call signalling

8.1.5 Both endpoints registered to different gatekeepers

In the scenario shown in Figure 36, both endpoints are registered to different Gatekeepers, and both Gatekeepers choose direct call signalling. Endpoint 1 (calling endpoint) initiates the ARQ (1)/ACF (2) exchange with Gatekeeper 1. Gatekeeper 1 may return the Call Signalling Channel Transport Address of Endpoint 2 (called endpoint) in the ACF if Gatekeeper 1 has a method of communicating with Gatekeeper 2. Endpoint 1 then sends the Setup (3) message to either the Transport Address returned by the Gatekeeper (if available) or to the well-known Call Signalling Channel Transport Address of Endpoint 2. If Endpoint 2 wishes to accept the call, it initiates an ARQ (5)/ACF (6) exchange with Gatekeeper 2. It is possible that an ARJ (6) is received by Endpoint 2, in which case it sends Release Complete to Endpoint 1. Endpoint 2 responds with the Connect (8) message which contains an H.245 Control Channel Transport Address for use in H.245 signalling.

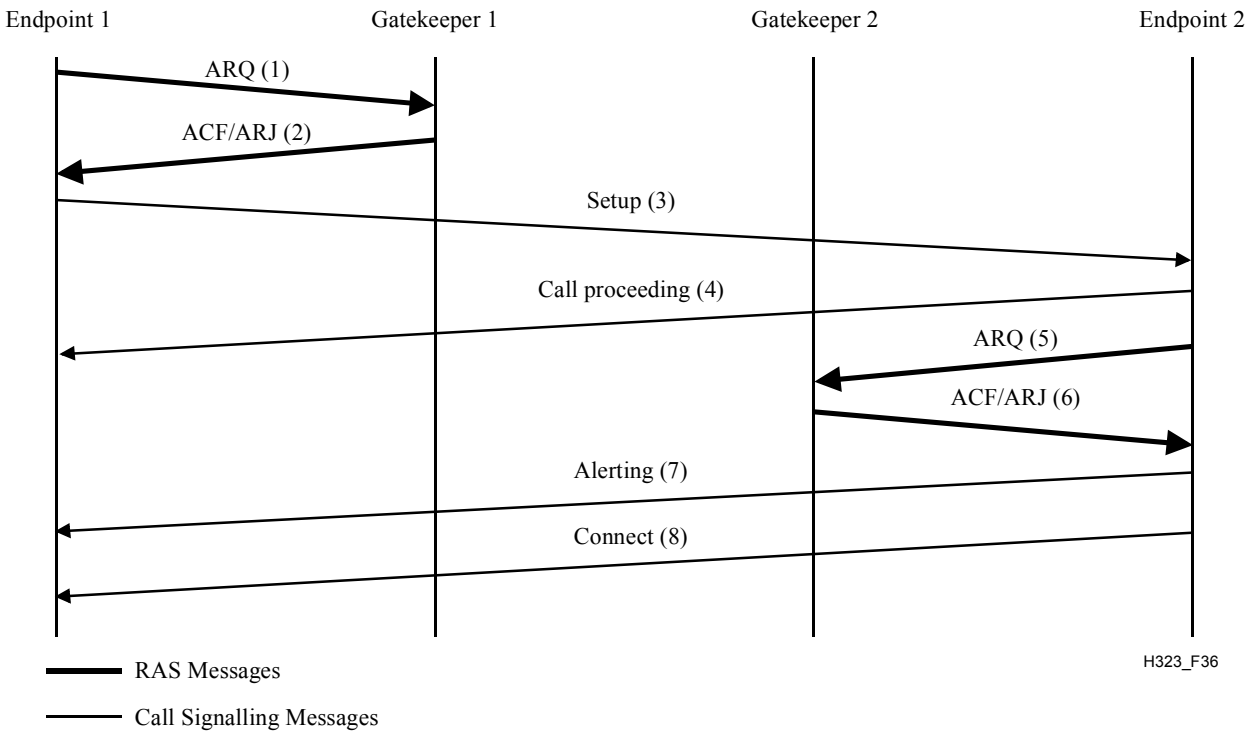


Figure 36/H.323 – Both endpoints registered – Both gatekeepers direct call signalling

In the scenario shown in Figure 37, both endpoints are registered to different Gatekeepers, the calling endpoint's Gatekeeper chooses direct call signalling, and the called endpoint's Gatekeeper chooses to route the call signalling. Endpoint 1 (calling endpoint) initiates the ARQ (1)/ACF (2) exchange with Gatekeeper 1. Gatekeeper 1 may return the Call Signalling Channel Transport Address of Endpoint 2 (called endpoint) in the ACF (2) if Gatekeeper 1 has a method of communicating with Gatekeeper 2. Endpoint 1 then sends the Setup (3) message to either the Transport Address returned by the Gatekeeper (if available) or to the well-known Call Signalling Channel Transport Address of Endpoint 2. If Endpoint 2 wishes to accept the call, it initiates the ARQ (5)/ACF (6) exchange with Gatekeeper 2. If acceptable, Gatekeeper 2 shall return a Call Signalling Channel Transport Address of itself in the ARJ (6) with a cause code of **routeCallToGatekeeper**. Endpoint 2 replies to Endpoint 1 with a Facility (7) message containing the Call Signalling Transport Address of Gatekeeper 2. Endpoint 1 then sends the Release Complete (8) message to Endpoint 2. Endpoint 1 shall send a DRQ (9) to Gatekeeper 1 which responds with DCF (10). Endpoint 1 then initiates a new ARQ (11)/ACF (12) exchange with Gatekeeper 1. Endpoint 1 sends a Setup (13) message to the Gatekeeper's Call Signalling Channel Transport Address. Gatekeeper 2 sends the Setup (14) message to Endpoint 2. Endpoint 2 initiates the ARQ (15)/ACF (16) exchange with Gatekeeper 2. Endpoint 2 then responds with the Connect (18) message which contains its H.245 Control Channel Transport Address for use in H.245 signalling. Gatekeeper 2 sends the Connect (19) message to Endpoint 1 which may contain the Endpoint 2 H.245 Control Channel Transport Address or a Gatekeeper 2 H.245 Control Channel Transport Address, based on whether the Gatekeeper chooses to route the H.245 Control Channel or not.

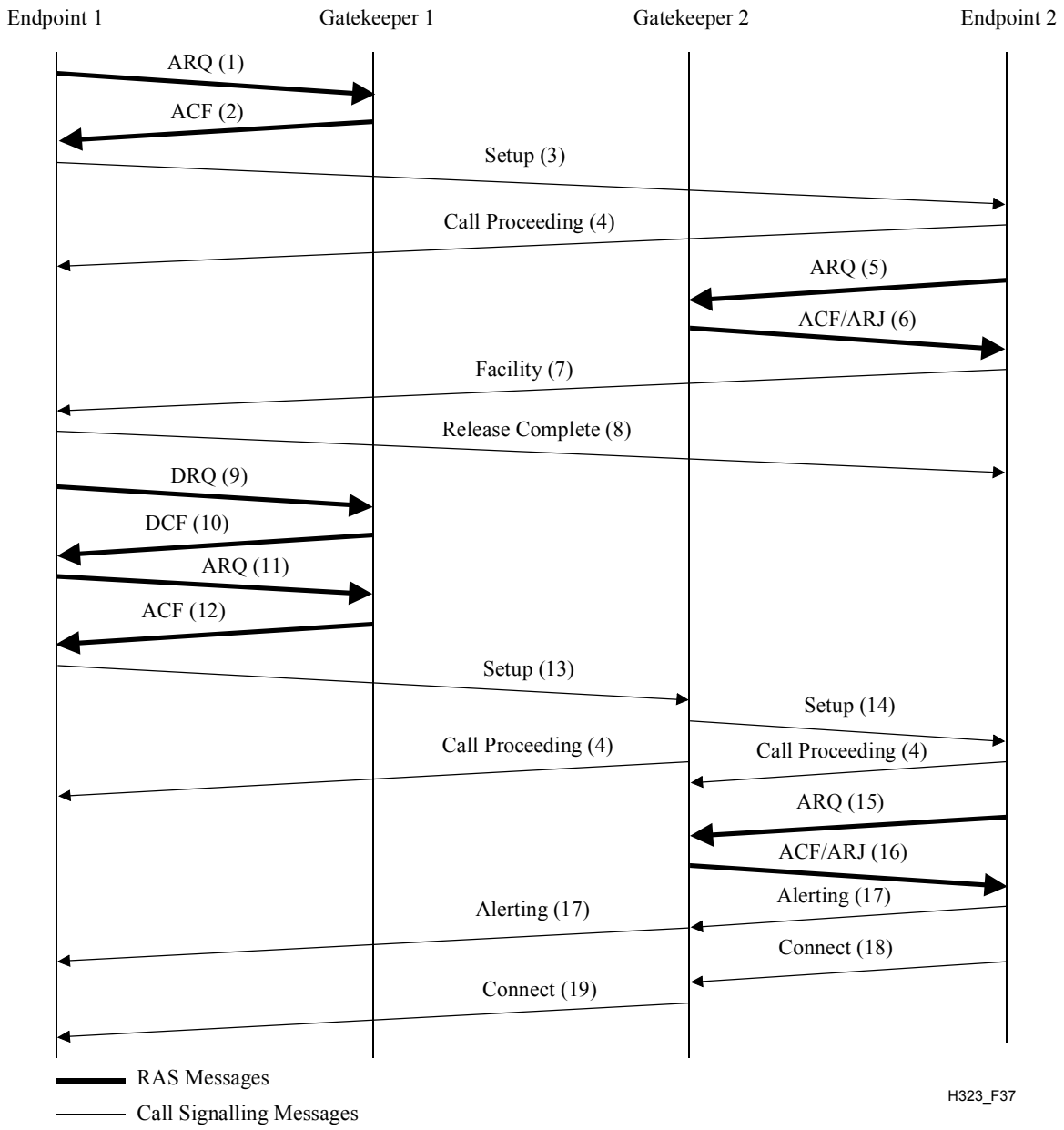


Figure 37/H.323 – Both endpoints registered – Direct/routed call signalling

In the scenario shown in Figure 38, both endpoints are registered to different Gatekeepers, the calling endpoint's Gatekeeper chooses to route the call signalling, and the called endpoint's Gatekeeper chooses direct call signalling. Endpoint 1 (calling endpoint) initiates the ARQ (1)/ACF (2) exchange with Gatekeeper 1. Gatekeeper 1 shall return a Call Signalling Channel Transport Address of itself in the ACF (2). Endpoint 1 then sends the Setup (3) message using that Transport Address. Gatekeeper 1 then sends the Setup (4) message containing its Call Signalling Channel Transport Address to the well-known Call Signalling Channel Transport Address of Endpoint 2. If Endpoint 2 wishes to accept the call, it initiates the ARQ (6)/ACF (7) exchange with Gatekeeper 2. It is possible that an ARJ (7) is received by Endpoint 2, in which case it sends Release Complete to Endpoint 1. Endpoint 2 responds to Gatekeeper 1 with the Connect (9) message which contains its H.245 Control Channel Transport Address for use in H.245 signalling. Gatekeeper 1 sends the Connect (10) message to Endpoint 1 which may contain the Endpoint 2 H.245 Control Channel Transport Address or a Gatekeeper 1 H.245 Control Channel Transport Address, based on whether the Gatekeeper chooses to route the H.245 Control Channel or not.

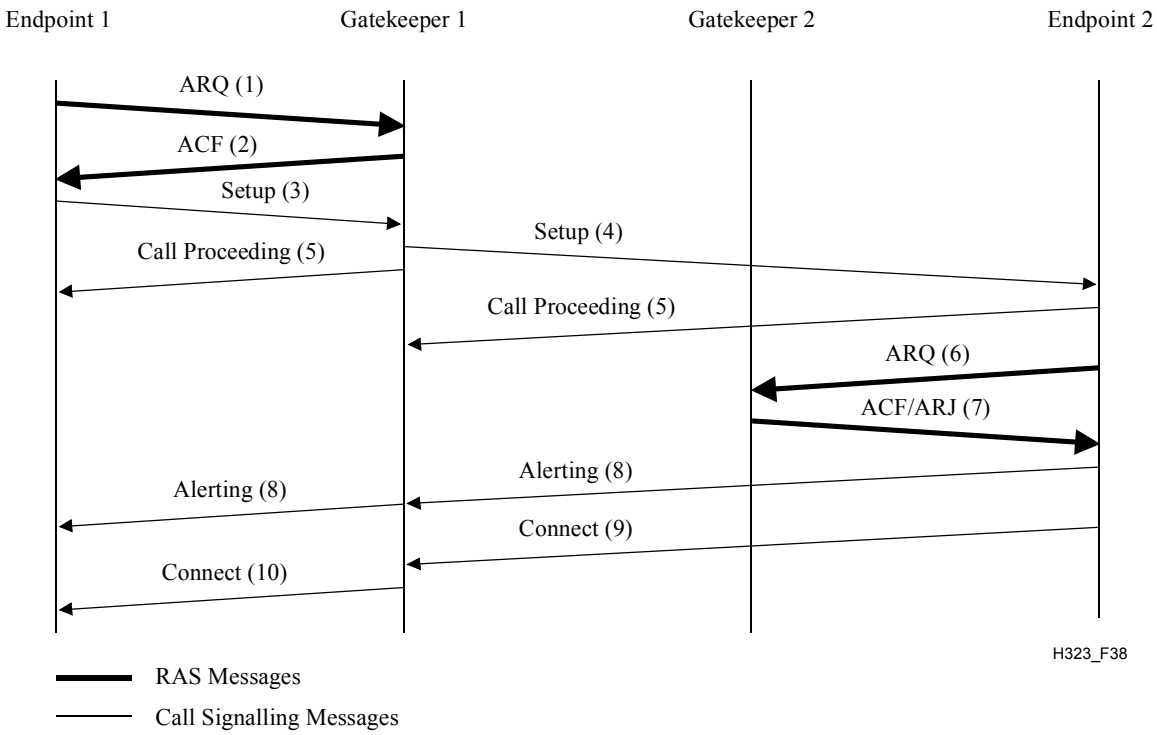


Figure 38/H.323 – Both endpoints registered – Routed/direct call signalling

In the scenario shown in Figure 39, both endpoints are registered to different Gatekeepers, and both Gatekeepers choose to route the call signalling. Endpoint 1 (calling endpoint) initiates the ARQ (1)/ACF (2) exchange with Gatekeeper 1. Gatekeeper 1 shall return a Call Signalling Channel Transport Address of itself in the ACF (2). Endpoint 1 then sends the Setup (3) message using that Transport Address. Gatekeeper 1 then sends the Setup (4) message to the well-known Call Signalling Channel Transport Address of Endpoint 2. If Endpoint 2 wishes to accept the call, it initiates the ARQ (6)/ACF (7) exchange with Gatekeeper 2. If acceptable, Gatekeeper 2 shall return a Call Signalling Channel Transport Address of itself in the ARJ (7) with a cause code of **routeCallToGatekeeper**. Endpoint 2 replies to Gatekeeper 1 with a Facility (8) message containing the Call Signalling Transport Address of Gatekeeper 2. Gatekeeper 1 then sends the Release Complete (9) message to Endpoint 2. Gatekeeper 1 sends a Setup (10) message to Gatekeeper 2's Call Signalling Channel Transport Address. Gatekeeper 2 sends the Setup (11) message to Endpoint 2. Endpoint 2 initiates the ARQ (12)/ACF (13) exchange with Gatekeeper 2. Endpoint 2 then responds to Gatekeeper 2 with the Connect (15) message which contains its H.245 Control Channel Transport Address for use in H.245 signalling. Gatekeeper 2 sends the Connect (16) message to Gatekeeper 1 which may contain the Endpoint 2 H.245 Control Channel Transport Address or a Gatekeeper 2 H.245 Control Channel Transport Address, based on whether the Gatekeeper 2 chooses to route the H.245 Control Channel or not. Gatekeeper 1 sends the Connect (17) message to Endpoint 1 which may contain the H.245 Control Channel Transport Address sent by Gatekeeper 2 or a Gatekeeper 1 H.245 Control Channel Transport Address, based on whether the Gatekeeper 1 chooses to route the H.245 Control Channel or not.

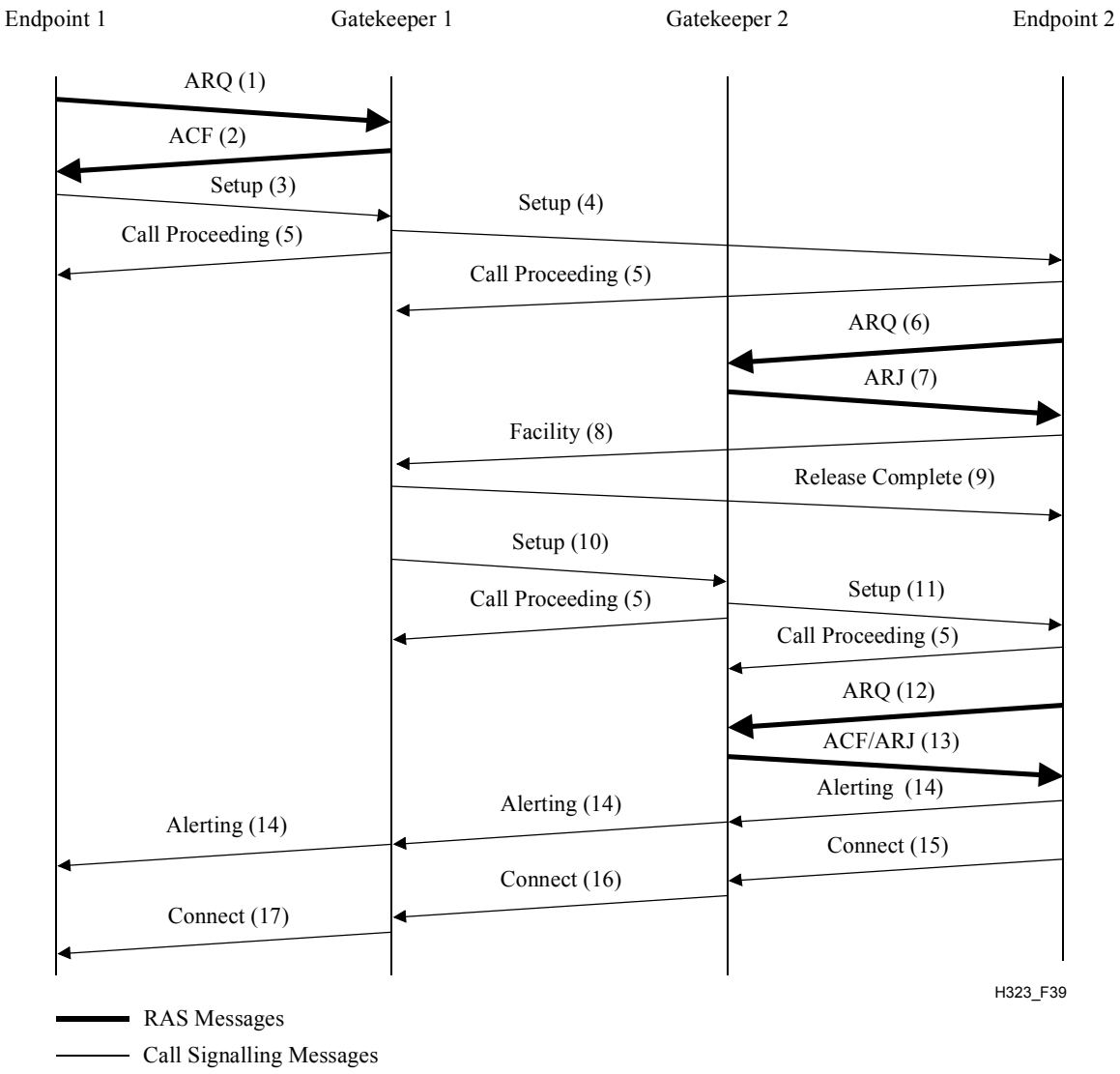


Figure 39/H.323 – Both endpoints registered – Both Gatekeepers routing call signalling

8.1.6 Optional called endpoint signalling

The procedures defined in 8.1.4 and 8.1.5 show that when a called endpoint is registered to a Gatekeeper, a Setup message is initially sent to the called endpoint from the calling endpoint or the calling endpoint's Gatekeeper. If the called endpoint's Gatekeeper wishes to use the Gatekeeper routed call model, it returns its own Call Signalling Channel Transport Address in the ARJ. The called endpoint then uses the Facility message to redirect the call to the called endpoint's Gatekeeper's Call Signalling Transport Address. These procedures assume that the calling endpoint or calling endpoint's Gatekeeper only knows the called endpoints Call Signalling Channel Transport Address. This address may have been received in an LCF sent in response to an LRQ requesting the address of the called endpoint or it may be known through out-of-band methods.

If the called endpoint's Gatekeeper desires a Gatekeeper routed call model, it may return its own Call Signalling Transport Address in the LCF. This will allow the calling endpoint or calling endpoints Gatekeeper to send the Setup message directly to the called endpoints Gatekeeper, thus eliminating the redirection process.

An example of this scenario is shown in Figure 40. In this example, both endpoints are registered to different Gatekeepers, and both Gatekeepers choose to route the call signalling (similar to the case in Figure 39). Endpoint 1 (calling endpoint) sends an ARQ (1) to Gatekeeper 1. Gatekeeper 1 multicasts an LRQ (2) to locate called Endpoint 2. Gatekeeper 2 returns an LCF (3) with the Call

Signalling Channel Transport Address of itself. Thus, Gatekeeper 1 will subsequently send a Setup (6) message to Gatekeeper 2's Call Signalling Channel Transport Address and Gatekeeper 2 will send a Setup (8) message to Endpoint 2. Endpoint 2 initiates the ARQ (9)/ACF (10) exchange with Gatekeeper 2. Endpoint 2 then responds to Gatekeeper 2 with the Connect (12) message which contains its H.245 Control Channel Transport Address for use in H.245 signalling. Gatekeeper 2 sends the Connect (13) message to Gatekeeper 1 which may contain the Endpoint 2 H.245 Control Channel Transport Address or a Gatekeeper 2 H.245 Control Channel Transport Address, based on whether the Gatekeeper 2 chooses to route the H.245 Control Channel or not. Gatekeeper 1 sends the Connect (14) message to Endpoint 1 which may contain the H.245 Control Channel Transport Address sent by Gatekeeper 2 or a Gatekeeper 1 H.245 Control Channel Transport Address, based on whether the Gatekeeper 1 chooses to route the H.245 Control Channel or not.

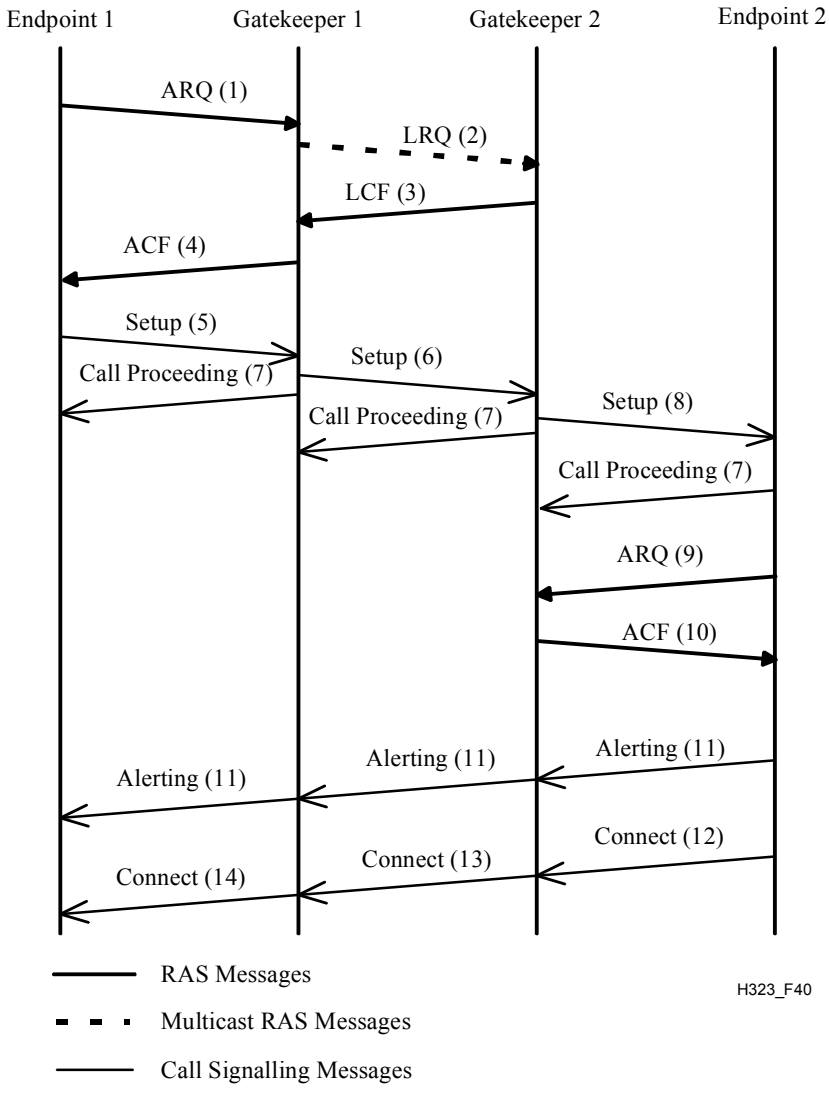


Figure 40/H.323 – Optional called endpoint signalling

8.1.7 Fast connect procedure

H.323 endpoints may establish media channels in a call using either the procedures defined in ITU-T Rec. H.245 or the "Fast Connect" procedure described in this clause. The Fast Connect procedure allows the endpoints to establish a basic point-to-point call with as few as one round-trip message exchange, enabling immediate media stream delivery upon call connection.

The calling endpoint initiates the Fast Connect procedure by sending a Setup message containing the **fastStart** element to the called endpoint. The **fastStart** element consists of a sequence of **OpenLogicalChannel** structures describing media channels which the calling endpoint proposes to send and receive, including all of the parameters necessary to immediately open and begin transferring media on the channels. Details of the content and usage of the **fastStart** element are discussed below.

The called endpoint may refuse to use the Fast Connect procedure, either because it does not implement it or because it intends to invoke features that require use of the procedures defined in ITU-T Rec. H.245. Refusal of the Fast Connect procedure is accomplished by either not returning the **fastStart** element or including the **fastConnectRefused** element in any H.225.0 call signalling message up to and including the Connect message. Note that an endpoint may omit **fastStart** elements in a message prior to Connect, but then later return **fastStart** element in the Connect message thereby accepting the Fast Connect procedure. Refusing the Fast Connect procedure (or not initiating it) requires that H.245 procedures be used for capabilities exchange and opening of media channels.

When the called endpoint desires to proceed with the Fast Connect procedure, it sends a H.225.0 call signalling message (Call Proceeding, Progress, Alerting, or Connect) containing a **fastStart** element selecting from amongst the **OpenLogicalChannel** proposals offered by the calling endpoint. The calling endpoint shall process each of these messages until it determines that Fast Connect is accepted or refused. Although the calling endpoint may receive the **fastStart** element in a Facility message sent by a Gatekeeper, the called endpoint shall not use the Facility message to send **fastStart**. Channels thus accepted are considered opened as though the usual H.245 **openLogicalChannel** and **openLogicalChannelAck** procedure had been followed. The called endpoint shall not include a **fastStart** element in any H.225.0 call signalling message sent after the Connect message and shall not include **fastStart** in any H.225.0 call signalling message unless the Setup message contained a **fastStart** element.

The called endpoint may begin transmitting media (according to the channels opened) immediately after sending a H.225.0 call signalling message containing **fastStart**. The calling endpoint must therefore be prepared to receive media on *any* of the receive channels it proposed in the Setup message, since it is possible for the media to be received prior to the H.225.0 call signalling message indicating precisely which channels will be used. Once a H.225.0 call signalling message containing **fastStart** is received by the calling endpoint, the calling endpoint may discontinue attempting to receive media on the channels for which proposals were not accepted by the called endpoint. Note that national requirements may prohibit called endpoints from transmitting media or limit the nature of the content of the media stream, prior to transmission of a Connect message; it is the responsibility of the endpoint to comply with applicable requirements. If the calling endpoint sets the **mediaWaitForConnect** element to TRUE in the Setup message, then the called endpoint shall not send any media until after the Connect message is sent.

The calling endpoint may begin transmitting media (according to the channels opened) immediately upon receiving a H.225.0 call signalling message containing **fastStart**. Thus, the called endpoint must be prepared to immediately receive media on the channels it accepted in the H.225.0 call signalling message containing **fastStart**. Note that national requirements may prohibit calling endpoints from transmitting media prior to receipt of a Connect message; it is the responsibility of the endpoint to comply with applicable requirements.

NOTE 1 – An entity shall not send an empty **fastStart** element in any message (i.e., a **fastStart** element shall contain at least one **OpenLogicalChannel** proposal). If an endpoint does receive a **fastStart** element that contains no **OpenLogicalChannel** proposals, it shall ignore the **fastStart** element.

NOTE 2 – When an endpoint or a gatekeeper intervening in call signalling receives a **fastStart** element in a Call Proceeding message, it will not be able to relay the Call Proceeding if the Call Proceeding message has already been sent to the originating side. In that case, the **fastStart** element in the Call Proceeding message shall be mapped to a **fastStart** element in a Facility message.

8.1.7.1 Proposal, selection and opening of media channels

The calling endpoint may propose multiple media channels or multiple alternative sets of characteristics for each media channel by encoding multiple **OpenLogicalChannel** structures within the **fastStart** element of the Setup message. Each **OpenLogicalChannel** structure within the **fastStart** element describes exactly one unidirectional media channel or one bidirectional media channel.

In the Setup message, each **OpenLogicalChannel** is a proposal to establish a media channel. **OpenLogicalChannel** proposals are included in the **fastStart** element in order of preference, with the most preferred alternatives listed first in the **fastStart** sequence; proposals to open audio channels shall be listed prior to channels for any other media types. In the H.225.0 call signalling message containing **fastStart** sent in response to Setup, each **OpenLogicalChannel** is an acceptance of a proposed media channel and indicates the channels that are established and can immediately be used for media transmission.

If an offered **dataType** element specifies encryption via the **h235Media** choice, the included **encryptionAuthenticationAndIntegrity** element may include an **encryptionCapability** element containing multiple encryption algorithms (including the NULL algorithm). This construct shall be taken to offer a choice of any one of the specified algorithms for encryption of the associated media capability.

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

In the Setup message, each **OpenLogicalChannel** that proposes a unidirectional channel for transmission from the calling endpoint to the called endpoint and that will carry media using RTP shall contain the **mediaControlChannel** element (indicating the reverse RTCP channel) in the **H2250LogicalChannelParameters** element of the **forwardLogicalChannelParameters** structure.

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

In the Setup message, each **OpenLogicalChannel** that proposes a unidirectional channel for transmission from the called endpoint to the calling endpoint and that will carry media using RTP shall contain the **mediaControlChannel** element (indicating the RTCP channel going in the same direction) in the **H2250LogicalChannelParameters** element of the **reverseLogicalChannelParameters** structure.

In an **OpenLogicalChannel** that proposes a bidirectional channel between the calling endpoint and the called endpoint, the **forwardLogicalChannelParameters** and **reverseLogicalChannelParameters** element shall contain parameters specifying the characteristics of the proposed channel. Each such **OpenLogicalChannel** structure shall have a unique **forwardLogicalChannelNumber** value. Alternative proposals for the same bidirectional channel shall contain the same **sessionID** value in **H2250LogicalChannelParameters**. The **mediaChannel** element shall be omitted in the proposal; it will be provided in the **reverseLogicalChannelParameters** element by the called endpoint should the proposal be accepted. The other **H2250LogicalChannelParameters** and **dataType** shall be set to correctly describe the transmit capabilities of the calling endpoint associated with this proposed channel.

All **mediaControlChannel** elements inserted by the calling endpoint for the same **sessionID** for both directions shall have the same value.

Upon receipt of a Setup message containing **fastStart**, determining that it is willing to proceed with the Fast Connect procedure, and reaching the point in the connection at which it is ready to begin media transmission, the called endpoint shall choose from amongst the proposed **OpenLogicalChannel** structures containing **reverseLogicalChannelParameters** elements for each media type it wants to transmit, from amongst the proposed **OpenLogicalChannel** structures specifying **forwardLogicalChannelParameters** (and omitting **reverseLogicalChannelParameters**) for each media type it wants to receive, and from amongst the proposed **OpenLogicalChannel** structures containing both **forwardLogicalChannelParameters** and **reverseLogicalChannelParameters** elements for each bidirectional channel it wants to transmit and or receive. If alternative proposals are presented, only one **OpenLogicalChannel** structure shall be selected from amongst each alternative set; alternatives within a set have the same **sessionID**. If multiple encryption algorithms are offered for a channel, the called endpoint must select one and modify the **OpenLogicalChannel** to remove the others. The called endpoint accepts a proposed channel by returning the corresponding **OpenLogicalChannel** structure in any H.225.0 call signalling message sent in response to Setup, up to and including Connect. A called endpoint may choose to repeat the **fastStart** element in all subsequent messages up to and including Connect: the contents of the **fastStart** element shall be the same. Calling endpoints shall react to the first **fastStart** element received in a response message to the Setup message and ignore any subsequent **fastStart** elements. The called endpoint may choose not to open media flow in a particular direction or of a particular media type by not including a corresponding **OpenLogicalChannel** structure in the **fastStart** element of the H.225.0 call signalling response.

When accepting a proposed channel for transmission from called endpoint to calling endpoint, the called endpoint shall return the corresponding **OpenLogicalChannel** structure to the calling endpoint, inserting a unique **forwardLogicalChannelNumber** into the **OpenLogicalChannel** structure and, for channels that will carry media using RTP, a valid **mediaControlChannel** element (indicating the reverse RTCP channel) in the **H2250LogicalChannelParameters** element of the **reverseLogicalChannelParameters** structure. The called endpoint may begin transmitting media on the accepted channel according to the parameters specified in **reverseLogicalChannelParameters** immediately after sending the H.225.0 call signalling response containing **fastStart**, unless **mediaWaitForConnect** was set to TRUE in which case it must wait until after sending the Connect message.

When accepting a proposed channel for transmission from the calling endpoint to the called endpoint, the called endpoint shall return the corresponding **OpenLogicalChannel** structure to the calling endpoint. The called endpoint shall insert valid a **mediaChannel** and, for channels that will carry media using RTP, a **mediaControlChannel** field (indicating the RTCP channel going in the same direction) in the **h2250LogicalChannelParameters** element of the **forwardLogicalChannelParameters** structure. All **mediaControlChannel** elements inserted by the called endpoint for the same **sessionID** for both directions shall have the same value. The called endpoint shall then prepare to immediately receive media flow according to the parameters specified in **forwardLogicalChannelParameters**. The calling endpoint may begin transmitting media on the accepted and opened channels upon receipt of the H.225.0 call signalling response containing **fastStart** and may release any resources allocated to reception on proposed channels that were not accepted.

When accepting a proposed bidirectional channel for transmission between the calling endpoint and the called endpoint, the called endpoint shall return the corresponding **OpenLogicalChannel** structure to the calling endpoint. The called and calling endpoints shall use the value in the **forwardLogicalChannelNumber** element as the logical channel number of the forward and reverse transmission paths of the bidirectional channel. The called endpoint shall insert a valid **mediaChannel** element in the **h2250LogicalChannelParameters** element of the **reverseLogicalChannelParameters** structure. The called and calling endpoints shall receive media flow according to the parameters specified in **forwardLogicalChannelParameters** and the **reverseLogicalChannelParameters**, respectively. The called endpoint shall be prepared to accept a connection for the bidirectional channel prior to returning the **fastStart** element. The calling endpoint may begin transmitting media on the accepted channels upon receipt of the H.225.0 call signalling response containing **fastStart** and may release any resources allocated for proposed channels that were not accepted.

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

8.1.7.2 Switching to H.245 procedures

After establishment of a call using the Fast Connect procedure, either endpoint may determine that it is necessary to invoke call features that require the use of H.245 procedures. Either endpoint may initiate the use of H.245 procedures at any point during the call using tunnelling as described in 8.2.1 (if **h245Tunnelling** remains enabled). An H.323 Version 4 or higher entity that uses Fast Connect in a call shall use H.245 tunnelling when an H.245 Control Channel is required and shall always set the **h245Tunnelling** field to TRUE. The process for switching to a separate H.245 connection is described in 8.2.3 and may be used by Version 3 or older entities or by newer H.323 entities when communicating with Version 3 or older entities for the purpose of maintaining backward compatibility.

When a call is established using the Fast Connect procedure, both endpoints shall keep the H.225.0 Call Signalling Channel open until either the call is terminated or, for compatibility with older endpoints, until a separate H.245 connection is established.

When H.245 procedures are activated, all mandatory procedures of H.245 that normally occur upon initiation of an H.245 connection shall be completed prior to initiation of any additional H.245 procedures. The media channels that were established in the Fast Connect procedure are "inherited" as though they had been opened using normal H.245 **openLogicalChannel** and **openLogicalChannelAck** procedures.

If the calling endpoint utilizes Fast Connect to initiate a call, it shall not open the H.245 Control Channel using the normal H.245 tunnelling or via a separate H.245 connection until the called endpoint has returned **fastStart**, **fastConnectRefused**, **h245Address**, or the Connect message. Note that older H.323 endpoints may open the H.245 Control Channel even before receiving one of these message elements or message, in spite of the fact that it initiated a Fast Connect call. While this behaviour was strongly discouraged in previous publications and is now forbidden, endpoints need to be aware of this older behaviour. If an endpoint opens the H.245 Control Channel before receiving the aforementioned message elements or message, the endpoint shall assume that Fast Connect is terminated and shall not send a **fastStart** element.

However, an endpoint may exchange the **terminalCapabilitySet** message and the **masterSlaveDetermination** message in the Setup message as described in 8.2.4. Such an exchange constitutes the opening of the H.245 Control Channel, but does not preclude either endpoint from proceeding with Fast Connect.

The called endpoint shall not initiate H.245 before returning **fastConnectRefused**, **fastStart**, or the Connect message. A called endpoint that returns the **h245Address** element in any message up to and including the Connect message, and which has not already explicitly accepted or rejected Fast Connect, shall also return either **fastStart** or **fastConnectRefused** in the same message. Note that older endpoints may not return **fastStart** or **fastConnectRefused**. For backward compatibility with older endpoints, H.323 endpoints may assume that Fast Connect is refused if the called endpoint sends the **h245Address** element or opens the H.245 Control Channel without simultaneously or previously sending **fastStart** or **fastConnectRefused**.

Note that in the case where a separate H.245 connection is opened from the called endpoint to the calling endpoint that supplied its **h245Address** in the Setup message, a race condition exists: the calling endpoint may detect the opening of the H.245 Control Channel from the called endpoint before it receives the **fastStart** element. For this reason, it is recommended that if an endpoint accepts Fast Connect and initiates a separate connection for H.245, it should introduce a delay between sending the H.225.0 message containing the **fastStart** element and the initiation of the separate H.245 connection. In the event that the called endpoint fails to introduce a delay, the calling endpoint should still be prepared for a possible late arrival of the **fastStart** element in this scenario. Older endpoints may assume that Fast Connect is refused if the H.245 Control Channel is opened prior to receiving the **fastStart** element.

8.1.7.3 Terminating a call

If a call connected using the Fast Connect procedure continues to completion without initiation of H.245 procedures, then the call may be terminated by either endpoint sending a H.225.0 call signalling Release Complete message. If H.245 procedures are initiated during the call, then the call is terminated as described in 8.5.

If a separate H.245 connection has not been established and the H.225.0 Call Signalling Channel is terminated, the call shall also be terminated.

8.1.7.4 In-band and out-of-band tones and announcements

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

On completing call setup, the endpoint on the terminating side shall decide if it will provide in-band tones or if locally generated tones at the originating side shall be used. Note that other type of indication can replace locally generated tones and announcement in some systems (visual indications on a screen for example). For the purpose of this clause, they will be referred to as locally generated tones and announcements. Locally generated tones, provided at the originating side, are the default. The terminating side may wish to provide in-band-generated tones and announcements, for example when the terminating endpoint is a gateway to an analogue network.

To instruct the originating side not to generate locally generated tones, such as ringback or busy, the terminating side shall open the media channel by responding to the Fast Connect request and sending a Progress indicator information element with progress descriptor #1, *Call is not end-to-end ISDN; further call progress information may be available in-band*, or #8, *In-band information or an appropriate pattern is now available* in a Call Proceeding, Progress or Alerting message, or in a Connect message if an Alerting message was not sent. The response to the Fast Connect message shall be done before or at the same time the Progress indicator is sent (i.e., up to and including the same message the Progress indicator is sent). The terminating side can provide in-band tones or announcements (such as ringback or busy) as soon as the progress descriptor has been sent and the media channel has been opened. Note that the Progress indicator should be in an Alerting message only if the endpoint is being alerted. If another in-band tone, such as busy or re-order tone is provided, the Progress indicator should not be in an Alerting. When no appropriate call setup message is available, a Progress message can be used to carry the Progress indicator.

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

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

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

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

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

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

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

8.1.8 Call setup via gateways

8.1.8.1 Gateway in-bound call setup

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

A Gateway which cannot directly route an incoming SCN call to an H.323 endpoint shall be able to accept two-stage dialling. For Gateways to H.320 networks (also H.321, H.322 and H.310 in H.321 mode), the Gateway shall accept SBE numbers from the H.320 terminal. Optionally, Gateways to H.320 networks may support the TCS-4 and IIS BAS codes to retrieve the H.323 dialling information after a H.320 call has been established. For Gateways to H.310 native mode and H.324 networks, the Gateway shall accept H.245 **userInputIndication** messages from the H.324 terminal. In these two cases, support of DTMF is optional. For Gateways to speech-only

terminals, the Gateway shall accept DTMF numbers from the speech-only terminal. These numbers will indicate a second stage dialling number to access the individual endpoint on the network.

8.1.8.2 Gateway out-bound call setup

When a network endpoint calls an external terminal via the Gateway, call setup between the network endpoint and the Gateway proceeds the same as the endpoint-to-endpoint call setup. The Gateway will receive the destination **dialledDigits** or **partyNumber** (**e164Number** or **privateNumber**) in the Setup message. It will then use this address to place the out-bound call. The Gateway may return Call Proceeding messages to the network endpoint while establishing the outgoing call.

A Gateway should send a Call Proceeding message after it receives the Setup message (or after it receives ACF) if it expects more than 4 seconds to elapse before it can respond with Alerting, Connect, or Release Complete.

The Progress Indicator information element is used to indicate that inter-networking is occurring. The Gateway shall issue a Progress indicator information element within the Alerting, Call Proceeding or Connect messages. This information may also be sent in a Progress message.

The network endpoint shall send all **dialledDigits** or **partyNumber** addresses that it is calling in the Setup message. For example, a six B-channel call on the ISDN will require six **dialledDigits** or **partyNumber** addresses in the Setup message. The Gateway shall respond to the Setup message with a Connect or Release Complete message as well as Alerting, Call Proceeding, or Progress messages. Failure of the SCN call shall be reported to the network endpoint in the Release Complete message. The use of multiple CRV values and multiple Setup messages is for further study. Addition of channels on the SCN during a call is for further study.

A network endpoint that is registered with a Gatekeeper should request sufficient call bandwidth in the ARQ message for the aggregate of all SCN calls. If sufficient call bandwidth was not requested in the ARQ message, the procedures of 8.4.1, Bandwidth Changes, shall be followed in order to obtain additional call bandwidth.

The Gateway may advance to Phase B after placing the first call on the SCN. Additional calls for the additional SCN **dialledDigits** or **partyNumber** numbers may be placed after the capability exchange with the Gateway and establishment of audio communications with the SCN endpoint.

8.1.9 Call setup with an MCU

For Centralized Multipoint Conferences, all endpoints exchange call signalling with the MCU. Call setup between an endpoint and the MCU proceeds the same as the endpoint-to-endpoint call setup scenarios of 8.1.1 through 8.1.5. The MCU may be the called endpoint or the calling endpoint.

In a Centralized Multipoint Conference, the H.245 Control Channel is opened between the endpoints and the MC within the MCU. The audio, video, and data channels are opened between the endpoints and the MP within the MCU. In a Decentralized Multipoint Conference, the H.245 Control Channel is open between the endpoint and the MC (there may be many such H.245 Control Channels, one for each call). The Audio and Video Channels should be multicast to all endpoints in the conference. The Data Channel shall be opened with the Data MP.

In an ad hoc Multipoint Conference where the endpoints do not contain an MC and the Gatekeeper would like to provide an Ad Hoc multipoint service for the endpoints, the H.245 Control Channel may be routed through the Gatekeeper. Initially, the H.245 Control Channel would be routed between the endpoints through the Gatekeeper. When the conference switches to multipoint, the Gatekeeper may connect the endpoints to an MC associated with the Gatekeeper.

In an ad hoc Multipoint Conference where one or both of the endpoints contains an MC, the normal call setup procedures defined in 8.1.1 through 8.1.5 are used. These procedures may apply even if an endpoint that contains an MC is actually a MCU. The master-slave determination procedure is used to determine which MC will be the Active MC for the conference.

8.1.10 Call forwarding

An endpoint wishing to forward a call to another endpoint may issue a Facility message indicating the address of the new endpoint. The endpoint receiving this Facility indication should send a Release Complete and then restart the Phase A procedures with the new endpoint.

8.1.11 Broadcast call setup

Call setup for loosely controlled Broadcast and Broadcast Panel conferences shall follow the procedures defined in ITU-T Rec. H.332.

8.1.12 Overlapped sending

H.323 entities can optionally support overlap sending. If a Gatekeeper is present, and overlap sending is being used, endpoints should send an ARQ message to the Gatekeeper each time some new addressing information is input. The endpoint shall place the total cumulative addressing information into the **destinationInfo** field each time an ARQ message is sent. If there is insufficient addressing information in the ARQ, the Gatekeeper should respond with an ARJ with the **reason** set to **incompleteAddress**. This indicates that the endpoint should send another ARQ when more addressing information is available. When a Gatekeeper has sufficient addressing information to assign a suitable **destCallSignalAddress**, it shall return an ACF. Note that this does not necessarily mean that the addressing information is complete. If the Gatekeeper sends an ARJ with **AdmissionRejectReason** set to something other than **incompleteAddress**, the call setup process shall be aborted.

When an endpoint has a suitable **destCallSignalAddress**, it shall send a Setup message with the **canOverlapSend** field assigned according to whether it is capable of supporting the overlap sending procedures. If a remote entity receives a Setup message with an incomplete address and the **canOverlapSend** field set to TRUE, it should initiate overlap sending procedures by returning the Setup Acknowledge message. Additional addressing information should be sent using Information messages. If the address is incomplete and the **canOverlapSend** field set to FALSE, the remote entity should send Release Complete. Note that Gateways should not transfer Setup Acknowledge messages from the SCN to H.323 endpoints that have not indicated that they can support overlap sending procedures as the desired result may not be achieved.

8.1.13 Call setup to conference alias

Alias addresses (see 7.1.3) may be used to represent a conference at an MC. The procedures in the preceding subclauses apply, except as noted here.

8.1.13.1 Joining to a conference alias, with no gatekeeper

Endpoint 1 (calling endpoint) sends the Setup (1) message (see Figure 29) to the well-known Call Signalling Channel TSAP Identifier of Endpoint 2 (the MC). The Setup message includes the following fields:

```
destinationAddress          = conferenceAlias
destCallSignalAddress       = MC(U) transport address
conferenceID                = 0 (since the CID is unknown)
conferenceGoal              = join
```

Endpoint 2 responds with the Connect (4) message, which contains:

```
h245Address                 = Transport Address for H.245 signalling
conferenceID                = CID for the conference
```

8.1.13.2 Joining to a conference alias, with gatekeeper

Endpoint 1 (calling endpoint) initiates the ARQ (1)/ACF (2) exchange (reference Figure 30) with the Gatekeeper. The ARQ contains:

```
destinationInfo             = conferenceAlias
callIdentifier              = some value N
conferenceID                = 0 (since the CID is unknown)
```

The Gatekeeper shall return the Call Signalling Channel Transport Address of Endpoint 2 (called endpoint, containing the MC) in the ACF. Endpoint 1 then sends the Setup (3) message to Endpoint 2 using that Transport Address and the following fields:

```
destinationAddress          = conferenceAlias
destCallSignalAddress       = address supplied by ACF
conferenceID                = 0
conferenceGoal              = join
```

Ultimately, Endpoint 2 returns a Connect message with following fields:

```
h245Address                 = Transport Address for H.245 signalling
conferenceID                = CID for the conference
```

Endpoint 1 completes the call by informing its Gatekeeper of the correct CID. Endpoint 1 sends an IRR to the Gatekeeper with the following fields:

```
callIdentifier              = same value N as used in the first ARQ
conferenceID                = original CID from endpoint 1
substituteConferenceIDs     = CID from endpoint 2
```

8.1.13.3 Create or invite with a conference alias

Endpoint 1 (calling endpoint) may send a Setup message to Endpoint 2. The Setup message includes the following fields:

```
destinationAddress          = conferenceAlias
destCallSignalAddress       = MC(U) transport address
conferenceID                = CID of the conference
conferenceGoal              = create or invite
```

Endpoint 2 responds with the Connect message, which contains:

```
h245Address                 = Transport Address for H.245 signalling
conferenceID                = CID for the conference
```

8.1.13.4 Consideration for version 1 endpoints

When an H.323 entity (endpoint or MCU) receives a Setup message from a Version 1 entity and the **destinationAddress** matches one of its conferences aliases, then it shall ignore the **conferenceGoal** and treat the Setup request as a join request.

When a Gatekeeper receives an ARQ, from a Version 1 entity and the **destinationInfo** matches one of its conferences aliases, then it shall ignore the **conferenceID** field. Likewise, when an H.323 entity receives a Setup message from a Version 1 entity and the **destinationAddress** matches one of its conferences aliases, then it shall ignore the **conferenceID**.

These provisions allow a Version 1 endpoint to call a conference Alias.

8.1.14 Gatekeeper modification of destination addresses

An endpoint shall set the **canMapAlias** field to TRUE to indicate its ability to accept modified destination information from a Gatekeeper. The endpoint shall use the destination information returned in ACF or LCF instead of the destination information passed in the ARQ or LRQ. For an ingress Gateway, the destination information that appears in the ACF will be used in the Setup message that is sent on the packet network. For an egress Gateway, the destination information that appears in the ACF will be used to address a destination in the GSTN (for example, appearing in the Setup message sent to the ISDN).

In Gatekeeper routed cases, the Gatekeeper may modify destination addresses in the Setup message it receives before sending out a corresponding Setup message.

NOTE – H.323 systems prior to Version 4 were not required to set the **canMapAlias** field to TRUE.

8.1.15 Indicating desired protocols

When an endpoint places a call, it may indicate in various H.225.0 messages those protocols it desires to utilize during the course of a call, such as fax, H.320, T.120, etc., in the **desiredProtocols** field. If the endpoint provides a list of desired protocols to its Gatekeeper or if an entity sends an LRQ message to a Gatekeeper with a list of desired protocols, the Gatekeeper should attempt to locate an endpoint that can provide support for the desired protocols. If the Gatekeeper finds no endpoint that supports any of the desired protocols, the Gatekeeper shall still resolve the address so that the call may continue.

The calling endpoint may examine the **EndpointType** of the destination endpoint to determine exactly what protocols the remote endpoint possesses.

8.1.16 Gatekeeper requested tones and announcements

A Gatekeeper may request a Gateway to play a tone or an announcement for a variety of call events. These call events could be "pre-call" events (something that happens before the terminating gateway is signalled, such as prompting the caller for a destination number or an account code), "mid-call" events (something that happens in the middle of a call, such as providing an announcement to alert the parties on the call that the call will end in a few minutes), or "end-call" events (something that happens at the end of the call, such as a farewell message). In all cases, the Gatekeeper may use a **H248SignalsDescriptor** to describe the prompt the gateway should use.

The following pre-call events are supported:

- Prompting for a destination – In what is often called two-stage dialling, the caller dials one number to reach the Gateway and is prompted to dial the true destination number. Although a Gateway may have a general policy to always provide the prompt, in some circumstances it may make sense to allow the Gateway to consult the Gatekeeper. This "consult" operation is simply the ARQ with the called number as **destinationInfo**. If the Gatekeeper decides that a true destination number is required, the Gatekeeper may instruct the Gateway to prompt the caller, collect the additional digits, and consult the Gatekeeper with the destination. The Gatekeeper uses the ARJ with a **serviceControl** element and an **AdmissionRejectReason** of **collectDestination**. The **serviceControl** element has a **ServiceControlDescriptor** of type **signal** (which contains the **H248SignalsDescriptor**) and a **reason** of **open**. The **AdmissionRejectReason** of **collectDestination** instructs the Gateway to place the collected true destination into the **destinationInfo** of a new ARQ.

- Prompting for an authorization code, account code, or PIN – In this case, the Gatekeeper replies to the ARQ with an ARJ containing a **serviceControl** element and an **AdmissionRejectReason** of **collectPIN**. The **serviceControl** element has a **ServiceControlDescriptor** of type **signal** (which contains the **H248SignalsDescriptor**) and a reason of open. The **AdmissionRejectReason** of **collectPIN** instructs the Gateway to place the collected PIN (or authorization code or account code) into a token or **cryptoToken** of a new ARQ.
- Prompting for both a destination and a PIN – This is simply a serial operation of the first two cases.
- Rejecting a call – A Gatekeeper may choose to reject a call, but provide some feedback to the user (for example, providing a network busy tone or announcement if there are no available facilities for a destination). In this case, the ARJ would contain an **AdmissionRejectReason** that reflects the condition, but not **collectPIN** or **collectDestination**.

A Gatekeeper may initiate a mid-call signal by using the SCI message. The **serviceControl** element has a **ServiceControlDescriptor** of type **signal** (which contains the H.248 **H248SignalsDescriptor**) and a **reason** of **open**. The signal may be stopped by sending the **ServiceControlIndication** message, but with a **ServiceControlDescriptor** containing a **reason** of **close**. A Gateway should respond to the SCI message with a SCR with an appropriate **result**.

A Gatekeeper may initiate an end-call signal in a DRQ (for the direct endpoint routing case) or a Release Complete (for the Gatekeeper routed case) with a **serviceControl** element. The **serviceControl** element has a **ServiceControlDescriptor** of type **signal** (which contains the H.248 **H248SignalsDescriptor**) and a **reason** of **open**. The signal may be stopped by sending the **ServiceControlIndication** message, but with a **ServiceControlDescriptor** containing a **reason** of **close**.

8.2 Phase B – Initial communication and capability exchange

Once both sides have exchanged call setup messages from Phase A, the endpoints shall, if they plan to use H.245, establish the H.245 Control Channel. The procedures of ITU-T Rec. H.245 are used over the H.245 Control Channel for the capability exchange and to open the media channels.

NOTE – Optionally, the H.245 Control Channel may be set up by the called endpoint on receipt of Setup and by the calling endpoint on receipt of Alerting or Call Proceeding. In the event that Connect does not arrive or an endpoint sends Release Complete, the H.245 Control Channel shall be closed.

Endpoints shall support the capabilities exchange procedure of H.245 as described in 6.2.8.1.

Endpoint system capabilities are exchanged by transmission of the H.245 **terminalCapabilitySet** message. This capability message shall be the first H.245 message sent unless the endpoint is indicating that it understands the **parallelH245Control** field (see 8.2.4). If prior to successful completion of terminal capability exchange, any other procedure fails (i.e., rejected, not understood, not supported), then the initiating endpoint should initiate and successfully complete terminal capability exchange before attempting any other procedure. An endpoint which receives a **terminalCapabilitySet** message from a peer prior to initiating capabilities exchange shall respond as required by 6.2.8.1 and should initiate and successfully complete capabilities exchange with that peer prior to initiating any other procedure.

Endpoints shall support the master-slave determination procedure of H.245 as described in 6.2.8.4. In cases where both endpoints in a call have MC capability, the master-slave determination is used for determining which MC will be the Active MC for the conference. The Active MC may then send the **mcLocationIndication** message. The procedure also provides master-slave determination for opening bidirectional channels for data.

Master-slave determination shall be advanced (by sending either **MasterSlaveDetermination** or **MasterSlaveDeterminationAck** as appropriate) in the first H.245 message after Terminal Capability Exchange has been initiated.

If the initial capability exchange or master-slave determination procedures fail, these should be retried at least two additional times before the endpoint abandons the connection attempt and proceeds to Phase E.

Following successful completion of the requirements of Phase B, the endpoints shall proceed directly to the desired operating mode, normally Phase C.

8.2.1 Encapsulation of H.245 messages within H.225.0 Call Signalling messages

In order to conserve resources, synchronize call signalling and control, and reduce call setup time, it may be desirable to convey H.245 messages within the H.225.0 call signalling Call Signalling Channel instead of establishing a separate H.245 channel. This process, known as "encapsulation" or "tunnelling" of H.245 messages, is accomplished by utilizing the **h245Control** element of **h323-uu-pdu** on the Call Signalling Channel, copying an encoded H.245 message as an octet string.

When tunnelling is active, one or more H.245 messages can be encapsulated in any H.225.0 call signalling message. If tunnelling is being utilized and there is no need for transmission of a H.225.0 call signalling message at the time an H.245 message must be transmitted, then a Facility message shall be sent with the **reason** set to **transportedInformation**. (Note that H.323 systems prior to version 4 used a Facility message with **h323-message-body** set to **empty**.)

A calling entity capable of and willing to use H.245 encapsulation shall set the **h245Tunnelling** element to TRUE in the Setup message and any subsequent H.225.0 call signalling messages it sends so long as it desires tunnelling to remain active. A called entity capable of and willing to use H.245 encapsulation shall set the **h245Tunnelling** element to TRUE in the first H.225.0 call signalling message sent in response to Setup and in every subsequent H.225.0 call signalling message it sends so long as it desires tunnelling to remain active. The called entity shall not set **h245Tunnelling** to TRUE in any H.225.0 call signalling response (and tunnelling remains disabled) unless it was TRUE in the Setup message to which it is responding. If the called entity does not yet know if H.245 tunnelling can be supported, it shall include the **provisionalRespToH245Tunnelling** flag. This may happen, for example, when a Gatekeeper is responding to a calling entity with a message such as Call Proceeding before the called endpoint responds to the **h245Tunnelling** flag. The **provisionalRespToH245Tunnelling** flag effectively eliminates the meaning of the **h245Tunnelling** flag in a message and the flag shall thus be ignored by the receiving endpoint.

If **h245Tunnelling** is not set to TRUE in any H.225.0 call signalling message that does not include the **provisionalRespToH245Tunnelling** flag, then tunnelling is disabled from that point for the duration of the call and a separate H.245 connection shall be established when and if H.245 procedures are invoked.

The calling entity may include tunnelled H.245 messages in the Setup message; it must also set the **h245Tunnelling** element to TRUE. If the called entity does not set **h245Tunnelling** to TRUE and the **provisionalRespToH245Tunnelling** flag is absent in the first H.225.0 call signalling message sent in response to Setup, then the calling entity shall assume that the H.245 messages it had encapsulated in Setup were ignored by the called entity and repeat them, as necessary, after the separate H.245 channel is established. The called entity, if it sets **h245Tunnelling** to TRUE, may also include encapsulated H.245 messages in the first and subsequent H.225.0 call signalling messages.

The calling endpoint shall not include both a **fastStart** element and encapsulated H.245 messages in **h245Control** in the same Setup message, since the presence of the encapsulated H.245 messages would override the Fast Connect procedure. A calling endpoint may, however, include both a **fastStart** element and set **h245Tunnelling** to TRUE within the same Setup message; likewise, a called endpoint may include **fastStart** and set **h245Tunnelling** to TRUE within the same H.225.0 call signalling response. In this case, the Fast Connect procedures are followed, and the H.245 connection remains "unestablished" until actual transmission of the first tunnelled H.245 message or opening of the separate H.245 connection.

When H.245 encapsulation is being used, both endpoints shall keep the H.225.0 Call Signalling Channel open until either the call is terminated or a separate H.245 connection is established.

When an endpoint receives an **h245control** element encapsulating more than one H.245 PDU, the encapsulated H.245 PDUs shall be processed (i.e., provided to higher layers) sequentially by order of increasing offset from the beginning of the H.225.0 message.

H.323 Version 4 and higher entities shall indicate support for H.245 tunnelling as described in this clause by setting the **h245Tunnelling** field to TRUE in all messages containing this field.

8.2.2 Tunnelling through intermediate signalling entities

Entities in the signalling path such as Gatekeepers may perform functions such as divert on no-reply or other advanced call control that results in representing to an endpoint a call state that is different from the actual call state at the other endpoint. Such intermediate entities shall ensure that H.245 messages encapsulated in H.225.0 call signalling messages are forwarded to the other endpoint even if the H.225.0 call signalling message in which the H.245 message is encapsulated would be consumed and not forwarded to the other endpoint. This is accomplished by transferring the encapsulated H.245 message into a Facility message with the **reason** set to **transportedInformation**. (Note that H.323 systems prior to version 4 used the Facility message with the **h323-message-body** set to **empty**.) For example, if a Gatekeeper has already sent a Connect message to a calling endpoint and later receives a Connect message from a called endpoint that contains an encapsulated H.245 message, it must forward the H.245 message using a Facility message.

Entities in the signalling path shall also use the Facility message or the Progress message to convey any new information (such as Q.931 information elements, CallProceeding-UUIE fields, tunnelled non-H.323 protocols, and encapsulated H.245 messages) received in a Call Proceeding message to the other endpoint if the entity has already sent a Call Proceeding message. This will allow the entity, for example, to transmit the **fastStart** element to facilitate proper establishment of a Fast Connect call and/or a Progress Indicator to indicate the presence of in-band tones and announcements. When using the Facility message to carrying such information extracted from the Call Proceeding message, the **reason** in the Facility should be set to **forwardedElements**.

8.2.3 Switching to a separate H.245 connection

When H.245 encapsulation or Fast Connect is being used, either endpoint may choose to switch to using the separate H.245 connection at any time. In order to facilitate initiation of the separate H.245 connection by either endpoint, each endpoint may include **h245Address** in any H.225.0 call signalling message it sends during the call. If, at the time an endpoint deems it necessary to initiate the separate H.245 connection, it finds that it has not yet received the **h245Address** of the other endpoint, the endpoint shall transmit a Facility message with a **FacilityReason** of **startH245** and provide its H.245 address in the **h245Address** element. An endpoint receiving a Facility message with a **facilityReason** of **startH245** which has not already independently initiated the separate H.245 channel shall open the H.245 channel using the **h245Address** specified. Use of the separate H.245 connection is initiated by opening the H.245 TCP connection and accepted by acknowledgement of the H.245 TCP connection.

If tunnelling was being used, the endpoint initiating the separate H.245 connection shall not send any further tunnelled H.245 messages on the Call Signalling Channel and shall send no H.245 messages on the separate H.245 connection until the establishment of the TCP connection is acknowledged. The endpoint acknowledging opening of the separate H.245 connection shall not send any further tunnelled H.245 messages on the Call Signalling Channel after acknowledging the opening of the separate H.245 connection. Because of the possibility that H.245 messages have already been sent and are in transit when the separate H.245 channel is initiated, endpoints shall continue to receive and correctly process tunnelled H.245 messages until a H.225.0 call signalling message is received with the **h245Tunnelling** flag set to FALSE; responses to such "late" tunnelled H.245 messages or acknowledgement of such messages shall be sent on the separate H.245 connection after it is established. Once a separate H.245 connection has been established, it is not possible to switch back to using tunnelling.

In the event that both endpoints simultaneously initiate the separate H.245 connection, the endpoint with the numerically smaller **h245Address** shall close the TCP connection it opened and use the connection opened by the other endpoint. For purposes of comparing the numeric values of **h245Address**, each octet of the address shall be individually compared beginning with the first octet of the OCTET STRING and continuing through the OCTET STRING left to right until unequal numeric octet values are found. Comparison shall first be performed on the network-layer address element of **h245Address** and, if found to be equal, then on the transport (port) address element.

8.2.4 Initiating H.245 tunnelling in parallel with fast connect

As detailed in 8.2, the first two H.245 messages sent by an endpoint on the H.245 Control Channel are the **terminalCapabilitySet** message and the **masterSlaveDetermination** message. Even when Fast Connect is being utilized, there are advantages to exchanging those messages as quickly as possible. In particular, an entity may need to know as early as possible whether DTMF is supported in **UserInputIndication** or with RTP payload types (as described in 10.5) by the other entity. Additionally, if Fast Connect is refused, there are obvious advantages to having already transmitted these messages, as there are fewer messages to exchange in order to open logical channels.

Therefore, to expedite the exchange of capabilities and overall call setup, an entity may include the H.245 **terminalCapabilitySet** message and the **masterSlaveDetermination** message in the Setup message by including those messages in the **parallelH245Control** field of the Setup message. Unlike the **h245Control** field, the calling entity may send these messages in the Setup message along with the **fastStart** element. The calling entity shall set the **h245Tunnelling** field to TRUE when including the **parallelH245Control** field.

NOTE – A calling entity should not include the **parallelH245Control** field without also including the **fastStart** field, since H.245 tunnelling in the context of a call that does not utilize the Fast Connect procedures should be handled according to 8.2.1.

To indicate that the called entity understands the **parallelH245Control** field, the first H.245 message that the called entity sends shall be the **terminalCapabilitySetAck** message tunnelled in the H.225.0 Call Signalling Channel. This response message should be sent by the called entity at the same time that **fastConnectRefused** or **fastStart** is sent to the calling entity. Note that if an endpoint does not indicate that it understands the **parallelH245Control** field, it shall abide by 8.2 and send **terminalCapabiltySet** and not **terminalCapabilitySetAck** as the first H.245 message. The called entity shall set the **h245Tunnelling** field to TRUE if it understands the **parallelH245Control** field. Figure 41 shows the message exchanges of a Fast Connect call between two endpoints that understand the **parallelH245Control** field.

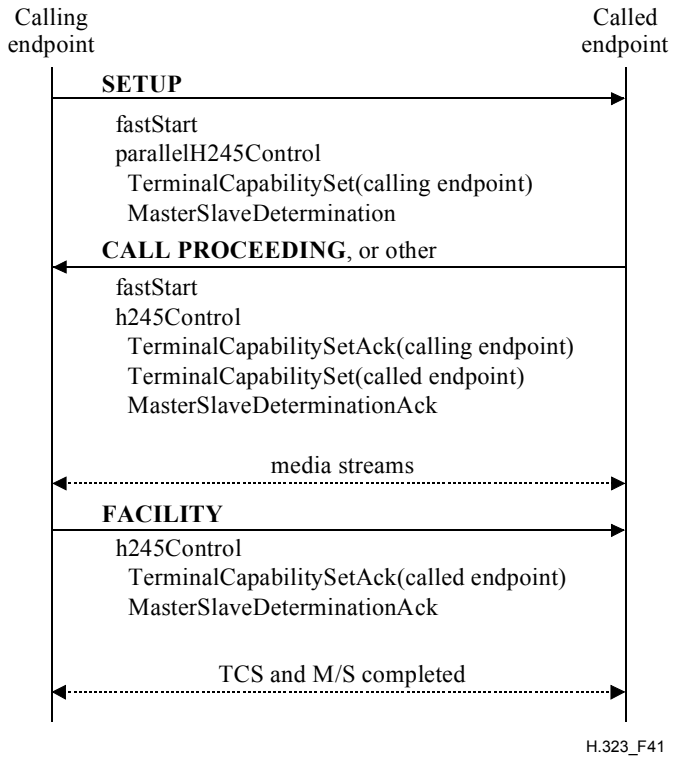


Figure 41/H.323 – Successful initiation of H.245 in parallel with fast connect

The calling entity shall recognize that the **parallelH245Control** field was not understood when either it receives a Connect message and still has not received a response to the initial **terminalCapabilitySet** message, the first H.245 message received from the called entity is not a tunnelled **terminalCapabilitySetAck** message, or **fastStart** or **fastConnectRefused** is received and no response has been received for the **terminalCapabilitySet** message. Figure 42 shows a message exchange between an endpoint that sends the **parallelH245Control** field and a called endpoint that does not understand the field.

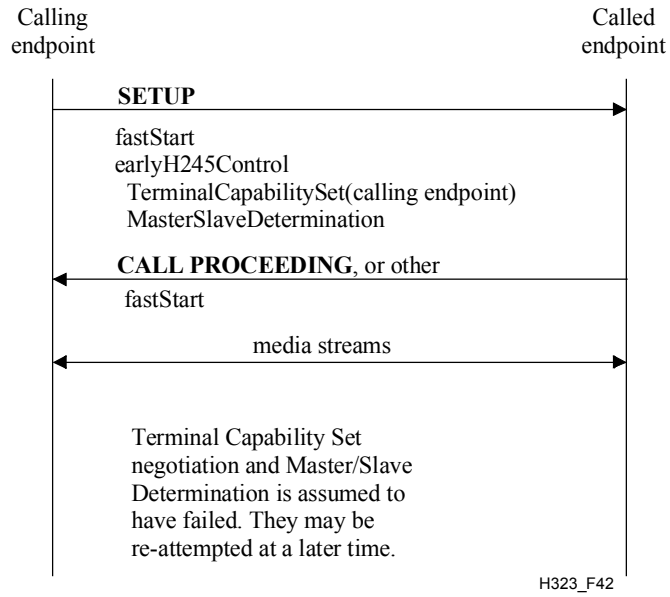


Figure 42/H.323 – Unsuccessful initiation of H.245 in parallel with fast connect

8.3 Phase C – Establishment of audiovisual communication

Following the exchange of capabilities and master-slave determination, the procedures of ITU-T Rec. H.245 shall then be used to open logical channels for the various information streams. The audio and video streams, which are transmitted in the logical channels setup in H.245, are transported over dynamic TSAP Identifiers using an unreliable protocol (see ITU-T Rec. H.225.0). Data communications which is transmitted in the logical channels setup in H.245, are transported using a reliable protocol (see ITU-T Rec. H.225.0).

The **openLogicalChannelAck** message returns, or the **reverseLogicalChannelParameters** of the **openLogicalChannel** request contains, the Transport Address that the receiving endpoint has assigned to that logical channel. The transmitting channel shall send the information stream associated with the logical channel to that Transport Address.

Following the opening of logical channels for audio and video, one **h2250MaximumSkewIndication** message shall be sent by the transmitter for each associated audio and video pair.

8.3.1 Mode changes

During a session, the procedures for changing channel structure, capability, receive mode, etc., shall be carried out as defined in ITU-T Rec. H.245. Appendix V/H.245 contains a procedure for changing modes on a logical channel which may minimize the disruption of the audio.

8.3.2 Exchange of video by mutual agreement

The indication **videoIndicateReadyToActivate** is defined in ITU-T Rec. H.245. Its use is optional, but when used the procedure shall be as follows.

Endpoint 1 has been set so that video is not transmitted unless and until Endpoint 2 has also indicated readiness to transmit video. Endpoint 1 shall send the indication **videoIndicateReadyToActivate** when the initial capability exchange has been completed, but shall not transmit a video signal until it has received either **videoIndicateReadyToActivate** or incoming video from Endpoint 2.

An endpoint which has not been set in this optional way is not obliged to wait until receipt of **videoIndicateReadyToActivate** or video before initiating its video transmission.

8.3.3 Media stream address distribution

In unicast, the endpoint shall open logical channels to the MCU or other endpoint. Addresses are passed in the **openLogicalChannel** and **openLogicalChannelAck**.

In multicast, the multicast addresses are assigned by the MC and distributed to the endpoints in the **communicationModeCommand**. It is the responsibility of the MC to allocate and assign unique multicast addresses. The endpoint shall signal an **openLogicalChannel** to the MC with the assigned multicast address. The MC shall forward the **openLogicalChannel** to each receiving endpoint. In cases where media from multiple endpoints are transmitted on a single session (e.g., single multicast address), the MC shall open a logical channel to each endpoint receiving media from an endpoint in the conference.

In cases where an endpoint joins a conference after the initial **communicationModeCommand** has been transmitted, it is the responsibility of the MC to send an updated **communicationModeCommand** to the new endpoint and to open the appropriate logical channels for media sourced from the new endpoint. In cases where an endpoint leaves the conference after the initial **communicationModeCommand** has been transmitted, it is the responsibility of the MC to close the appropriate logical channels which were being sourced from the endpoint which left the conference.

In multi-unicast, the endpoint must open logical channels to each of the other endpoints. The **openLogicalChannel** is sent to the MC and shall contain the terminal number of the endpoint for which the channel is intended. The endpoint can match a **openLogicalChannelAck** by the **forwardLogicalChannelNumber**.

8.3.4 Correlation of media streams in multipoint conferences

The following method shall be used to associate a logical channel with an RTP stream within a multipoint conference. The media stream source endpoint sends the **openLogicalChannel** message to the MC. In cases where the source would like to indicate a destination for the **openLogicalChannel**, the source endpoint should place the **terminalLabel** of the destination endpoint in the destination field of the **h2250LogicalChannelParameters**. The source endpoint shall also place its own **terminalLabel** in the **source** field of **h2250LogicalChannelParameters**. Note that in the multicast model, the absence of a **destination** indicates that the stream is applicable to all endpoints.

If a source endpoint has been assigned a **terminalLabel** by an MC, the source endpoint shall use an SSRC that contains the lowest byte of its **terminalLabel** as the lowest byte of its SSRC.

The destination endpoint may associate the logical channel number with the RTP stream source by comparing the **openLogicalChannel.h2250LogicalChannelParameters.source** field with the lowest byte of the SSRC in the RTP header.

It is possible for SSRC collisions when an H.323 endpoint is in an H.332 conference. The endpoint detecting the collision shall follow the procedures in RTP for SSRC collision resolution.

8.3.5 Communication mode command procedures

The H.245 **communicationModeCommand** is sent by an H.323 MC to specify the communication mode for each media type: unicast or multicast. This command may cause a switch between a centralized and decentralized conference and therefore may involve closing all existing logical channels and opening new ones.

The **communicationModeCommand** specifies all the sessions in the conference. For each session, the following data are specified: the RTP session identifier, the associated RTP session ID if applicable, a terminal label if applicable, a description of the session, the **dataType** of the sessions (e.g., G.711), and a unicast or multicast address for the media and media control channels as appropriate for the conference configuration and type.

The **communicationModeCommand** conveys the transmit modes which conference endpoints are to use in a conference. The command does not convey receive modes, as they are specified by **openLogicalChannel** commands which are sent from the MC to the endpoints.

It is presumed that the **communicationModeCommand** is defining the modes of a conference and is therefore sent after the **multipointConference** indication which notifies an endpoint that it must comply with the commands of the MC. Endpoints should wait for a **communicationModeCommand** before opening logical channels when they have received a **multipointConference** indication.

Endpoints receiving a **communicationModeCommand** use the **terminalLabel** field of each table entry to determine if the entry is applicable for its own processing. Entries which do not contain a **terminalLabel** apply to all endpoints in the conference. Entries which contain **terminalLabels** are commands to specific endpoints which match the **terminalLabel** in the entry. For example, when audio streams from all endpoints are placed on one multicast address (one session), the table entry for the audio mode, media address, and media control address will not contain a **terminalLabel**. When the table entry commands an endpoint to send its video to a multicast address, the MC will include that endpoint's **terminalLabel**.

The **communicationModeCommand** can be used to instruct endpoints in a conference (or a point-to-point call) to change modes by indicating a new mode for a **mediaChannel** that is already in use. It can also be used to tell an endpoint to transmit the media stream to a new address by indicating the mode currently in use, but with new **mediaChannel**. Similarly, an endpoint that receives a **communicationModeCommand** indicating the mode currently in use and no **mediaChannel** should close the appropriate channel and the attempt to reopen using the **openLogicalChannel-openLogicalChannelAck** sequence, where the **openLogicalChannelAck** contains the address to which the endpoint will send the media.

Appendix I contains examples of the **communicationModeTable** entries for various cases.

8.4 Phase D – Call services

8.4.1 Bandwidth changes

Call bandwidth is initially established and approved by the Gatekeeper during the admissions exchange. An endpoint shall assure that the aggregate for all transmitted and received audio and video channels, excluding any RTP headers, RTP payload headers, network headers, and other overhead, is within this bandwidth. Data and control channels are not included in this limit.

At any time during a conference, the endpoints or Gatekeeper may request an increase or decrease in the call bandwidth. An endpoint may change the bit rate of a logical channel without requesting a bandwidth change from the Gatekeeper if the aggregate bit rate of all transmitted and received channels does not exceed the current call bandwidth. If the change will result in a aggregate bit rate that exceeds the current call bandwidth, the endpoint shall request a change in the call bandwidth from its Gatekeeper and await confirmation prior to actually increasing any bit rate. A bandwidth change request is recommended when an endpoint will use a reduced bandwidth for an extended period of time, thus freeing up bandwidth for other calls.

An endpoint wishing to change its call bandwidth sends a Bandwidth Change Request (BRQ) message (1) to the Gatekeeper. The Gatekeeper determines if the request is acceptable. The criteria for this determination is outside the scope of this Recommendation. If the Gatekeeper determines

that the request is not acceptable, it returns a Bandwidth Change Reject (BRJ) message (2) to endpoint. If the Gatekeeper determines that the request is acceptable, it returns a Bandwidth Change Confirm (BCF) message (2).

If Endpoint 1 wishes to increase its transmitted bit rate on a logical channel, it first determines if the call bandwidth will be exceeded. See Figure 43. If it will, Endpoint 1 shall request a bandwidth change (1 and 2) from Gatekeeper 1. When the call bandwidth is sufficient to support the change, Endpoint 1 sends a **closeLogicalChannel** (3) message to close the logical channel. It then reopens the logical channel using the **openLogicalChannel** (4) specifying the new bit rate. If the receiving endpoint wishes to accept the channel with the new bit rate, it must first assure that its call bandwidth is not exceeded by the change. If it is, the endpoint shall request a call bandwidth change (5 and 6) with its Gatekeeper. When the call bandwidth is sufficient to support the channel, the endpoint replies with an **openLogicalChannelAck** (7); otherwise, it responds with an **openLogicalChannelReject** indicating unacceptable bit rate.

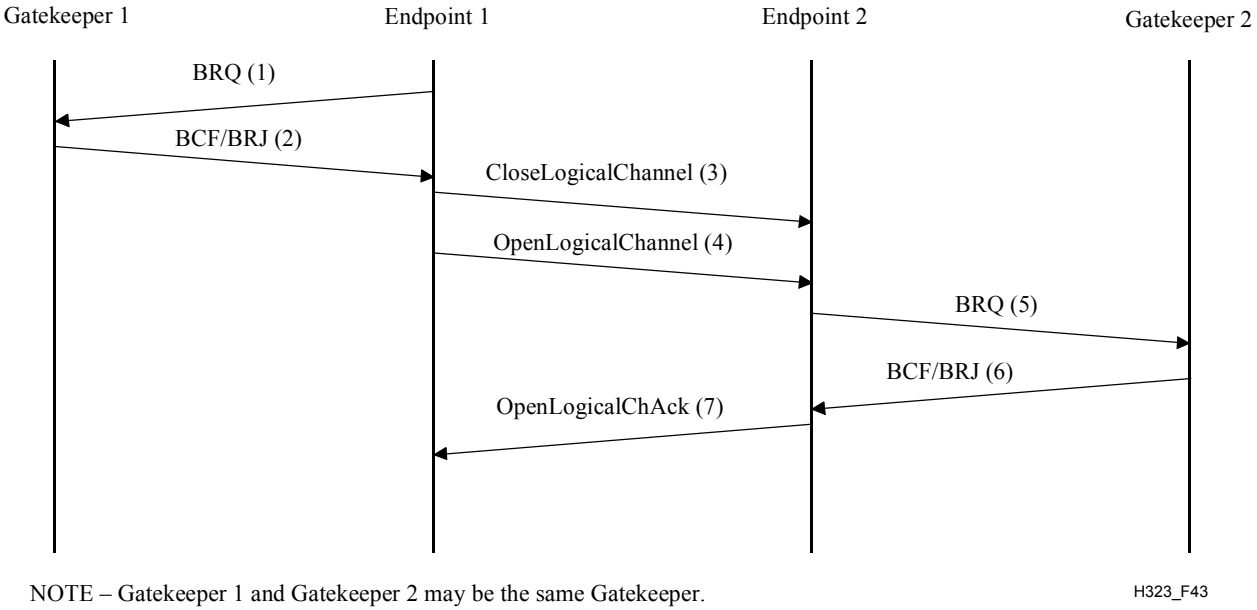
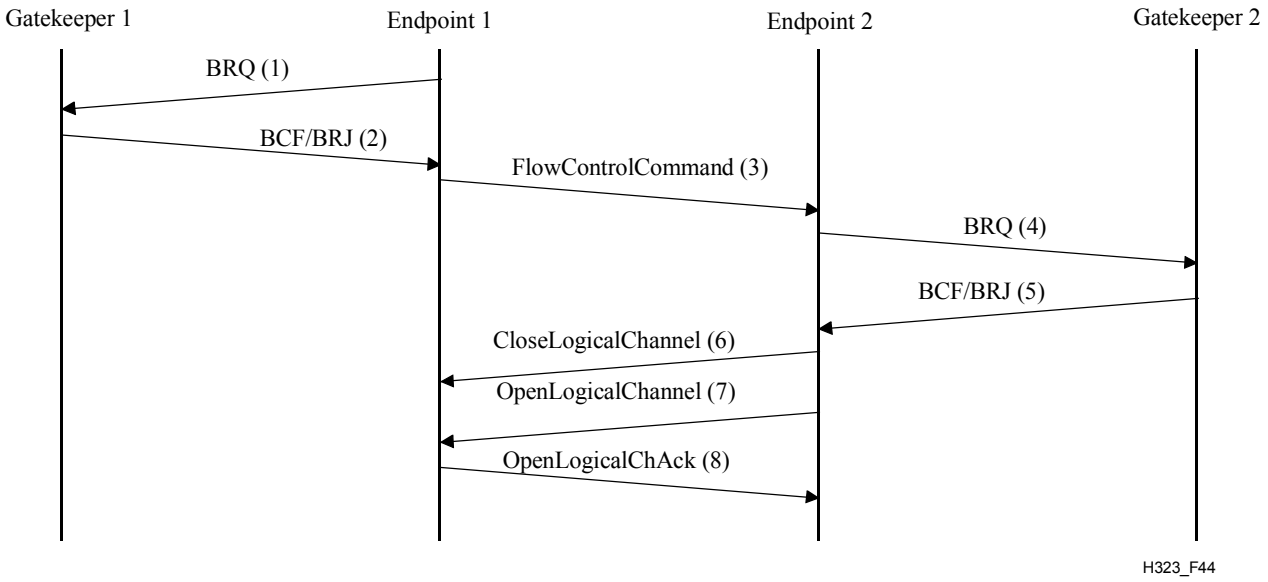


Figure 43/H.323 – Bandwidth change request – Transmitter change

If Endpoint 1 wishes to increase its transmitted bit rate on a logical channel from Endpoint 2, which it previously flow controlled to a lower bit rate, Endpoint 1 first determines if the call bandwidth will be exceeded. See Figure 44. If it will, Endpoint 1 shall request a bandwidth change from Gatekeeper 1. When the call bandwidth is sufficient to support the change, Endpoint 1 sends a **flowControlCommand** (3) to indicate the new upper limit on bit rate for the channel. If Endpoint 2 decides to increase the bit rate on the channel, it must first assure that its call bandwidth is not exceeded by the change. If it is, Endpoint 2 shall request a call bandwidth change (4 and 5) with its Gatekeeper. When the call bandwidth is sufficient to support the channel, Endpoint 2 will send the **closeLogicalChannel** (6) message to close the logical channel. It then reopens the logical channel using the **openLogicalChannel** (7) specifying the new bit rate. Endpoint 1 should then accept the channel with the new bit rate, and it replies with an **openLogicalChannelAck** (8).



NOTE – Gatekeeper 1 and Gatekeeper 2 may be the same Gatekeeper.

Figure 44/H.323 – Bandwidth change request – Receiver change

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

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

The endpoint shall send a BRQ message to the Gatekeeper whenever bandwidth utilization decreases below that which was specified in the original ARQ or the last BRQ or BCF message. The endpoint shall also send a BRQ message to the Gatekeeper whenever logical channel signalling results in the addition or removal of a unique multicast stream to or from the endpoint.

Bandwidth information may be used by a Gatekeeper to better manage bandwidth usage on the network. It should be noted that precise bandwidth management requires the Gatekeeper to understand the network topology, which is outside the scope of this Recommendation. In addition, the bandwidth usage by the endpoint may actually be different than that which was reported due to the use of silence suppression, variable bit-rate codecs, or other factors. An endpoint shall not repeatedly send BRQ messages to its Gatekeeper when actual bandwidth utilization fluctuates due to these factors. Rather, the endpoint should request necessary bandwidth based on the set of open logical channels and should not consider periods of silence or other factors as a decrease in bandwidth.

8.4.2 Status

In order for the Gatekeeper to determine if an endpoint is turned off or has otherwise entered a failure mode, the Gatekeeper may use the Information Request (IRQ)/Information Request Response (IRR) message sequence (see ITU-T Rec. H.225.0) to poll the endpoints at an interval decided by the manufacturer. The Gatekeeper may request information for a single call or for all

active calls. Except when requesting additional IRR segments, the polling interval to request information for a particular call or all calls shall be greater than 10 s. However, the Gatekeeper may send IRQ messages that contain unique **callReferenceValue** values without regard to the polling period. This message may also be used by a diagnostic device as described in 11.2.

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

If the Gatekeeper wants to receive call details for all of the active calls on an endpoint, it may send an IRQ message with the **callReferenceValue** field set to 0. The Gatekeeper should include the **segmentedResponseSupported** field to allow requests for all calls to be segmented if necessary. If the **segmentedResponseSupported** field is included, the endpoint shall return all or part of the call information in the **perCallInfo** field in a single IRR message. If segmentation is not allowed, but not all call details can be included in the IRR message, the endpoint shall include the **incomplete** field in the IRR message. If segmentation is allowed, the endpoint may return one or multiple IRR messages in response to the IRQ message. If one IRR message containing all call detail information is returned, the **irrStatus** element shall not be present. If the response is segmented into multiple IRR messages, the endpoint shall send the first IRR message and include the **segment** field. If the Gatekeeper wishes to receive the next segment, it shall transmit another IRQ message that includes the **segmentedResponseSupported** field, has the **callReferenceValue** set to 0, and has the **nextSegmentRequested** field set to the value of the next segment that the Gatekeeper expects to receive. If the Gatekeeper wishes to receive additional segments, it shall send the next IRQ message within 5 seconds after receiving the previous IRR message. If the endpoint receives a request for additional segments after 5 seconds (plus locally determined appropriate time for network delay), it may return an IRR message with the **incomplete** field included. When receiving an IRQ message from the Gatekeeper requesting the next segment within the allotted time, the endpoint shall transmit the next IRR message containing the next segment of call information. Note that if an IRR message is lost, the Gatekeeper may retransmit a request for the previously transmitted segment. Therefore, the endpoint shall be prepared to transmit the previous or next segment. If no additional segments are available or when the endpoint transmits the last segment of a series of IRR messages, the endpoint shall return an IRR message that includes the **complete** field. The Gatekeeper shall not transmit a different IRQ message to the endpoint requesting all call detail information until the last segment of information is transmitted or until the 10-second polling period has elapsed.

NOTE 1 – Since calls may begin or end after sending the first IRR message segment in response to an IRQ message requesting call details for all calls, the endpoint may or may not choose to include such calls when sending subsequent IRR message segments. The decision to report such calls when sending subsequent IRR segments is left to the manufacturer.

NOTE 2 – In order to improve performance and achieve better scalability, a Gatekeeper should limit the frequency at which it requests call details for all calls. Requesting call details for all calls is beneficial when an endpoint initially registers with the Gatekeeper, for example. However, repeatedly requesting such information – especially from very large-scale Gateways or MCUs – may lead to unacceptable performance degradation.

The Gatekeeper may want an endpoint to periodically send an unsolicited IRR message. The Gatekeeper may indicate this to the endpoint by specifying the rate that this IRR is sent within the **irrFrequency** field of the Admission Confirm (ACF) message. An endpoint receiving this **irrFrequency** rate shall send an IRR message at that rate for the duration of the call. While this rate is in effect, the Gatekeeper may still send IRQ messages to the endpoint which shall respond as described above.

An endpoint may want some of the unsolicited IRRs to be delivered reliably. The Gatekeeper can enable this by using the **willRespondToIRR** field in the RCF or ACF that it can acknowledge unsolicited IRRs. In this case, the endpoint may explicitly request the Gatekeeper to send an acknowledgment for the IRR. The Gatekeeper shall respond to such an IRR message by sending either an acknowledgment (IACK) or a negative acknowledgment (INAK). If the Gatekeeper did not announce that it will acknowledge IRRs, or if the endpoint did not request such an acknowledgment, no response shall follow the IRR.

During the duration of a call, an endpoint or Gatekeeper may periodically request call status from another endpoint. The requesting endpoint or Gatekeeper issues a Status Enquiry message. The Endpoint receiving the Status Enquiry message shall respond with a Status message indicating the current call state. This procedure may be used by the Gatekeeper in order to periodically check if a call is still active. Endpoints shall be able to accept any valid state values received in the Status message, including those which it may not be capable of entering. Note that this is an H.225.0 message sent on the Call Signalling Channel and should not be confused with IRR which is a RAS message sent on the RAS Channel.

The Gatekeeper may want to receive copies of certain H.225.0 call signalling PDUs when they are received or sent by an endpoint. An endpoint indicates its capability to send these PDUs by setting the **willSupplyUUIEs** in the ARQ or RRQ message sent to the Gatekeeper. The Gatekeeper indicates the list of PDU types it wishes to receive copies of, in the **uuiesRequested** field in the ACF or RCF. It also indicates if it wants copies when the PDUs are sent or received. An endpoint indicating this capability and receiving this list, shall send an IRR to the Gatekeeper each time it receives/sends the type of PDU requested.

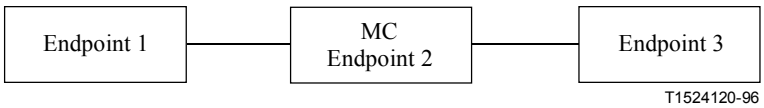
8.4.3 Ad hoc conference expansion

The following procedures are optional for terminals and Gateways and mandatory for MCs.

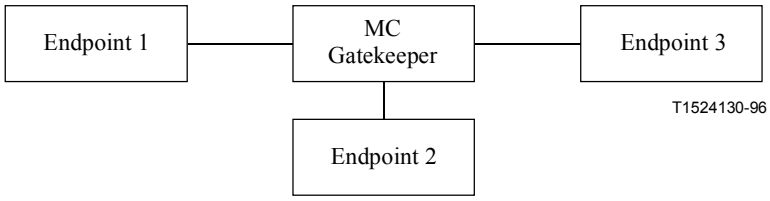
When a user places a call, the intent of the call is often not known to the calling endpoint. The user may wish to simply create a conference for itself and the called endpoint, the user may wish to join some conference at the called entity, or the user may wish to get a list of conferences that the called entity can provide. Using the procedures of this clause the conferences can be expanded from point-to-point calls into Ad Hoc Multipoint conferences.

An Ad Hoc Multipoint conference is one that can be expanded from a point-to-point conference involving an MC to a multipoint conference. First, a point-to-point conference is created between two endpoints (Endpoint 1 and Endpoint 2). At least one endpoint or the Gatekeeper must contain an MC. Once the point-to-point conference has been created, the conference may be expanded to multipoint conference in two different ways. The first way is when any endpoint in the conference invites another endpoint (Endpoint 3) into the conference by calling that endpoint through the MC. The second way is for an endpoint (Endpoint 3) to join an existing conference by calling an endpoint in the conference.

Ad Hoc Conference expansion can take place when using either the direct call signalling model or the Gatekeeper routed call signalling model. The H.245 Control Channel topology for the direct call signalling model appears as:



The H.245 Control Channel topology for the Gatekeeper routed call signalling model appears as:



In either case an MC must be present in the conference at the time of expansion to any number greater than 2 endpoints. Note that in the Gatekeeper routed model, the MC may be located in the Gatekeeper and/or one of the endpoints.

The procedures required to create a point-to-point conference and then expand the conference through invite and join, for each call model, is covered in the following subclauses. Procedures for the calling endpoint to discover a list of conferences that the called entity can provide are also covered.

It should be noted that the call is ended by a failure of the entity that is providing the MC.

8.4.3.1 Direct endpoint call signalling – conference create

Endpoint 1 creates a conference with Endpoint 2 as follows:

A1) Endpoint 1 sends a Setup message to Endpoint 2 containing a globally unique CID = N and **conferenceGoal** = **create** according to the procedure in 8.1.

A2) Endpoint 2 has the following options:

- A2a) If it wants to join the conference, it sends a Connect message with CID = N to Endpoint 1. In this case it is either:
  1) not participating in another conference; or
  2) it is participating in another conference, it is capable of participating in multiple conferences at the same time, and the received CID = N does not match the CID of any of the conferences in which it is currently participating.
- A2b) If it is in another conference with CID = M and can participate in only one conference at a time it either:
  1) rejects the call by sending Release Complete indicating in-conference; or
  2) it can request Endpoint 1 to join the conference with CID = M by sending a Facility message indicating **routeCallToMC** with the Call Signalling Channel Transport Address of the endpoint containing the MC and CID = M of the conference. The handling of the Facility message by Endpoint 1 is described in 8.4.3.7.
- A2c) If it does not wish to join this conference, it rejects the call by sending Release Complete indicating that the destination is busy.
- A2d) If Endpoint 2 is an MC(U) that hosts multiple conferences and wishes to provide Endpoint 1 with a choice of conferences to join, it can send a Facility message indicating **conferenceListChoice** and a list of conferences that Endpoint 1 may choose from. The list of conferences is sent as part of the Facility-UUIE. For backward compatibility, with Version 1 endpoints, conference lists are only provided if the **protocolIdentifier** in Endpoint 1's Setup message indicates that it is Version 2 or above.

Upon receipt of this **conferenceListChoice** Facility message, Endpoint 1 may join a conference from the list of conferences by sending a new Setup message to the MC(U) on the Call Signalling Channel which contains the selected CID and which has **conferenceGoal** = **join**. If Endpoint 1 chooses not to join any of the listed conferences, it shall send a Release Complete message to the MC(U).

A3) If Endpoint 2 enters the conference, Endpoint 1 uses the Transport Address of the Control Channel provided in the Connect message to open the Control Channel with Endpoint 2.

A4) The H.245 messages are then exchanged as described below:

A4a) **terminalCapabilitySet** messages are exchanged between the endpoints to determine the version number of the H.245 used in order to parse the remaining received messages correctly.

A4b) Using H.245 master-slave determination procedure, it is determined that Endpoint 2 is the master. In the Gatekeeper-Routed model, the master could be in an MC collocated with the Gatekeeper. If the master has an MC, it becomes the Active MC. It may then send the **mcLocationIndication** to the other endpoint(s). The MC may be active in the conference now or when the user initiates the multipoint conference function, at the choice of the manufacturer.

A4c) The master may send the **terminalNumberAssign** message to the endpoints. The endpoints shall use the 8-bit terminal number and not use the 8-bit MCU number from the 16-bit number assigned as the low 8 bits of the SSRC field in the RTP header. These low 8 bits in SSRC then identify the streams from a particular endpoint.

A4d) Since the capabilities of the receiver are known from the **terminalCapabilitySet** message, the transmitter opens the logical channels. It shall send one **h2250MaximumSkewIndication** for each pair of audio and video transmitted.

8.4.3.2 Direct endpoint call signalling – conference invite

There are two cases of the conference invite. First, the endpoint which contains the Active MC wishes to invite another endpoint into the conference. Second, an endpoint which does not contain the Active MC wishes to invite another endpoint into the conference.

1) After a point-to-point conference has been established using procedures A1) to A4) in 8.4.3.1, an endpoint (Endpoint 2) containing the Active MC wishing to add another endpoint to the conference shall use the following procedure:

B1) Endpoint 2 sends a Setup message to Endpoint 3 with CID = N and **conferenceGoal** = **invite** according to the procedures in 8.1. See Figure 45.

B2) Endpoint 3 has the following options:

B2a) If it wishes to accept the invitation to join the conference, it sends a Connect message with CID = N to Endpoint 2.

B2b) If it wishes to reject the invitation to join the conference, it sends a Release Complete message to Endpoint 2 indicating that the destination is busy.

B2c) If it is in another conference with CID = M, it can request Endpoint 2 to join the conference with CID = M by sending a Facility message indicating **routeCallToMC** with the Call Signalling Channel Transport Address of the endpoint containing the MC and CID = M of the conference. The handling of the Facility message by Endpoint 2 is described in 8.4.3.7.

B2d) If the received CID matches the CID of a conference that Endpoint 3 is currently participating in, it shall reject the call by sending Release Complete indicating that it is already in the conference.

B3) If Endpoint 3 accepts the invitation, Endpoint 2 uses the Transport Address of the Control Channel provided in the Connect message to open the Control Channel with Endpoint 3.

B4) The H.245 messages are then exchanged as described below:

C1) **terminalCapabilitySet** messages are exchanged between the MC and Endpoint 3.

C2) Using H.245 master-slave determination procedure, it is determined that Endpoint 2 is already the Active MC. The MC may then send the **mcLocationIndication** to the Endpoint 3.

C3) The MC shall send **multipointConference** at this time to all the three endpoints.

C4) The MC may send the **terminalNumberAssign** message to Endpoint 3. If received, the endpoints shall use the 8-bit terminal number and not use the 8-bit MCU number from the 16-bit number assigned as the low 8 bits of the SSRC field in the RTP header. These low 8 bits in SSRC then identify the streams from a particular endpoint.

C5) An endpoint can get the list of the other endpoints in the conference by sending the **terminalListRequest** message to the MC. The MC responds with the **terminalListResponse**.

C6) Whenever a new endpoint joins the conference, the MC sends the **terminalNumberAssign** message to Endpoint 4 and **terminalJoinedConference** message to Endpoints 1, 2 and 3.

C7) Whenever an endpoint leaves the conference, the MC sends **terminalLeftConference** to the remaining endpoints.

C8) The MC shall send the **communicationModeCommand** to all the endpoints in the conference.

C9) Endpoint 1 and Endpoint 2 will close their logical channels that were created during the point-to-point conference if they are inconsistent with the information contained in the **communicationModeCommand**.

C10) The logical channels can now be opened between the MC and the endpoints.

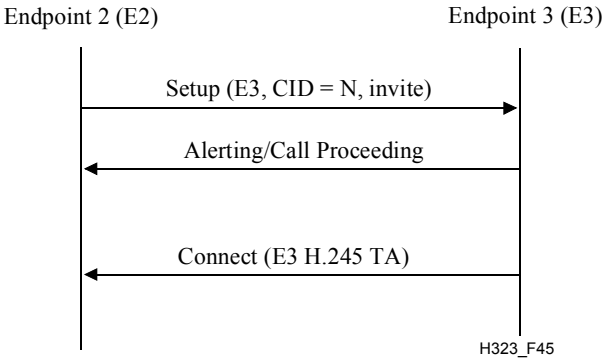


Figure 45/H.323 – MC invite signalling

2) After a point-to-point conference has been established using procedures A1) to A4) in 8.4.3.1, an endpoint (Endpoint 1) that does not contain the Active MC wishing to add another endpoint to the conference shall use the following procedure:

B1) Endpoint 1 sends a Setup message to the MC (Endpoint 2) with a new CRV indicating a call to Endpoint 3 by providing the Transport Address of Endpoint 3, CID = N, and **conferenceGoal** = **invite**. See Figure 46.

B2) Endpoint 2 sends a Setup message to Endpoint 3 with CID = N and **conferenceGoal = invite** according to the procedures in 8.1.

B3) During call signalling with Endpoint 3, Endpoint 2 shall pass Call Signalling messages received from Endpoint 3, including Connect, to Endpoint 1 (the original inviter).

B4) Endpoint 3 has the same options, described previously, of either accepting or rejecting the invitation.

B5) At some time after the completion of the call setup procedure between Endpoint 2 and Endpoint 3, Endpoint 2 shall send a Release Complete message to Endpoint 1.

B6) If Endpoint 3 accepts the invitation, Endpoint 2 uses the Transport Address of the Control Channel provided in the Connect message to open the Control Channel with Endpoint 3.

B7) The H.245 messages are then exchanged as previously described in procedures C1) to C10).

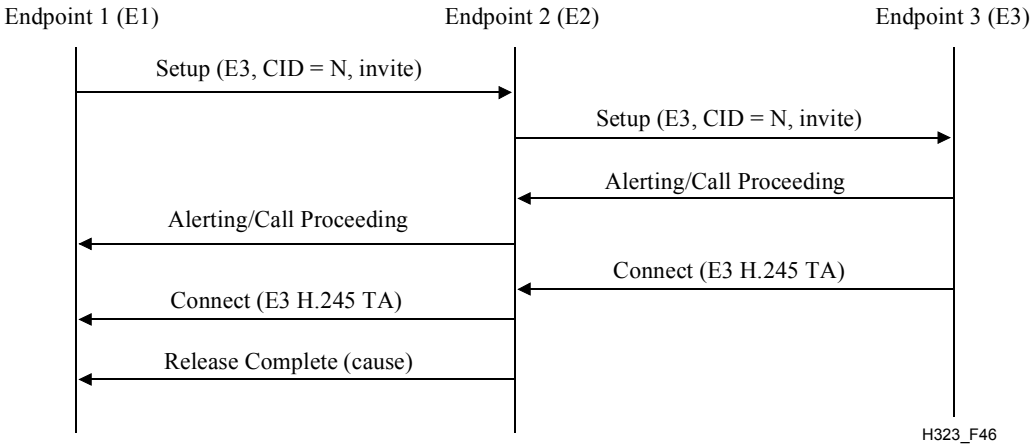


Figure 46/H.323 – Non-MC invite signalling

8.4.3.3 Direct endpoint call signalling – conference join

There are two cases of the conference join. First, an endpoint calls the endpoint which contains the Active MC. Second, an endpoint calls an endpoint which is not the Active MC.

After a point-to-point conference has been established using procedures A1) to A4) in 8.4.3.1, an endpoint (Endpoint 3) wishing to join a conference may attempt to connect with the endpoint containing the Active MC in the conference. In this case, the following procedure shall be used:

B1) Endpoint 3 sends a Setup message to Endpoint 2 with CID = N and **conferenceGoal = join** according to the procedures in 8.1. See Figure 47.

B2) If the CID matches the CID of an active conference in the MC, Endpoint 2 (MC) has the following options:

B2a) If it decides that Endpoint 3 should be allowed to join the conference, it sends the Connect message with CID = N.

B2b) If it decides that Endpoint 3 should not be allowed to join the conference, it sends the Release Complete message indicating that the destination is busy.

B3) If the CID does not match the CID of an active conference in the MC, Endpoint 2 shall send Release Complete indicating a bad CID.

B4) If Endpoint 2 allows the join, Endpoint 2 opens the Control Channel with Endpoint 3.

B5) The H.245 messages are then exchanged as previously described in procedures C1) to C10).

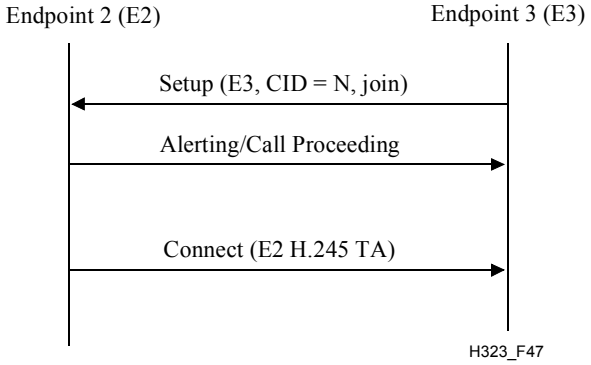


Figure 47/H.323 – MC join signalling

After a point-to-point conference has been established using procedures A1) to A4), an endpoint (Endpoint 3) wishing to join a conference may attempt to connect with an endpoint that does not contain the Active MC in the conference. In this case, the following procedure shall be used:

B1) Endpoint 3 sends a Setup message to Endpoint 1 with CID = N and **conferenceGoal** = **join** according to the procedures in 8.1. See Figure 48.

B2) Endpoint 1 returns a Facility message indicating **routeCallToMC** with the Call Signalling Channel Transport Address of Endpoint 2 (containing the Active MC) and the CID = N of the conference.

B3) Endpoint 3 then sends a Setup message to Endpoint 2 (MC) with CID = N and **conferenceGoal** = **join** as described in the previous conference join procedure.

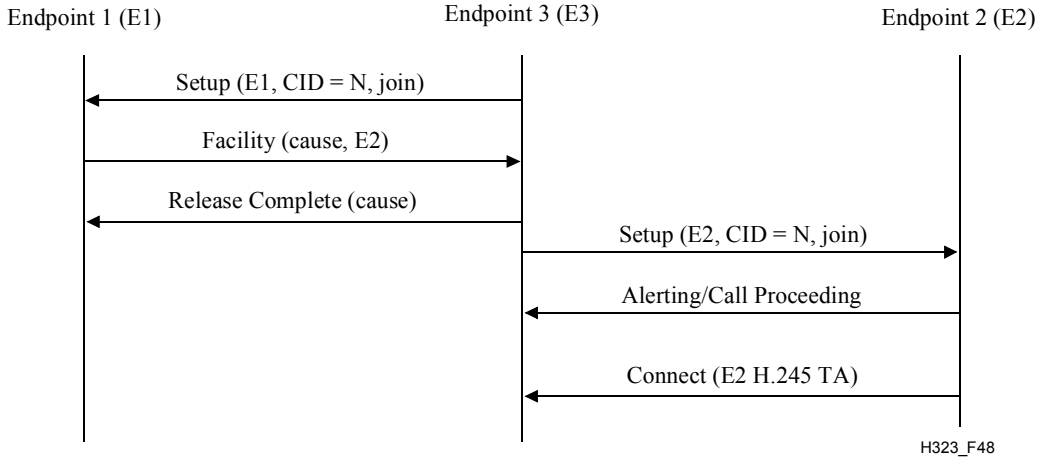


Figure 48/H.323 – Non-MC join signalling

8.4.3.4 Gatekeeper routed call signalling – conference create

In cases where the Gatekeeper routes the Call Signalling Channel and the H.245 Control Channel, the Gatekeeper may contain (or have access to) an MC or MCU. Procedures A1) to A4) are used to establish the point-to-point call.

If the MC(U) hosts multiple conferences and wishes to provide Endpoint 1 with a choice of conferences to join, it can send a Facility message indicating **conferenceListChoice** and a list of conferences that Endpoint 1 may choose from. The list of conferences is sent as part of the Facility-UUIE. For backward compatibility, with Version 1 endpoints, conference lists are only

provided if the **protocolIdentifier** in Endpoint 1's Setup message indicates that it is Version 2 or above.

Upon receipt of this **conferenceListChoice** Facility message, Endpoint 1 may join a conference from the list of conferences by sending a new Setup message to the MC(U) on the Call Signalling Channel which contains the selected CID and which has **conferenceGoal** = **join**. If Endpoint 1 chooses not to join any of the listed conferences, it shall send a Release Complete message to the MC(U).

During master-slave determination [A4b], if the Gatekeeper's **terminalType** is greater than the **terminalType** received in the **masterSlaveDetermination** message, the Gatekeeper may attempt to become master for the call. In this case, the Gatekeeper shall immediately send a **masterSlaveDeterminationAck** message to the source of the Master-Slave Determination message indicating that it is a slave, and the Gatekeeper performs Master-Slave Determination with the destination entity as defined in 6.2.8.4. If the Gatekeeper wins that Master-Slave Determination, the MC associated with the Gatekeeper shall be the Active MC. If the Gatekeepers **terminalType** is not greater than the **terminalType** of the endpoint or the Gatekeeper decides not to replace the endpoint's **terminalType** with its own, the Gatekeeper shall not modify the **terminalType** value and it shall transparently relay all messages of that Master-Slave Determination procedure.

8.4.3.5 Gatekeeper routed call signalling – conference invite

After a point-to-point conference has been established using procedures A1) to A4) as modified above, an endpoint (Endpoint 1 or 2) that does not contain the Active MC wishing to add another endpoint to the conference shall use the following procedure:

B1) Endpoint 1 sends a Setup message through the Gatekeeper directed to Endpoint 3 with a new CRV, CID = N and **conferenceGoal** = **invite**. See Figure 49.

B2) The Gatekeeper (MC) sends a Setup message to Endpoint 3 with CID = N and **conferenceGoal** = **invite** according to the procedures in 8.1.

B3) During call signalling with Endpoint 3, the Gatekeeper shall pass Call Signalling messages received from Endpoint 3, including Connect, to Endpoint 1 (the original inviter).

B4) Endpoint 3 has the same options, described previously, of either accepting or rejecting the invitation.

B5) At some time after the completion of the call setup procedure between the Gatekeeper and Endpoint 3, the Gatekeeper shall send a Release Complete message to Endpoint 1.

B6) If Endpoint 3 accepts the invitation, the Gatekeeper uses the Transport Address of the Control Channel provided in the Connect message to open the Control Channel with Endpoint 3.

B7) The H.245 messages are then exchanged as previously described in procedures C1) to C10) with the Gatekeeper taking part in all master-slave determination procedures as the Active MC (C2). At this time, the Control Channels from the endpoints should be connected to the MC, and the MC should be in control of the conference.

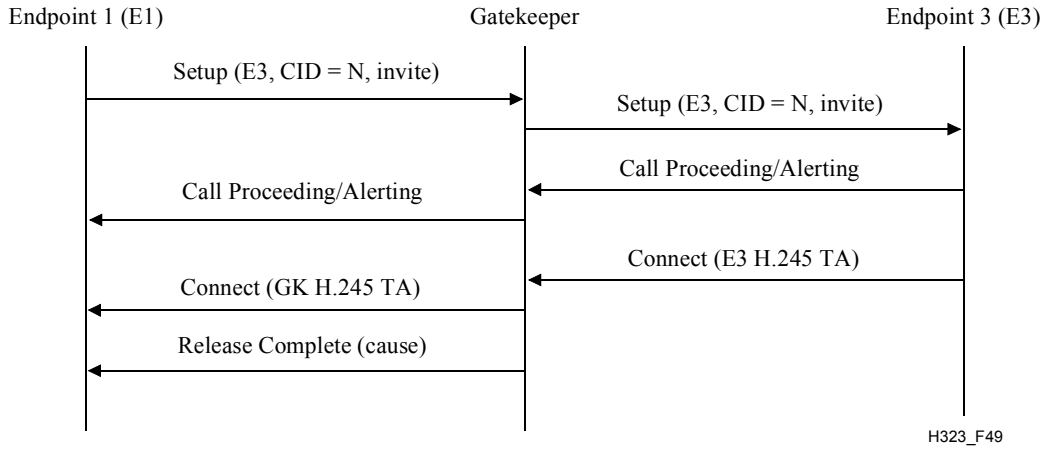


Figure 49/H.323 – Gatekeeper routed invite signalling

8.4.3.6 Gatekeeper routed call model – conference join

After a point-to-point conference has been established using procedures A1) to A4) as modified above, an endpoint (Endpoint 3), wishing to join a conference may attempt to connect with an endpoint that does not contain the Active MC in the conference. In this case, the following procedure shall be used:

B1) Endpoint 3 sends a Setup message through the Gatekeeper directed to Endpoint 1 with CID = N and **conferenceGoal = join** according to the procedures in 8.1. See Figure 50.

B2) If the CID matches the CID of an active conference in the MC, the Gatekeeper (MC) has the following options:

- B2a) If it decides that Endpoint 3 should be allowed to join the conference, it sends the Connect message with CID = N to Endpoint 3.
- B2b) If it decides that Endpoint 3 should not be allowed to join the conference, it sends the Release Complete message indicating that the destination is busy.
- B2c) The Gatekeeper may forward the Setup message to Endpoint 1. Endpoint 1 may respond with a Facility message indicating **routeCallToMC** or it may respond with a Release Complete.

B3) If the CID does not match the CID of an active conference in the MC, the Gatekeeper shall send Release Complete indicating a bad CID.

B4) If the Gatekeeper allows the join, the Gatekeeper uses the Transport Address of the Control Channel provided in the Setup message to open the Control Channel with Endpoint 3.

B5) The H.245 messages are then exchanged as previously described in procedures C1) to C10) with the Gatekeeper taking part in all master-slave determination procedures as the Active MC (C2). At this time, the Control Channels from the endpoints should be connected to the MC, and the MC should be in control of the conference.

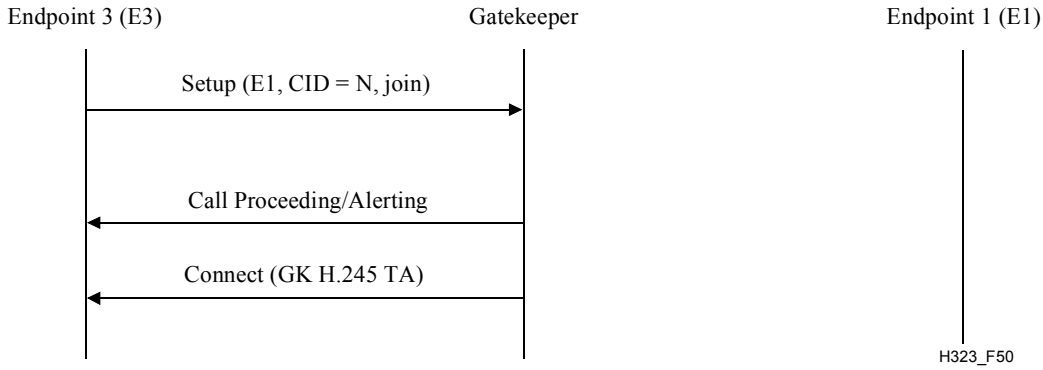


Figure 50/H.323 – Gatekeeper routed join signalling

8.4.3.7 Handling of the facility message

Upon receiving a Facility message indicating **routeCallToMC** with the Call Signalling Channel Transport Address of the endpoint containing the MC and CID of a conference, an endpoint may release the current call and attempt to join the indicated conference according to the procedures in 8.4.3.3 or in 8.4.3.6.

An endpoint may receive such a Facility message either as a direct reply to its Setup message or during the active phase of a call.

8.4.3.8 Conference out of consultation

This clause defines the procedures for an endpoint (endpoint A) requesting an ad-hoc conference with two or more other endpoints (remote endpoints B, C, etc.) with which endpoint A already has active calls. This typically applies – but is not limited to – ad-hoc conference being requested out of a consultation condition.

NOTE 1 – "Consultation condition" refers to a situation where endpoint A has an active call with endpoint C (consultation call) while having one or more endpoints on hold, i.e., held call(s). An endpoint may be put on hold by using the procedures of ITU-T Rec. H.450.4 [36], 8.4.6, or by local procedures.

Endpoint A has the capability of "merging" the independent calls to multiple endpoints into a single conference either at the endpoint A (as described in scenario 1 below) or by forming the conference on a separate MCU (as described in scenario 2 below).

NOTE 2 – Procedures in this clause relate only to the calls at an endpoint that are to be joined into a conference out of consultation. An endpoint may have additional calls that do not participate in the conference and to which this clause will not apply.

8.4.3.8.1 Scenario 1: Conference provided by endpoint

If endpoint A has the capabilities, it may "merge" the held call and the consulted call into a conference resulting in a three-way conversation between A, B and C. For this scenario endpoint A must have an MC. Both the centralized and the decentralized conferencing models are possible. If the centralized model is to be used (i.e., if the terminal provides the media mixing/switching), endpoint A shall have an MP.

An endpoint with MC and MP is actually an MCU and should use **terminalType** 170, 180 or 190 as appropriate for master-slave determination.

The following scenarios are possible:

1a) If endpoint A is the master of both calls to B and C, it may simply retrieve the held call into the conference with C and declare itself as the Active MC on both calls through master-slave negotiation.

1b) If endpoint A is a slave on one or more of the calls but no call on which it is the slave has an Active MC, endpoint A should reinitiate master-slave determination on all calls in which it is slave, using the **terminalType** 240, as specified in Table 1 for an Active MC. If it ends this procedure as master on all calls, it should act as in 1a) above; if it is slave in one or more calls, endpoint A should act as directed in 1c) below.

1c) If one or more of the calls in which endpoint A is participating is already a call in which endpoint A is not the Active MC, procedures for cascading MCUs shall be followed.

Once a conference is established within endpoint A, a further endpoint D – that is being consulted by endpoint A – may be invited into the existing conference as described in 8.4.3.2 and 8.4.3.5.

8.4.3.8.2 Scenario 2: Conference provided by MCU

If endpoint A has access to an MCU, the following procedure may be used to accomplish conference out of consultation:

2a) Endpoint A establishes a new call to the MCU using a Setup message with **conferenceGoal = create** and CID = N.

2b) Endpoint A drops its call with endpoint C using a Release Complete message with **reason** set to **replaceWithConferenceInvite** including argument CID = N.

2c) Endpoint A sends a Setup message to the MCU with **conferenceGoal = invite**, CID = N, and sufficient information for the MCU to make a call to endpoint C (see also 8.4.3.2).

2d) Steps 2b) and 2c) shall be repeated with "endpoint C" replaced by "endpoint B". Note that there is no requirement to retrieve the call to B from hold before inviting it to the conference.

2e) For exchange of H.245 conference related messages refer to 8.4.3.2 of H.323 steps C1)-C10).

Alternative mechanisms to steps 2b), 2c) and 2d) are:

1) H.450.2 [35] Call Transfer (with endpoint A acting as "transferring" endpoint, endpoints B and C acting as "transferred" endpoints and the MC/MCU acting as the "transferred-to" endpoint. The Facility message containing **callTransferInitiate Invoke APDU** shall also contain element CID = N.

2) H.225.0 "Facility re-route to MC" mechanism (sending an H.225.0 Facility message to endpoints B and C containing CID = N, **facilityReason = routeCallToMC** and the address of the MCU) if H.450.2 is not supported.

These alternative mechanisms are recommended if the remote endpoint is located within the SCN.

An endpoint (e.g., endpoint A) may split from the conference (e.g., by putting its call to the MCU on hold). Endpoint A may then consult with a further endpoint D that may subsequently be invited to the existing conference by using the procedures as described in 2b) and 2c) above with "endpoint C" replaced by "endpoint D". Alternative mechanisms as described above by means of using H.450.2 Call Transfer or H.225.0 "Facility re-route to MC" may be used instead.

8.4.4 Supplementary services

Support for Supplementary Services is optional. The H.450.x series of Recommendations describes a method of providing Supplementary Services in the H.323 environment.

8.4.5 Multipoint cascading

In order to cascade MCs, a call must be established between the entities containing the MCs. This call is established according to the procedures defined in 8.1 and 8.4.3. Once the call is established and the H.245 Control Channel is opened, the Active MC (determines according to the master/slave procedures in 6.2.8.4) may activate the MC in a connected entity. This is done by using the H.245 **remoteMC** message. The following results shall occur in response to the **remoteMC** message:

Calling entity	Called entity	Conference goal	RemoteMC Sender	RemoteMC Selection	Result
Active MC	Inactive MC	**create**	Calling entity	**masterActivate**	Called MC accepts request and becomes the master MC
Active MC	Inactive MC	**invite**	Calling entity	**slaveActivate**	Called MC accepts request and becomes a slave MC
Active MC	Inactive MC	**join**	N/A	N/A	Not allowed
Inactive MC	Active MC	**create**	N/A	N/A	Not allowed
Inactive MC	Active MC	**invite**	N/A	N/A	Not allowed
Inactive MC	Active MC	**join**	Called entity	**slaveActivate**	Calling MC accepts request and becomes a slave MC

Once the cascaded conference is established, either the master or slave MCs may invite other endpoints into the conference. There shall only be one master MC in a conference. A slave MC shall only be cascaded to a master MC. Slave MCs shall not be cascaded to other slave MCs. This allows only dumb-bell or star cascaded configurations.

The slave MC shall identify the cascaded conference using the CID established by the master when the conference was created.

The slave MC shall accept and act upon **communicationsModeCommand** messages from the master MC. The slave MC shall forward these messages to its locally connected endpoints. The slave MC may receive **requestMode** messages from its locally connected endpoints. It should forward these to the master MC. The slave MC shall not send **communicationsModeCommand** messages to the master MC.

The master MC should follow the procedures in 8.4.3.2, C3) through C10), in order to establish a common operating mode with the slave MC. Based on this information, each MC is responsible for opening logical channels for media distribution between its locally connected endpoints and endpoints designated by the master MC.

In addition to inviting new endpoints into the conference, an MC which supports multiple conferences may directly move endpoints into another conference without tearing down the existing connection. If this is done, the MC should send the **substituteCID** message to these endpoints. Endpoints which receive a **substituteCID** message during a call shall continue to use the conference ID (CID) used in the previous RAS messages (e.g., ARQ, BRQ, etc.) when conversing with its Gatekeeper for the duration of that particular call.

Terminal numbering and chair control functions may follow the procedures defined in ITU-T Rec. H.243. The use of T.120 for controlling MC cascading is for further study. The use of T.120 in cascaded connections is described in the T.120 series of Recommendations.

When a master sends a **remoteMC** Request with the selection **deActivate**, the slave MC should remove all endpoints from the conference.

8.4.6 Third party initiated pause and re-routing

For the purpose of this clause, an empty capability set is defined as a **terminalCapabilitySet** message that contains only a sequence number and a protocol identifier.

To allow Gatekeepers to re-route connections from endpoints that do not support supplementary services, endpoints shall respond to the reception of an empty capability set as defined in this clause. This feature allows "network" elements such as PBXs, call centers, and IVR systems to re-route connections independently of supplementary services and facilitates pre-connect announcements. It can also be used to delay H.245 media establishment when features such as Gatekeeper based user location are being used. It is also highly recommended that Version 1 endpoints support this feature.

On reception of an empty capability set, an endpoint shall enter a "transmitter side paused" state. On entering this state, the endpoint shall stop transmitting on established logical channels and shall close all logical channels that it previously opened, including bidirectional logical channels. It shall close these channels in the usual way by sending the **closeLogicalChannel** message. The endpoint shall not request the remote endpoint to close logical channels, either unidirectional or bidirectional, that the remote endpoint opened. The endpoint shall send the **terminalCapabilitySetAck** message in the usual way: the message may be sent before stopping transmission and so shall not be interpreted as an indication that transmission has stopped.

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

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

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

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

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

Unless its capabilities have changed, an endpoint need not resend a capability set as the Gatekeeper will have supplied this to the remote endpoint to remove any paused state in the remote endpoint. This option of not sending a capability set enables faster reconnection. If the first

terminalCapabilitySet message sent by an endpoint after leaving the "transmitter side paused" state differs from the capability set that the Gatekeeper provided to the remote endpoint, the Gatekeeper shall signal the remote endpoint to remove capabilities which were not indicated by the initiating endpoint.

NOTE 1 – An endpoint should take care with the capabilities it sends at this time. In particular, an endpoint shall send all capabilities it wants to advertise and not a small addition to previously signalled capabilities. In addition, if the endpoint has so many capabilities that it requires more than one **terminalCapabilitySet** to signal them, there may be a window of time when the gatekeeper has removed the capabilities described in second and subsequent **terminalCapabilitySet** messages.

NOTE 2 – A non-empty capability set shall not be sent to an endpoint until all of its transmit logical channels have been closed. A switching entity should also send an H.450 redirection indication Facility message if the endpoint is being re-routed.

8.5 Phase E – Call termination

Either endpoint or an intermediate call signalling entity may terminate a call. Call termination shall be accomplished by either Procedure A or Procedure B:

Procedure A:

A-1) It should discontinue transmission of video at the end of a complete picture, when applicable.

A-2) It should discontinue transmission of data, when applicable.

A-3) It should discontinue transmission of audio, when applicable.

A-4) It shall transmit a Release Complete message and close the H.225.0 call signalling channel and, if open separately, the H.245 Control Channel without sending any H.245 message. Note that closing the media channels is implied.

A-5) Endpoints shall clear the call by using the procedures defined in 8.5.1 or 8.5.2.

Procedure B:

B-1) It should discontinue transmission of video at the end of a complete picture and then close all logical channels for video, when applicable.

B-2) It should discontinue transmission of data and then close all logical channels for data, when applicable.

B-3) It should discontinue transmission of audio and then close all logical channels for audio, when applicable.

B-4) It shall transmit the H.245 **endSessionCommand** message in the H.245 Control Channel, indicating to the far end that it wishes to disconnect the call and then discontinue H.245 message transmission.

B-5) It shall then wait to receive the **endSessionCommand** message from the other endpoint and then shall close the H.245 Control Channel.

B-6) It shall transmit a Release Complete message and close the H.225.0 call signalling channel.

B-7) Endpoints shall clear the call by using the procedures defined in 8.5.1 or 8.5.2.

An endpoint receiving **endSessionCommand** without first having transmitted it shall carry out steps B-1) to B-7) above, except that in step B-5), it shall not wait for the **endSessionCommand** from the first endpoint.

Terminating a call may not terminate a conference; a conference may be explicitly terminated using an H.245 message (**dropConference**). In this case, the endpoints shall wait for the MC to terminate the calls as described above.

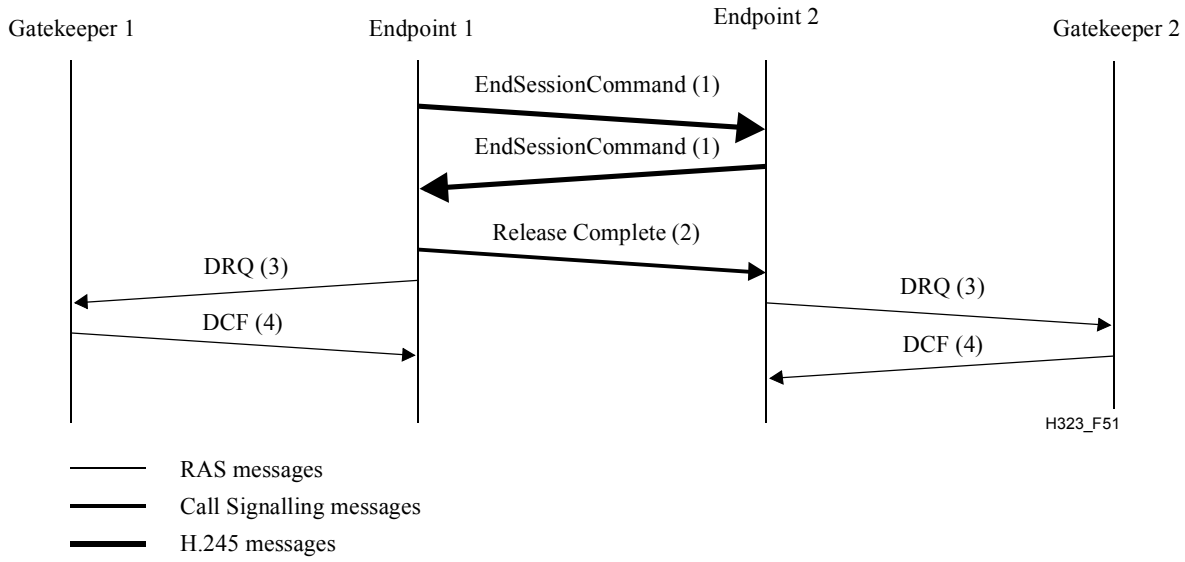
8.5.1 Call clearing without a gatekeeper

In networks that do not contain a Gatekeeper, after steps A-1) to A-5) or B-1) to B-6) above, the call is terminated. No further action is required.

8.5.2 Call clearing with a gatekeeper

In networks that contain a Gatekeeper, the Gatekeeper needs to know about the release of bandwidth. After performing steps A-1) to A-5 or B-1) to B-6) above, each endpoint shall transmit an H.225.0 Disengage Request (DRQ) message (3) to its Gatekeeper. The Gatekeeper shall respond with a Disengage Confirm (DCF) message (4). After sending the DRQ message, the endpoints shall not send further unsolicited IRR messages to the Gatekeeper. See Figure 51. At this point, the call is terminated. Figure 51 shows the direct call model; a similar procedure is followed for the Gatekeeper routed model.

The DRQ and DCF messages shall be sent on the RAS Channel.



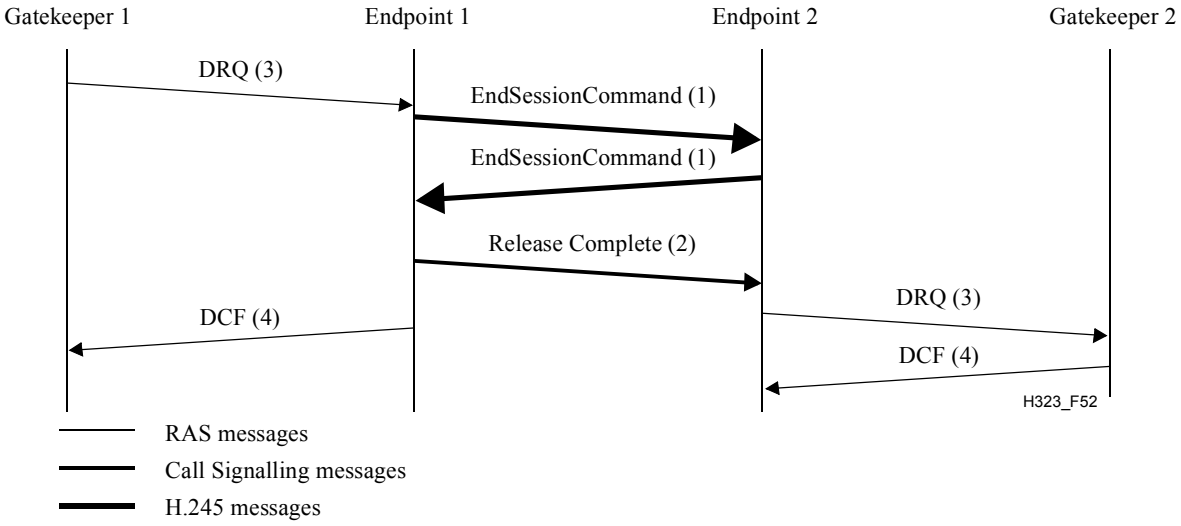
NOTE – Gatekeeper 1 and Gatekeeper 2 may be the same Gatekeeper.

Figure 51/H.323 – Endpoint initiated call clearing (Procedure B)

8.5.3 Call clearing by gatekeeper

The Gatekeeper may terminate call by sending a DRQ to an endpoint. See Figure 52. The endpoint shall immediately follow steps A-1) through A-5) or B-1) through B-6) from above and then reply to the Gatekeeper with DCF. The other endpoint, upon receiving **endSessionCommand**, shall follow the procedure described above. Figure 52 shows the direct call model; a similar procedure is followed for the Gatekeeper routed model.

If the conference is a multipoint conference, the Gatekeeper should send a DRQ to each endpoint in the conference, in order to close the entire conference.



NOTE – Gatekeeper 1 and Gatekeeper 2 may be the same Gatekeeper.

Figure 52/H.323 – Gatekeeper initiated call clearing

8.6 Protocol failure handling

The underlying reliable protocol of the H.245 Control Channel uses appropriate effort to deliver or receive data on the channel before reporting a protocol failure. Therefore, if a protocol failure is reported on the channel, the H.245 Control Channel, and all associated logical channels shall be closed. This shall be done following the procedures of Phase E, as if the other endpoint had issued the H.245 **endSessionCommand**. This includes transmission of the DRQ message to the Gatekeeper and termination of the Call Signalling Channel. In the case where the MC detects failure in a multipoint conference, the MC shall send **terminalLeftConference** messages to the remaining terminals. It is up to the implementation whether or not to try to re-establish the call without user intervention. In any case, this would appear to the other endpoint (and the Gatekeeper) as a new call.

The Call Signalling Channel also uses an underlying reliable protocol. Depending on the routing of the Call Signalling Channel, either the Gatekeeper or an endpoint may detect the protocol failure. If the Gatekeeper detects the failure, it shall attempt to re-establish the Call Control Channel. This implies that the endpoint shall always have the ability to establish a channel on its Call Signalling Channel Transport Address. Failure of the Call Signalling channel shall not change the call state. After re-establishment of the Call Signalling Channel, the Gatekeeper may send a Status message to request the call state of the endpoint to assure that they are in synchronization.

If the endpoint detects the failure, the endpoint may choose to terminate the call as described in Phase E, or it may attempt to re-establish the Call Signalling Channel as described above.

If, during a call, an endpoint wants to determine if the other endpoint is still functioning and connected, it may send the H.245 **roundTripDelayRequest**. Since H.245 Control Channel is carried on a reliable channel, this will result in a response from the other endpoint or an error from the transport interface. In the latter case, the procedures described above shall be used. An endpoint in a multipoint conference may use the same mechanism; however, it will learn only whether it still has a connection to the MC. Note that it is possible for an endpoint to have an error-free connection with the MC but still be receiving no audio or video from the rest of the terminals in the conference.

NOTE – The requirement to close the H.245 Control Channel and all associated logical channels does not apply to equipment that is capable of recovering the H.245 control channel.

9 Interoperation with other terminal types

Interoperation with other terminals shall be accomplished through the Gateway. See 6.3 and ITU-T Rec. H.246.

9.1 Speech-only terminals

Interoperation with speech-only terminals (telephony) over the ISDN or GSTN can be provided by:

1) using a H.323-ISDN speech Gateway;
2) using a H.323-GSTN speech Gateway.

The Gateway should consider the following issues:

– Audio code conversion:
  - ISDN: if desired, since ISDN uses G.711.
  - GSTN: from analogue to G.711.
– Bit stream conversion:
  - ISDN: H.225.0 to/from unframed.
  - GSTN: generate H.225.0.
– Control conversion (generate H.245).
– Call Control Signalling conversion.
– DTMF tone conversion to/from H.245 **userInputIndication** message and RTP payload types (as per 10.5).

9.2 Visual telephone terminals over the ISDN (ITU-T Rec. H.320)

Interoperation with visual telephone terminals over the ISDN (ITU-T Rec. H.320) can be provided by:

– using a H.323-H.320 Gateway.

The Gateway should consider the following issues:

– Video format conversion. (If desired, H.261 is mandatory for both terminal types.)
– Audio code conversion. (If desired, G.711 is mandatory for both terminal types.)
– Data protocol conversion.
– Bit stream conversion. (H.225.0 to/from H.221.)
– Control conversion. (H.245 to/from H.242.)
– Call Control Signalling conversion.
– SBE Number conversion to/from H.245 **userInputIndication** message and RTP payload types (as per 10.5).

9.3 Visual telephone terminals over GSTN (ITU-T Rec. H.324)

Interoperation with visual telephone terminals over the GSTN (ITU-T Rec. H.324) can be provided by two methods:

1) using a H.323-H.324 Gateway;
2) using a H.323-H.320 Gateway, assuming that there exists an H.320-H.324 Gateway in the circuit switched network.

The Gateway should consider the following issues:

– Video format conversion. (If desired, H.261 is mandatory for both terminal types.)
– Data protocol conversion.
– Audio code conversion. (G.711 is mandatory for H.323 terminal, G.723.1 is mandatory for H.324 terminal.)
– Bit stream conversion. (H.225.0 to/from H.223.)
– Call Control Signalling conversion.

9.4 Visual telephone terminals over mobile radio (ITU-T Rec. H.324/M – Annex C/H.324)

For further study.

9.5 Visual telephone terminals over ATM (H.321 and H.310 RAST)

Interoperation with visual telephone terminals over ATM networks (H.321 and H.310 RAST terminals operating in H.320/H.321 interworking mode) can be provided by two methods:

1) using a H.323-H.321 Gateway;
2) using a H.323-H.320 Gateway, assuming that there exists an I.580 ISDN/ATM Interworking Unit in the network.

The Gateway should consider the following issues:

– Video format conversion. (If desired, H.261 is mandatory for both terminal types.)
– Data protocol conversion.
– Audio code conversion. (If desired, G.711 is mandatory for both terminal types.)
– Bit stream conversion. (H.225.0 to/from H.221.)
– Control conversion. (H.245 to/from H.242.)
– Call Control Signalling conversion.

9.6 Visual telephone terminals over guaranteed quality of service LANs (ITU-T Rec. H.322)

Interoperation with visual telephone terminals over Guaranteed Quality of Service LANs (ITU-T Rec. H.322) can be provided by:

– using a H.323-H.320 Gateway, assuming that there exists a GQOS LAN-ISDN Gateway in the network.

The Gateway should consider the following issues:

– Video format conversion. (If desired, H.261 is mandatory for both terminal types.)
– Data protocol conversion.
– Audio code conversion. (If desired, G.711 is mandatory for both terminal types.)
– Bit stream conversion. (H.225.0 to/from H.221.)
– Control conversion. (H.245 to/from H.242.)
– Call Control Signalling conversion.

9.7 Simultaneous voice and data terminals over GSTN (ITU-T Rec. V.70)

Interoperation with Simultaneous Voice and Data Terminals over GSTN (ITU-T Rec. V.70) can be provided by:

– using a H.323-V.70 Gateway.

The Gateway should consider the following issues:

– Audio code conversion. (G.711 to/from Annex A/G.729.)
– Data protocol conversion.
– Bit stream conversion. (H.225.0 to/from V.76/V.75.)
– Control conversion. (Both terminals use H.245.)
– Call Control Signalling conversion.

9.8 T.120 terminals on the packet based network

An H.323 terminal that has T.120 capability should be capable of being configured as a T.120-only terminal which listens and transmits on the standard T.120 well-known TSAP Identifier. This will allow the T.120 capable H.323 terminal to participate in T.120-only conferences.

A T.120-only terminal on the network shall be able to participate in the T.120 portion of multipoint H.323 conferences. See 6.2.7.1.

9.9 Gateway for H.323 media transport over ATM

It is possible to transport H.323 media streams originating from non-ATM IP networks over an ATM network using H.323-to-H.323 Gateways. This mechanism is described in AF-SAA-0124.000 [33].

10 Optional enhancements

10.1 Encryption

Authentication and security for H.323 systems is optional; however, if it is provided, it shall be provided in accordance with ITU-T Rec. H.235.

10.2 Multipoint operation

10.2.1 H.243 control and indication

H.245 contains multipoint control and indication messages carried forward from H.243. These messages may be used to provide certain multipoint capabilities (such as chair control) by following the procedures defined in ITU-T Rec. H.243.

NOTE – Clause 15/H.243 contains guidance for the implementation of these capabilities using the T.120 series of Recommendations.

10.3 Call Linkage in H.323

10.3.1 Description

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

10.3.1.1 General description

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

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

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

10.3.1.2 Service definitions

10.3.1.2.1 Thread identification, thread ID, TID

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

10.3.1.2.2 Global call identification, global call ID, GID

A value assigned to an end-to-end call to uniquely identify that call from end-to-end. If different calls are being transformed into a new call (i.e., due to service interactions), the GIDs of the old calls are updated (if already assigned previously) or assigned by a new GID value for the new end-to-end call.

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

10.3.2 Invocation and operation

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

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

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

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

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

10.3.3 Interaction with H.450 supplementary services

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

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

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

10.3.3.1 Call transfer

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

10.3.3.1.1 Transfer without consultation

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

A new Global Call ID shall be assigned to a transferred call. If a Gatekeeper establishes the transferred call on behalf of a transferred endpoint, the Gatekeeper shall assign the same Global Call ID to the remaining call leg of the primary call. This ensures that the resulting call after successful transfer has one unique GID end-to-end.

10.3.3.1.2 Transfer with consultation

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

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

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

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

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

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

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

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

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

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

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

10.3.3.2 Call diversion

This clause describes the usage of the Call Linkage fields when using ITU-T Rec. H.450.3 [40].

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

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

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

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

10.3.3.3 Call hold and consultation

This clause describes the usage of the Call Linkage fields when using ITU-T Rec. H.450.4.

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

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

A consultation call shall use a new Global Call ID.

10.3.3.4 Call park/call pickup

This clause describes the usage of the Call Linkage fields when using ITU-T Rec. H.450.5 [41].

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

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

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

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

10.3.3.5 Call waiting

There is no interaction with Call Linkage and ITU-T Rec. H.450.6 [42].

10.3.3.6 Message waiting indication

There is no interaction with Call Linkage and ITU-T Rec. H.450.7 [43].

10.3.3.7 Name identification service

There is no interaction with Call Linkage and ITU-T Rec. H.450.8 [44].

10.4 Tunnelling of non-H.323 signalling messages

In order to support existing non-H.323 signalling information in an H.323 system, it is necessary to allow for transport of non-H.323 signalling information in H.323. This clause provides a generic means of tunnelling signalling messages in any H.225.0 call control message.

The procedures of this clause apply to any type of endpoint. Signalling tunnels are terminated in a logical entity called a "tunnel termination". Typically, these tunnel terminations are located in gateways that interconnect parts of a non-H.323 network over a H.323 network as shown in Figure 53. If a Gatekeeper is present in the H.323 network, it may participate in the tunnelling of non-H.323 signalling.

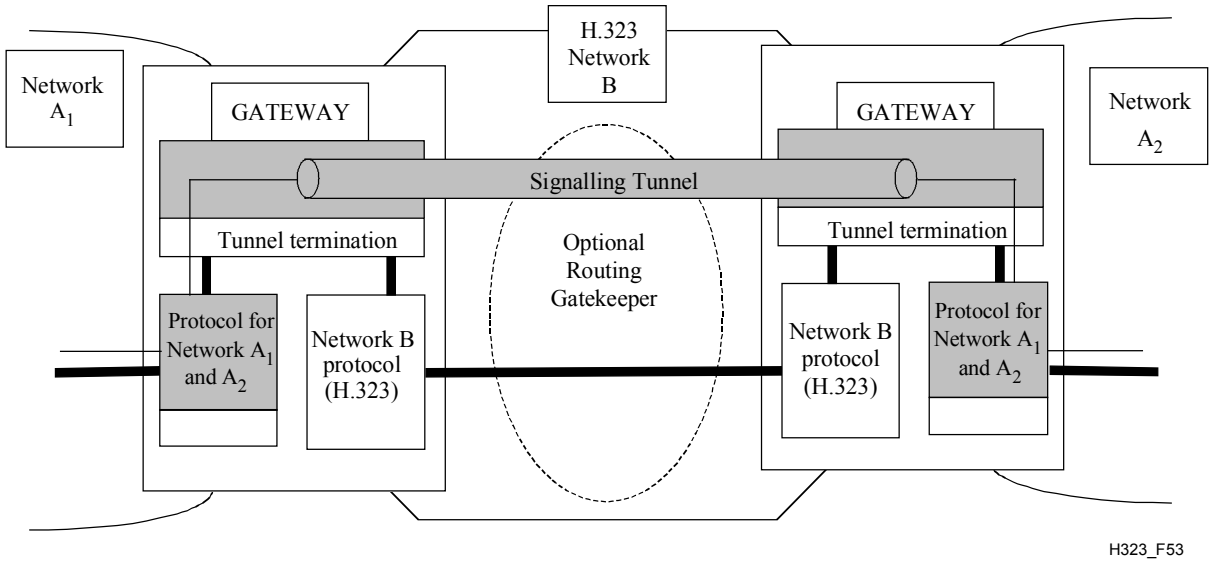


Figure 53/H.323 –Signalling tunnelling between gateways

In some cases, the tunnel termination may be located in a Gatekeeper, as illustrated in Figure 54. Clause 10.4.2 describes Gatekeeper intervention in a tunnel.

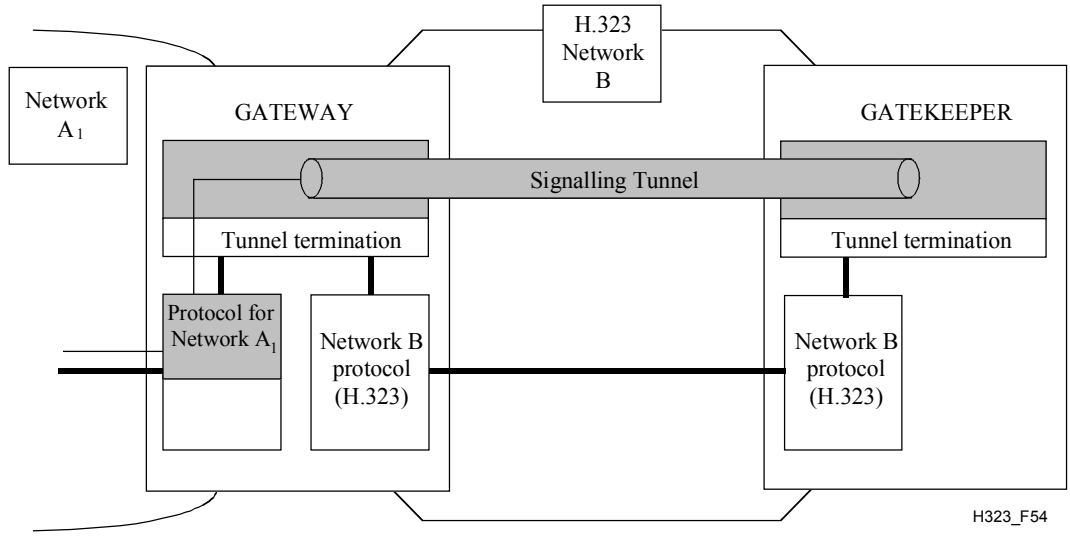


Figure 54/H.323 – Signalling tunnelling between a gateway and an tunnel termination in a gatekeeper

The call control states and procedures of the tunnelled protocol are distinct from the call control states and procedures of the H.225.0 protocol: an endpoint supporting tunnelled signalling should view the two separately.

Any signalling protocol may be tunnelled and is identified by the **TunnelledProtocol**. Examples of signalling protocols that may be tunnelled include:

- QSIG.
- ISUP.
- ISDN DSS1.
- DPNSS.
- Proprietary PBX networking protocol.

10.4.1 Indicating support of tunnelled protocols

Tunnelling support for a prioritized list of protocols is indicated with the **supportedTunnelledProtocols** field of the **EndpointType**. This list consists of a prioritized list of protocols that can be tunnelled.

When registering with its Gatekeeper, an endpoint may indicate the tunnelling protocols supported in the GRQ and RRQ as part of the **EndpointType**. The **EndpointType** contains a prioritized list of supported tunnelled protocols, with the first one being the preferred one. In the ACF or LCF that a Gatekeeper returns in response from an ARQ or LRQ, the **destinationType** indicates the destination's supported tunnelled signalling protocols also in a prioritized list. Since Annex G/H.225.0 imports the **EndpointType** sequence, this capability may also be conveyed through Annex G/H.225.0.

An originating endpoint wishing to indicate the signalling protocols it can tunnel shall include the prioritized list in the **sourceInfo.supportedTunnelledProtocols** in the Setup message. A terminating endpoint wishing to indicate the signalling protocols it can tunnel shall include the prioritized list in the **destinationInfo.supportedTunnelledProtocols** in all the messages including the **destinationInfo** field it sends in response to the Setup message. If an originating endpoint does not receive this indication, it shall assume that the terminating endpoint does not support any tunnelled protocols.

10.4.2 Requesting a specific protocol tunnel to a gatekeeper

An entity may request a specific protocol tunnel to a Gatekeeper by specifying the particular protocol in the **desiredTunnelledProtocol** field in an ARQ or LRQ.

10.4.3 Tunnelling a signalling protocol in H.225.0 call signalling messages

An endpoint may tunnel a signalling protocol by including the **tunnelledSignallingMessage** in any H.225.0 call signalling message. However, it is not recommended to tunnel a signalling protocol in H.225.0 call signalling messages that are not of end-to-end significance, such as Call Proceeding, since the information may not be received by the other end.

If an endpoint will only allow the call to proceed if tunnelling is supported, it shall set the **tunnellingRequired** flag in the Setup message; the **tunnellingRequired** flag shall not be included in any other message than Setup. If an endpoint receives a **tunnelledSignallingMessage** with the **tunnellingRequired** flag set in the Setup message and is not able to tunnel the protocol, it shall terminate the call by sending a Release Complete with a **reason** of **tunnelledSignallingRejected**; a **tunnellingRequired** flag in any other message than Setup shall be ignored.

The tunnelled protocol information is included in the **messageContent** field and the **tunnelledProtocolID** field identifies the protocol being tunnelled. Only a single protocol can be tunnelled in an H.323 call. Multiple tunnelled messages of the same protocol may be aggregated in one single H.225.0 call signalling message.

The tunnel shall be released using the normal H.323 release procedures.

The call signalling procedures of H.225.0 can be used to establish a call independent signalling connection between the peer endpoints. Tunnelling can be used in this context to provide bearer independent signalling for the tunnelled protocol. In this case, no H.245 Control Channel and no media channels are required. A bearer capability information element should be included in the H.225.0 Setup message and coded as described in Table 2/H.450.1. The Setup message used for call independent procedures shall include a **conferenceGoal** within Setup set to value **callIndependentSupplementaryService**. These call independent signalling connection procedures for tunnelling shall not be used in conjunction with an H.450 supplementary service in the same call independent signalling connection.

10.4.4 Gatekeeper considerations

In a direct routed call model, the Gatekeeper is not involved in the H.225.0 call control signalling and therefore does not perform signalling tunnelling in H.225.0. Such Gatekeepers do not affect tunnelling between two endpoints supporting signalling tunnelling. In a Gatekeeper routed model, the Gatekeeper participates in providing a tunnel between peer endpoints by passing on received tunnelled signalling information. The Gatekeeper may also utilize the Facility or Progress message to convey tunnelled messages, as discussed in 8.2.2.

In the Gatekeeper routed model, the Gatekeeper may intercept and act on tunnelled signalling messages. Termination of a signalling tunnel is performed by a tunnel termination function, which, as described earlier, can be located in the Gatekeeper. What the Gatekeeper does with the tunnelled protocol is outside the scope of this Recommendation. However, if the Gatekeeper is capable of providing non-H.323 signalling service, it may terminate the signalling tunnel and generate appropriate H.225.0 messages for the endpoints involved in the call. Alternatively, it may modify the tunnelled signalling information: if it does, it is taking the responsibility of terminating and initiating the tunnelled protocol. A Gatekeeper that does not understand the tunnelled protocol, or does not intend to act on the tunnelled protocol or provide any services in that plane, shall pass the tunnelled signalling message through unchanged to preserve the integrity of the tunnelled protocol.

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

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

RFC 2833 [58] describes means for transporting these tones and events over RTP. An endpoint may indicate support for receiving these RFC 2833 tones and events by including the **receiveRTPAudioTelephonyEventCapability** or the **receiveRTPAudioToneCapability** in the terminal capability set. Alternatively, an endpoint may indicate support for RFC 2833 tones and events by including the **audioTelephonyEvent** or the **audioToneAudioCapability** in the terminal capability set. When using fast connect procedures, these capabilities can be sent using parallelH245 procedures of 8.2.4.

Named telephone events are a logical description of DTMF tones, fax-related tones, standard subscriber line tone, country-specific tones and trunk events. A decimal number identifies each event. When telephone events are used, support for the following DTMF is mandatory: 0-9, #, *, A, B, C, D. All others are optional.

Telephony tones are a description of the waveform properties. This is useful in cases where it is necessary to accurately reproduce non-standard tones.

After a logical channel has been opened for the media stream, the sender may send any of the telephony events or tones advertised by the receiver in the terminal capability set on that same logical channel using the RTP payload type negotiated in the terminal capability set negotiation.

If an endpoint sends DTMF information, it may send it in a **UserInputIndication** and/or using RTP payload for DTMF digits, telephony tones, and telephony signals.

If the DTMF is sent both via RTP and in a **UserInputIndication** in alphanumeric form, it shall be encoded in the **extendedAlphanumeric** structure and the **rtpPayloadIndication** field shall be included. If the DTMF is sent both via RTP and in a **UserInputIndication** in the signal form, the **rtpPayloadIndication** field shall be included in the **signal** structure. If the DTMF is sent only in alphanumeric form, it shall be encoded in the **alphanumeric** field. If the DTMF is sent only in signal form, the **rtpPayloadIndication** field shall not be included.

RFC 2833 shall not be used to relay fax information in H.323 systems. Instead, the procedures defined in Annex D shall be followed for endpoints that wish to transmit T.38 fax information.

NOTE – H.323 entities prior to version 4 did not have the capability of sending DTMF information via RTP as described in this clause. Therefore, all entities shall support the ability to send DTMF information via the **UserInputIndication** message.

11 Maintenance

11.1 Loopbacks for maintenance purposes

Some loopback functions are defined in ITU-T Rec. H.245 to allow verification of some functional aspects of the terminal, to ensure correct operation of the system and satisfactory quality of the service to the remote party.

The **systemLoop** request and **logicalChannelLoop** request shall not be used. The **mediaLoop** request is optional. An endpoint that receives the **maintenanceLoopOffCommand** shall turn off all loopbacks currently in effect.

For the purpose of loopbacks, two modes are defined:

a) Normal operation mode: No loopback. Indicated in **a)** of Figure 55. This shall be the default mode, and the mode entered when the **maintenanceLoopOffCommand** is received.

b) Media loop mode: Loopback of media stream at the analogue I/O interface. Upon receiving the **mediaLoop** request as defined in ITU-T Rec. H.245, loopback of the content of the selected logical channel shall be activated as close as possible to the analogue interface of the video/audio codec towards the video/audio codec, so that decoded and re-coded media content is looped, as indicated in **b)** of Figure 55. This loopback is optional. It should be used only when a single logical channel containing the same media type is opened in each direction. Operation when multiple channels are opened in the return direction is undefined.

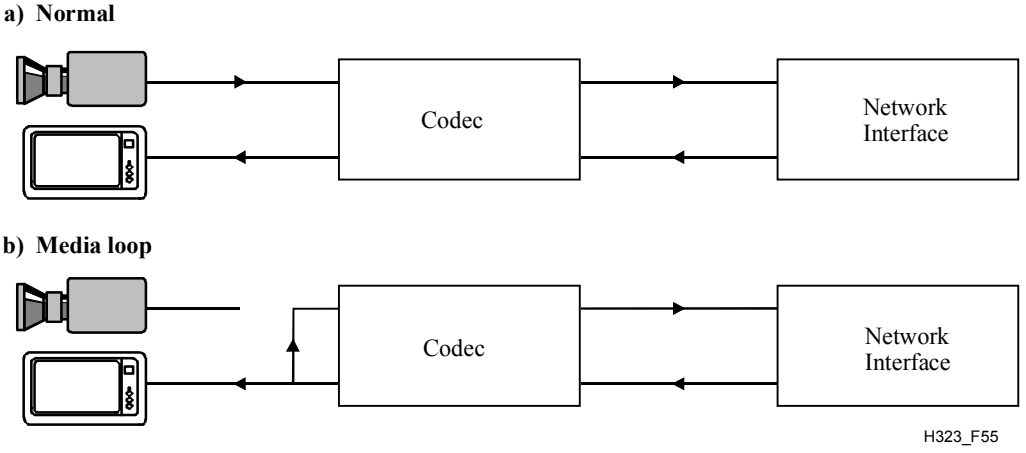


Figure 55/H.323 – Loopback

A Gateway to H.324, which receives an H.245 **systemLoop** request, H.245 **logicalChannelLoop** request, or a Gateway to H.320, H.321, or H.322, which receives an H.230 Dig-Loop command from an SCN endpoint may perform the appropriate loopback function within the Gateway. The Gateway shall not pass these requests to the network endpoint. A Gateway to H.324, receiving H.245 **mediaLoop** from an SCN endpoint shall pass the request to the network endpoint. A Gateway to H.320, H.321, or H.322, receiving H.230 Vid-loop or Aμ-loop command from an SCN endpoint shall convert it to the appropriate H.245 **mediaLoop** request and send it to the network endpoint.

A Gateway to H.320, H.321, or H.322, which receives an H.245 **mediaLoop** request from a network endpoint shall convert it to the appropriate H.230 Vid-loop or Aμ-loop command and send it to the SCN endpoint.

A Gateway to H.324 may send an H.245 **systemLoop** request or H.245 **logicalChannelLoop** request to the SCN endpoint. A Gateway to H.320, H.321, or H.322 may send an H.230 Dig-Loop command to the SCN endpoint. If a network endpoint is in a call to the SCN endpoint, the audio and video sent to the network endpoint may be the looped back audio or video, pre-recorded audio or video message indicating the loopback condition, or no audio or video.

11.2 Monitoring methods

All terminals shall support the Information Request/Information Request Response (IRQ/IRR) message of ITU-T Rec. H.225.0. The Information Request Response message contains the TSAP Identifier of all channels currently active on the call, including T.120 and H.245 control, as well as audio and video. This information can be used by third party maintenance devices to monitor H.323 conferences to verify system operation.

Annex A

H.245 messages used by H.323 endpoints

The following rules apply to the use of H.245 messages by H.323 endpoints:

- An endpoint shall not malfunction or otherwise be adversely affected by receiving H.245 messages that it does not recognize. An endpoint receiving an unrecognized request, response, or command shall return "function not supported". (This is not required for indications.)
- The following abbreviations are used in Tables A.1 to A.12:

  M Mandatory.

  O Optional.

  F Forbidden to transmit.
- A message marked as mandatory for the receiving endpoint indicates that the endpoint shall accept the message and take the appropriate action. A message marked as mandatory for the transmitting endpoint indicates that the endpoint shall generate the message under the appropriate circumstances.

Table A.1/H.323 – Master-slave determination messages

Message	Receiving endpoint status	Transmitting endpoint status
Determination	M	M
Determination Acknowledge	M	M
Determination Reject	M	M
Determination Release	M	M

Table A.2/H.323 – Terminal capability messages

Message	Receiving endpoint status	Transmitting endpoint status
Capability Set	M	M
Capability Set Acknowledge	M	M
Capability Set Reject	M	M
Capability Set Release	M	M

Table A.3/H.323 – Logical channel signalling messages

Message	Receiving endpoint status	Transmitting endpoint status
Open Logical Channel	M	M
Open Logical Channel Acknowledge	M	M
Open Logical Channel Reject	M	M
Open Logical Channel Confirm	M	M
Close Logical Channel	M	M
Close Logical Channel Acknowledge	M	M
Request Channel Close	M	O
Request Channel Close Acknowledge	O	O
Request Channel Close Reject	O	M
Request Channel Close Release	O	M

Table A.4/H.323 – Multiplex table signalling messages

Message	Status
Multiplex Entry Send	F
Multiplex Entry Send Acknowledge	F
Multiplex Entry Send Reject	F
Multiplex Entry Send Release	F

Table A.5/H.323 – Request multiplex table signalling messages

Message	Status
Request Multiplex Entry	F
Request Multiplex Entry Acknowledge	F
Request Multiplex Entry Reject	F
Request Multiplex Entry Release	F

Table A.6/H.323 – Request mode messages

Message	Receiving endpoint status	Transmitting endpoint status
Request Mode	M	O
Request Mode Acknowledge	M	O
Request Mode Reject	O	M
Request Mode Release	O	M

Table A.7/H.323 – Round trip delay messages

Message	Receiving endpoint status	Transmitting endpoint status
Round Trip Delay Request	M	O
Round Trip Delay Response	O	M

Table A.8/H.323 – Maintenance loop messages

Message	Receiving endpoint status	Transmitting endpoint status
Maintenance Loop Request		
System Loop	F	F
Media Loop	O (Note)	O (Note)
Logical Channel Loop	F	F
Maintenance Loop Acknowledge	O	O
Maintenance Loop Reject	O	M
Maintenance Loop Command Off	M	O
NOTE – Mandatory in Gateways.		

Table A.9/H.323 – Conference request and response messages

Message	Receiving endpoint status	Transmitting endpoint status
Terminal List Request	O	O
Drop Terminal	O	O
Make Me Chair	O	O
Cancel Make Me Chair	O	O
Enter H.243 Password	O	O
Enter H.243 Terminal Id	O	O
Enter H.243 Conference ID	O	O
Request Terminal ID	O	O
Terminal ID Response	O	O
MC Terminal ID Response	O	O
Enter Extension Address	O	O
Enter Address Response	O	O
Terminal List Response	O	O
Make Me Chair Response	O	O
Conference ID Response	O	O
Password Response	O	O

Table A.10/H.323 – Commands

Message	Receiving endpoint status	Transmitting endpoint status
Send Terminal Capability Set	M	M
Encryption	O	O
Flow Control	M	O
End Session	M	M
Miscellaneous Commands		
Equalize Delay	O	O
Zero Delay	O	O
Multipoint Mode Command	M	O
Cancel Multipoint Mode Command	M	O
Video Freeze Picture	M	O
Video Fast Update Picture	M	O
Video Fast Update GOB	M	O
Video Fast Update MB	M	O
Video Temporal Spatial Trade Off	O	O
Video Send Sync Every GOB	O	O
Video Send Sync Every GOB Cancel	O	O
Terminal ID Request	O	O
Video Command Reject	O	O
Make Me Chair Response	O	O
Conference Commands		
Broadcast My Logical Channel Me	O	O
Cancel Broadcast My Logical Channel Me	O	O
Make Terminal Broadcaster	O	O
Cancel Make Terminal Broadcaster	O	O
Send This Source	O	O
Cancel Send This Source	O	O
Drop Conference	O	O

Table A.11/H.323 – Conference mode commands

Message	Receiving endpoint status	Transmitting endpoint status
Communication Mode Command	M	O
Communication Mode Request	O	O
Communication Mode Response	O	O

Table A.12/H.323 – Indications

Message	Receiving endpoint status	Transmitting endpoint status
Function Not Understood	M	M
Function Not Supported	M	M
Miscellaneous Indication		
Logical Channel Active	O	O
Logical Channel Inactive	O	O
Multipoint Conference	M	O
Cancel Multipoint Conference	M	O
Multipoint Zero Comm	O	O
Cancel Multipoint Zero Comm	O	O
Multipoint Secondary Status	O	O
Cancel Multipoint Secondary Status	O	O
Video Indicate Ready to Activate	O	O
Video Temporal Spatial Trade Off	O	O
Video Not Decoded MBs	O	O
Conference Indications		
SBE Number	O	O
Terminal Number Assign	M	O
Terminal Joined Conference	O	O
Terminal Left Conference	O	O
Seen By At Least One Other	O	O
Cancel Seen By At Least One Other	O	O
Seen By All	O	O
Cancel Seen By All	O	O
Terminal You Are Seeing	O	O
Request For Floor	O	O
Vendor Indications	O	O
MC Location Indication	M	O
Jitter Indication	O	O
H.223 Skew Indication	F	F
H2250MaximumSkewIndication	O	M
New ATM Virtual Channel Indication	F	F
User Input	M (for 0-9, *and #)	M (for 0-9, *and #)

Non-standard commands, requests, etc. are allowed.

Annex B

Procedures for layered video codecs

B.1 Scope

This annex describes enhancements within the framework of the H.323 specification, to incorporate layered video codecs. The described procedure is scalable for multipoint conferences.

B.2 Introduction

Layered video coding is a technique that allows the video information to be transmitted in multiple data streams in order to achieve scalability. These may provide bandwidth scalability, temporal scalability, SNR scalability, and/or spatial scalability. Annex O/H.263 describes the use of layered coding within H.263. Conferences can take advantage of this feature to service connected endpoints that have different capabilities, using one bitstream. This will allow more efficient use of network bandwidth.

B.3 Scalability methods

Scalability of a video stream refers to the generation of a stream that may only be decoded in part due to limitations of available resources. Scalability may be desired to overcome limitations of available computing power or to accommodate bandwidth limitations.

There are three types of scaling: Temporal, Signal-to-Noise Ratio (SNR), and Spatial that are available in ITU-T Rec. H.263. Other video codecs may have similar layering capability. All of these methods can be used separately or together to create a multi-layer scalable bit stream. The resolution, frame rate, and quality of the image can only increase by adding scaling layers. The base layer can be used to guarantee a minimum level of image quality. Endpoints can then use additional layers to add image quality by increasing frame rate, display frame size, or accuracy of decoded images. Allowing multiple scaling methods in a conference can add resource efficiency, especially when endpoints participating have varying processing and bandwidth capabilities. This is especially true for multipoint and loosely-coupled conferences.

B.4 Call establishment

H.323 call establishment takes place following the same procedures described in clause 8. The layered coding capability will be signalled using the H.245 capabilities exchange methods. Codepoints within H.245 exist which clearly identify what layering methods are supported by the endpoints. The endpoints shall use these capabilities in order to signal the exact layering methods they support.

The use of simultaneous capabilities methods in H.245 shall be used to indicate which layering methods will be used together to create the video layers when they are going to be sent in two or more logical channels. It is also possible to send two or more layers in single logical channels. The exact video layers that will be used are signalled during the **openLogicalChannel** in the same manner that is currently used to indicate what video **dataType** will be used, except that the endpoint shall indicate dependencies between the base layer logical channel and the enhancement layer logical channels.

B.5 Use of RTP sessions and codec layers

It is desired to allow separate RTP sessions for the different qualities of video that are available. The base layer should be considered the primary video session, and its level considered the minimum quality of video that is available in the conference. Enhancement layers can be sent on separate RTP sessions. The **forward/reverseLogicalChannelDependency** parameter, added to

H.245 **openLogicalChannel** command, shall be used to indicate how the video layers are organized. This is outlined in the following clauses. RTP Timestamps must be the same in the base and all dependent enhancement layers corresponding to a frame to allow reassembly and proper display.

B.5.1 Associate base to audio for lip synchronization

The base video session should be associated with the audio session corresponding with the audio track of the video, for lip synch purposes. This is done in the same manner that existing non-layered video sessions are associated with their corresponding audio. This is done using the **associatedSessionID** and the **sessionID** parameters located in the **H2250LogicalChannelParameters**. The enhancement layers may also be associated with the audio or with the base layer using the **associatedSessionID**. Coding dependency shall be indicated using the **forwardLogicalChannelDependency** and the **reverseLogicalChannelDependency** parameter in the **openLogicalChannel** command as explained below.

B.5.2 Enhancement layer dependency

Enhancement layer dependency can create many complex cases using multiple layers that contain multiple enhancement frame types. Dependency between layers shall be indicated using the **forward/reverseLogicalChannelDependency** parameter, added to H.245 **openLogicalChannel** command. Dependency is used to indicate that the data sent on the logical channel cannot be used without the contents of the logical channel it is dependent on. Enhancement layers, by definition, must be differentially coded from the video layer they are enhancing and are therefore dependent on that video layer for meaningful decoding. If an enhancement layer is sent on a separate logical channel, it shall indicate the layer it was differentially coded from in the **forward/reverseLogicalChannelDependency** parameter.

Since the **forward/reverseLogicalChannelDependency** parameter allows the indication of a single logical channel, the logical channels need to be opened in order of dependence starting with the base layer. An endpoint shall have either sent or received the **openLogicalChannelAck** for any logical channel that is used in a **forward/reverseLogicalChannelDependency** parameter. An endpoint shall send an **openLogicalChannel** for a dependent logical channel, only after the logical channel on which it is dependent is opened and acknowledged. Logical channels that have common dependency may be opened in parallel. Enhancement layers must be indicated to be dependent on the highest layer that is required for proper decoding.

Assuming that separate RTP sessions are used for each layer, an example can be built as shown in Figure B.1.

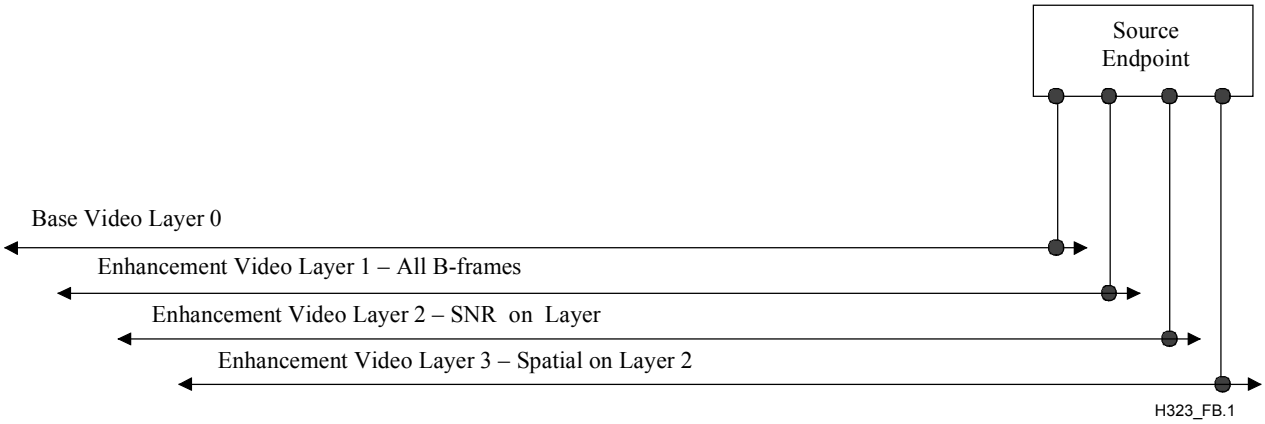


Figure B.1/H.323 – Model with layered video

In this example, layered video is created that has four layers:

1) The base video, not dependent on any other layer. This is associated with its corresponding audio.
2) Enhancement level one consisting of B-frames, dependent on the base video. This is indicated to be dependent on the base video session, Layer 0.
3) Enhancement level two that is SNR enhancement of the base video, dependent only on the base video, Layer 0. This is indicated to be dependent on the base video session.
4) Enhancement level three that consists of spatial enhancement of enhancement level two, dependent on Layer 2, which implies the base is also required. This is indicated to be dependent on the video in Layer 2.

In this example, the base video logical channel must be opened first. The **openLogicalChannel** for enhancement Layers 1 and 2 may be sent in parallel, only after receiving the **openLogicalChannelAck** for the base video logical channel. The **openLogicalChannel** for enhancement Layer 3 can only be sent after the **openLogicalChannelAck** has been received or sent for the logical channel used for enhancement Layer 2.

B.6 Possible layering models

There are many possible methods for layering of the video and organization of the corresponding RTP sessions. The reason that the layers may need to be separated is that they are used for either decoder power scaling or for bandwidth usage scaling. It may be desirable to separate all non-B-frames into separate layers that can be discarded if they cannot be used. An important feature of the layered codec is that at any time an endpoint may discard any or all enhancement layers, without affecting the quality of the base video, in order to provide decoder power scaling.

In a similar manner, the layers may need to be organized into bandwidth usage levels that correspond to the bandwidths reported by the endpoints that are connected to the conference. This would allow the conference to accommodate multipoint conferences that have endpoints using connection methods that may limit the available bandwidth and create a layer that gives them the best possible video at that bandwidth. The endpoint may add or subtract layers as its available bandwidth varies up and down.

B.6.1 Multiple logical channels and RTP sessions for a layered stream

If bandwidth scaling is the goal of using layering, each layer should flow on a separate logical channel with a separate RTP session. This means that what is a single video source will now have to be coordinated amongst multiple logical channels and RTP sessions.

If the goal of layering is processor-power scaling, the enhancement layers can be sent, with the base video on a single logical channel and RTP session.

If the goal is a mixture of bandwidth and processor-power scaling, then groups of enhancement layers, sent in logical channels on a group basis can be sent. The choice of layers and grouping is a choice based on system need. The method used to make these choices is an implementation issue and outside the scope of this Recommendation.

B.6.2 Impact of one layer per logical channel and per RTP session

The impact of using a single logical channel and RTP session for each layer is that the encoder and decoder are burdened with having to split and reassemble the video stream according to the chosen layering model. This model is signalled to the receiving side so that it can properly interpret the layer information. It is signalled using H.245 capabilities, with a capability per logical channel that, when combined with the dependencies, will sufficiently describe the layering model. Possible layering models are signalled during capabilities exchange, using the simultaneous capabilities feature of ITU-T Rec. H.245.

Strict timing consideration will need to be used to ensure that the layers are properly synchronized. For H.323, this will be handled in the RTP payload format.

B.7 Impact on multipoint conferences

The most likely envisaged usage of video layering is in multipoint conferences. In H.323, this can be performed by a centralized MCU, used for audio mixing and video switching, or using a decentralized model, with each endpoint responsible for video switching and audio mixing. In either case, the MC should perform the function of reporting what the layering model is for the conference. This is done using the **communicationModeCommand**.

In order for an endpoint to receive a video layer, a logical channel containing that layer must be opened. The decision to open a logical channel can be made by either the MC or the endpoint sending an **openLogicalChannel**. If an MC or endpoint decides not to open a logical channel, it must reject the **openLogicalChannel** when it is offered. The MC or endpoint can only offer a logical channel that corresponds to a **dataType** that is supported by the receiving endpoint.

When implementing support for layer codecs, an MC can take two approaches. If the MC does not make any decisions as to what logical channels will be opened, it can be called the "MC Impartial" model. In this model the MC offers all media to all endpoints without regard to any reported QOS. When the MC makes the decision to strictly enforce QOS, it is called the "MC Decision" model. These models are explained further below.

B.7.1 MC Impartial model

The MC Impartial model does not depend on the QOS capability set additions and as such may allow for a simpler MC implementation. In this case, the endpoint must judge whether it has sufficient bandwidth to accept logical channels offered by the MC. If it will exceed the transmission capabilities of the endpoint or the underlying network, then the endpoint may reject the logical channel. This method will require the endpoint to have knowledge of the network bandwidth available. The MC should indicate all available media in the **communicationModeCommand**.

B.7.2 MC Decision model

The MC Decision model depends upon the addition of Quality of Service (QOS) capabilities to the Terminal Capability Set. This has been previously proposed and is work in progress. The MC can then examine the QOS capabilities of the endpoints and only offer logical channels that are within the QOS of the endpoint. The endpoint will need to determine its available QOS at the start of the conference and indicate this using the QOS capabilities defined by work in progress.

In the MC Decision model, the MC may send a **communicationModeCommand** to an endpoint that only shows the sessions within the endpoint's QOS capabilities. In this way, the MC can strictly enforce bandwidth usage.

B.7.3 Multipoint conference containing endpoints on different bandwidths

In the model where the multipoint conference contains endpoints that have different bandwidth capabilities, the layering will need to be tuned to match these bandwidth levels. This can be done by using two possible models. One is illustrated in Figure B.2.

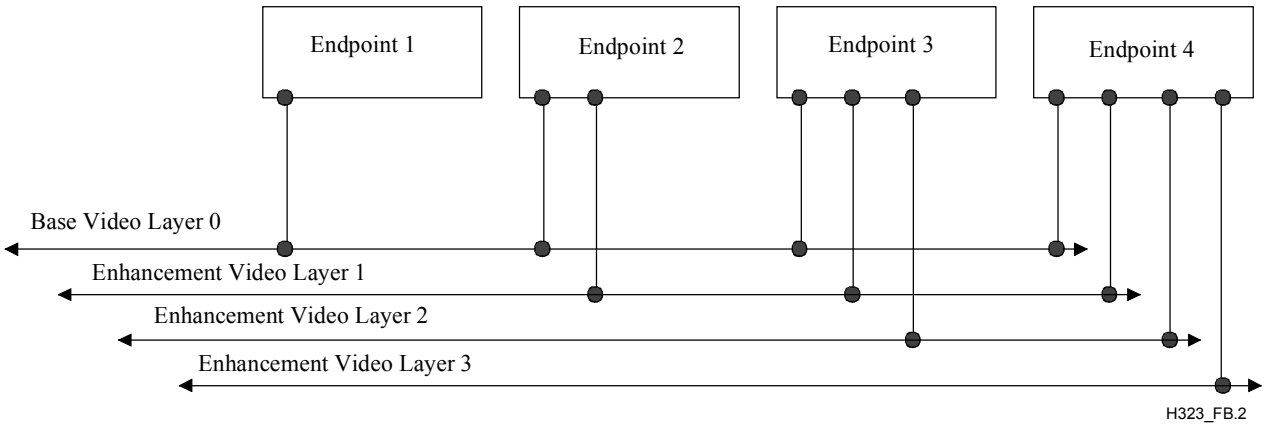


Figure B.2/H.323 – Endpoints attached to one or more layers according to bandwidth

In this case, the endpoints are attached to the base layer of video and the enhancement layers up to the total bandwidth desired. Each enhancement layer is on a separate logical channel. The endpoints are burdened with recombination of the layers to create the video stream. The sending endpoint must have capability for the combined bandwidth of all streams it sources. In this case, each endpoint may have communicated a different set of capabilities. The MC will examine the capabilities and QOS and create a layering model that is likely to provide the best use of the endpoints capabilities and bandwidth. This layering is indicated in the **communicationModeCommand** by the indication of **sessionDependency** in the **communicationModeTableEntry**. The **sessionDependency** field is set by the MC to indicate when a session is dependent on another session for meaningful decoding of its data. This information will be translated into **logicalChannelNumbers** when opening a dependent logical channel, according to the actual logical channels that are opened.

In the above case, using the MC Decision model, the MC will then offer the endpoints the logical channels that correspond to the layers that match the endpoint's capabilities. The MC will offer Endpoint 1 only the logical channel corresponding to the Base Video Layer. Endpoint 2 will be offered the logical channels corresponding to base video and enhancement video Layer 1. Endpoint three is offered three logical channels corresponding to the base video and two enhancements layers, and Endpoint 4 is offered all video logical channels.

In the MC impartial case, the MC will offer all logical channels, to all endpoints, that are within their **dataType** capabilities. The endpoints will refuse any logical channel that will cause them to exceed their bandwidth capabilities.

A second layering model is shown in Figure B.3. In this model each logical channel contains a totally independent video stream.

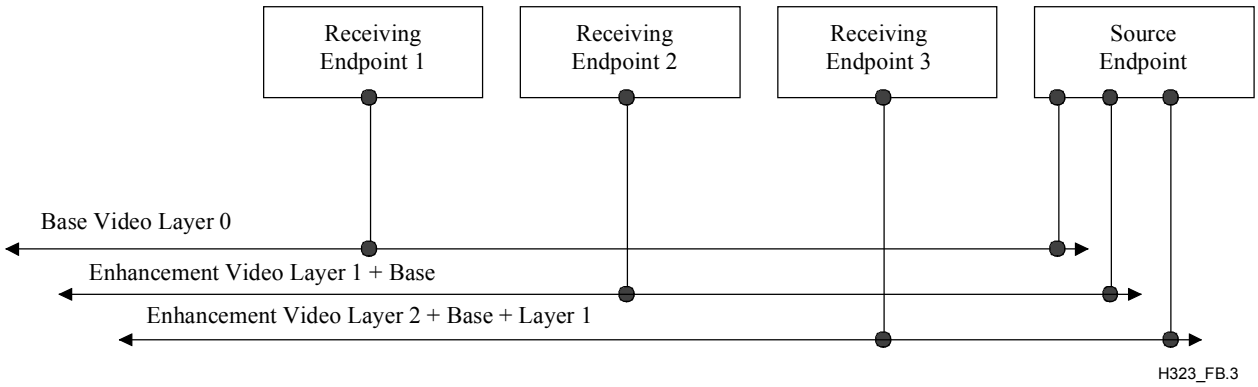


Figure B.3/H.323 – Endpoints attached to single Layer according to bandwidth

In this case, the endpoint shall connect only to the logical channel that corresponds to the bandwidth it has available. This stream has all layers that build the video stream to the bandwidth of the logical channel. This method eliminates the burden from the endpoints to recombine the video, but burdens the sender with producing several video streams. This is a less efficient use of network resources, since enhancement layers include all lower layers.

In order to perform proper lip synch, any session containing base video should be associated with the audio session corresponding to its audio track, using the **associatedSessionID** in the **H2250LogicalChannelParameters**. In the example shown in Figure B.2, the base video session should be associated with the audio session for lip synch. In the example shown in Figure B.3, all three video sessions should be associated with the audio session for lip synch, since all three contain base video.

B.8 Use of network QOS for layered video streams

Several important characteristics of the nature of layered coding usage should be considered when using network QOS for delivery of layered coded video streams. An enhancement layer cannot be decoded properly without receiving the layers on which it is dependent. Enhancement video layers may be discarded without affecting the decoding of the layer on which they are dependent.

If available, network QOS may be used to help guarantee that a video stream will be delivered by the network. Since layered video may be delivered using multiple streams, delivered on separate network connections, different QOS can be used on each video layer. QOS used on layered video streams should be specified when the logical channel is opened.

It is important that a dependent video layer has the information on which they are dependent at the time the dependent layer is to be decoded. This leads to general rules regarding use of QOS:

1) Dependent layers that are delivered using network QOS should have the layer they are dependent on, also delivered using QOS.

2) The base layer should be delivered using network QOS, if any other video layers in the conference are to be delivered using QOS.

3) The nearer the video layer is to the base layer, the stronger the delivery guarantees should be.

Annex C

H.323 on ATM

C.1 Introduction

This is an optional enhancement allowing H.323 endpoints to establish QOS-based media streams on ATM networks using AAL 5.

C.2 Scope

This annex specifies an improved method of using H.323 on AAL 5. H.323 can always be used on ATM by making use of an IP over ATM method. However, this is less efficient than using AAL 5 Virtual Channels (VCs) directly for the transport of the audio and video streams of H.323. When the media streams flow directly on AAL 5, they can benefit from a QOS-based ATM VC.

This annex retains the use of a packet network protocol for H.245 and H.225.0 communications to ensure interoperability with H.323 endpoints that are using a packet network protocol for all streams (whether over ATM or other media). Interoperability with legacy H.323 endpoints is achieved, without the use of a Gateway, by first requiring the basic mode of operation, in which an endpoint sends media streams on a datagram service using a packet network protocol, for example UDP/IP over ATM. In basic mode, unless a packet network protocol infrastructure has been upgraded, QOS may not be available from the network.

C.2.1 Point-to-point conferencing

This annex specifies a method of point-to-point communication between two H.323 endpoints using AAL 5 VCs for the media streams. The protocol necessary for entering into this mode is specified, as are information elements to be used in ATM signalling.

C.2.2 MCU-based multipoint

It follows that multipoint MCU-based communications can occur among several H.323 endpoints using AAL 5 VCs for the media streams. Currently no support is specified for the H.323 Decentralized Multipoint using ATM point-to-multipoint capability. This is left for further study.

C.2.3 H.323 interoperability with endpoints using IP

Interoperability is guaranteed with an endpoint using IP for the entire H.323 connection. This annex defines methods that allow an endpoint to detect if support is present for the option of using AAL 5 directly. An endpoint conforming to this annex must accept that the audio and video streams may occur on either AAL 5 VCs or UDP/IP ports.

C.3 Architecture

The basic protocol architecture of the system is shown in Figure C.1. It uses IP on ATM for delivery of the H.225.0 and H.245 messages and for the RTCP part of the audio and video streams. It uses AAL 5 directly for the RTP part of the audio and video streams.

NOTE – The H.323 media streams, compressed into variable length packets according to ITU-T Rec. H.225.0, are easily mapped to AAL 5. It would be difficult to map them to AAL 1, and this alternative has no clear benefit.

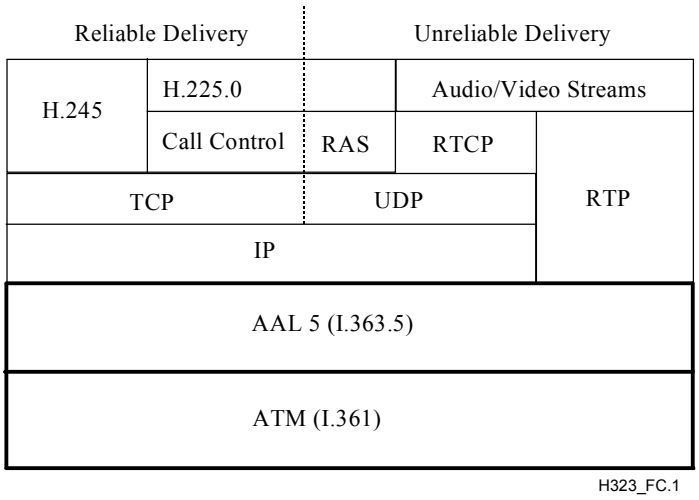


Figure C.1/H.323 – Architecture for H.323 on ATM-AAL 5

C.3.1 Overview of system

The system architecture is designed to make use of H.323 and its component protocols as they are presently specified. It is further designed to use commonly available services of AAL 5 on ATM.

C.3.2 Interoperation with other ITU-T H-series Recommendations endpoints

Interoperation with other H-series endpoints shall be done through the use of Gateway devices as described in ITU-T Rec. H.323. Gateway vendors will need to support the methods described in this annex, if they wish to support the direct use of AAL 5 VCs by H.323 endpoints.

It should be noted that interoperation with other IP-based H.323 endpoints does not require a Gateway.

C.3.3 H.225.0 on IP over ATM

H.225.0 communication requires TCP/IP and UDP/IP using one of the available methods for IP over ATM. No preference is expressed here for which method of IP over ATM to use. If two endpoints on the same network segment use different IP over ATM methods, they must rely on IP routers to forward their packets.

The endpoint shall listen on the well-known TCP ports identified in ITU-T Rec. H.225.0. If the endpoint is being used on a network with a Gatekeeper, the endpoint should use the methods described in ITU-T Rec. H.225.0 to discover and register with the Gatekeeper. This requires the support of UDP multicast. If multicast is not available on the network, the endpoint may be pre-configured with the Gatekeeper(s) address(es).

The methods outlined in ITU-T Rec. H.225.0, combined with an IP over ATM method, shall be used to establish the H.245 control channel on TCP/IP.

C.3.4 H.245 on TCP/IP over ATM

Once the reliable H.245 control channel has been established using methods described in ITU-T Rec. H.225.0, additional channels for audio, video, and data are established based on the outcome of the H.245 capability exchange using H.245 open logical channel procedures.

C.3.5 Addressing for A/V streams

H.323 has the capability for the audio and video streams to be established to a different address than the H.245 control channels. This is fortunate since a TCP/IP channel is established to an IP address, and the audio and video, optionally, are to be sent on RTP over AAL 5 directly to an ATM address.

H.323 also has the capability for the RTCP stream to be addressed separately from the RTP stream. The RTCP stream shall continue to be addressed to an IP address, even though the RTP stream is addressed to an ATM address.

C.3.6 Transport Capabilities added to TransportCapability Set

For operation of H.323 on AAL 5, an addition to the **TransportCapability** set is made in H.245. This includes transport level capabilities such as support for ATM Transfer Capability (DBR, SBR1, SBR2, SBR3, ABT/DT, ABT/IT, ABR) as defined in ITU-T Rec. I.371. Terminals that do not send this new capability parameter shall not make use of the new methods described in this annex. The **TransportCapability** information can be sent as part of the Terminal Capability set exchange in the capability exchange phase. It is also included in the **openLogicalChannel**.

C.3.7 Elements of ATM signalling

C.3.7.1 ATM address

The ATM address for an RTP stream shall be given in the **mediaChannel** subfield of **H2250LogicalChannelParameters** of the H.245 **openLogicalChannelAck** message (or the **OpenLogicalChannel** in the case of Fast Connect). The **mediaChannel** subfield **UnicastAddress** or **MulticastAddress** shall be filled with the 20-octet NSAP-style ATM End System Address.

The use of E.164 for the address is handled by embedding it as the IDP part (AFI = 0x45) of an NSAP address. In this case, an international E.164 number is required.

C.3.7.2 Port Number

The **portNumber** field of the **openLogicalChannel** message is conveyed in the GIT information element as per [34]. The format of the GIT information element is specified in C.4.1.1. This enables the receiving side to associate the ATM VC with the proper RTP logical channel.

For backward compatibility with ITU-T Rec. H.323 Version 2 endpoints, ITU-T Rec. H.323 Version 3 (and later) endpoints shall also be able to use the B-HLI, according to ITU-T Rec. H.323 Version 2 Annex C, for conveying the **portNumber** field of **openLogicalChannel**. An H.323 Version 3 (or later) endpoint shall use the B-HLI only if it has prior knowledge that the terminating endpoint is H.323 Version 2. In cases where the H.323 version of the terminating endpoint is not known, such as establishing a call using Fast Connect, the endpoints shall first attempt to establish the ATM VC using the GIT information element for carrying the **portNumber**. If the connection fails the calling endpoint shall reattempt call setup using B-HLI instead of GIT. If the VC setup with B-HLI also fails, the terminal shall assume that ATM connectivity is not available and shall fall back to using RTP/UDP/IP for media channels. The format of the B-HLI information element is specified in C.4.1.2.

C.3.8 A/V streams on RTP on AAL 5

Servicing the **openLogicalChannel** primitive in H.245 triggers the connection establishment. The audio and video streams are then set up to the destination ATM address. The size of the Maximum Transmission Unit (MTU) shall be signalled in the AAL Parameters information element. The MTU choice may effect system efficiency because of AAL 5 packetization. The packetization rules for AAL 5 are contained in ITU-T Rec. I.363.5. If the non-AAL 5 default of 1536 octets is used, the MTU is packetized in 33 ATM cells and the last AAL 5 cell contains only padding and the AAL 5 number. The address field in the **mediaChannel** should be used to determine whether an ATM VC or a UDP port should be opened.

In the event that the ATM VC setup fails, the endpoint shall retry using RTP/RTCP and the higher layer transport protocol such as UDP.

RTP header compression can optionally be used, as described in section 2 of AF-SAA-0124.000 [33], in which case it must be negotiated using the **mediaTransportType**.

C.3.8.1 Unidirectional logical channels

H.323 has no concept of the reverse direction of a unidirectional logical channel. However, an important characteristic of point-to-point ATM VCs is that they are inherently bidirectional. The use of both directions of an ATM VC is therefore desirable. Otherwise, the audio and video streams will each need to be sent on two different VC's, one for each direction.

Endpoints conforming to this annex are encouraged to open their media streams as bidirectional logical channels. This reduces the number of AAL 5 VCs to two in typical situations, one VC each for audio and for video.

C.3.8.2 Bidirectional logical channels

If the bidirectional usage is indicated, the receiving endpoint shall send an **openLogicalChannelAck** (or the **openLogicalChannel** in the case of Fast Connect) and then it must watch for an ATM VC to be opened by the other endpoint. When an ATM VC is completed, it may then use the reverse direction for the media type indicated in the **openLogicalChannel** command. The endpoint that initiates the **openLogicalChannel** command is the endpoint that shall open the ATM VC.

If QOS is to be used, it shall be limited to the **H2250Capability** declared by the other endpoint. The chosen QOS is signalled as part of the establishment of an ATM VC.

If both endpoints have uncompleted **openLogicalChannel** commands for the same media session, these are resolved using the master/slave methods described in ITU-T Rec. H.245.

C.3.8.3 Maximum transmission unit size

The maximum MTU for AAL 5 is 65 535 octets. As part of **H2250Capability**, the MTU size can be specified in the capabilities exchange during H.245 setup. The forward and backward maximum MTU size shall be equal and will be taken from the smallest of the local and remote values specified in the capabilities exchange.

The MTU size is signalled as the AAL 5 maximum CPCS-PDU size for an ATM VC.

C.3.8.4 RTCP on IP over ATM

It is mandatory to open the logical channel for RTCP traffic on a UDP/IP port, using IP over ATM. RTCP is not permitted to ride directly on an AAL 5 VC.

C.3.9 QOS considerations (Optional)

C.3.9.1 QOS classes defined in ITU-T Rec. I.356

ITU-T Rec. I.356 defines four QOS classes, Class 1 (stringent class), Class 2 (tolerant class), Class 3 (bi-level class), and U class. Table C.1 summarizes the differences among the QOS classes.

Table C.1/H.323 – Provisional QOS class definitions and network performance objectives

	CTD	2-pt CDV	CLR (0+1)	CLR (0)	CER	CMR	SECBR
Default	None	None	None	None	4×10^{-6}	1/day	10^{-4}
Class 1 (stringent)	400 ms	3 ms	3×10^{-7}	None	Default	Default	Default
Class 2 (tolerant)	U	U	10^{-3}	None	Default	Default	Default
Class 3 (bi-level)	U	U	U	10^{-5}	Default	Default	Default
U class	U	U	U	U	U	U	U
CDV: Cell Delay Variation; CER: Cell Error Ratio; CLR: Cell Loss Ratio; CMR: Cell Misinsertion Rate; CTD: Cell Transfer Delay; SECBR: Severely Errored Cell Block Ratio; U: Unspecified/Unbounded.							

C.3.9.2 ATM transfer capability defined in ITU-T Recs I.371 and I.371.1

ATM Transfer Capability (ATC), defined in ITU-T Recs I.371 and I.371.1 as a set of ATM layer parameters and procedures, is intended to support an ATM layer service model and a range of associated QOS classes. Open-loop control ATCs (DBR and SBR) and closed-loop controlled ATCs (ABT and ABR) are specified in ITU-T Recs I.371 and I.371.1. SBR is subdivided into SBR1, SBR2 and SBR3, depending on how to handle CLP = 0/1 cells. ABT is subdivided into ABT/DT and ABT/IT depending on the use of negotiation regarding the block cell rate. Table C.2 summarizes the association of ATCs with QOS classes.

Table C.2/H.323 – Association of ATCs with QOS classes (from Table 3/I.356)

ATM Transfer Capabilities (ATC)	DBR, SBR1, ABT/DT, ABT/IT	DBR, SBR1, ABT/DT, ABT/IT	SBR2, SBR3, ABR	Any ATC
Applicable QOS class	Class 1 (stringent)	Class 2 (tolerant)	Class 3 (bi-level)	U class
ABR: Available Bit Rate; ABT/DT: ATM Block Transfer/Delayed Transmission; ABT/IT: ATM Block Transfer/Immediate Transmission; DBR: Deterministic Bit Rate; SBR1: Statistical Bit Rate configuration 1; SBR2: Statistical Bit Rate configuration 2; SBR3: Statistical Bit Rate configuration 3.				

C.3.9.3 Broadband transfer capability defined in ITU-T Rec. Q.2961.2

Broadband Transfer Capability (BTC) codes (DBR, BTC5, BTC9, BTC10 and SBR1) in Broadband bearer capability information element are defined in ITU-T Rec. Q.2961.2, and valid combinations of bearer class, broadband transfer capability and ATM traffic descriptor parameters are specified in Annex A/Q.2961.2. In the Setup message, the user can specify the BTC according to the traffic he/she generates and the intended use of network services. In Table A.1/Q.2961.2, 3 valid combinations are listed for bearer class BCOB-A, 8 combinations for BCOB-C and 13 combinations for BCOB-X or FR.

C.3.9.4 Opening of Virtual Channels

The endpoint that originated the accepted **openLogicalChannel** is responsible for opening the ATM VC. Support for QOS in the ATM VC is signalled at the time it is established. If successful, the ATM network provides a guaranteed QOS for the lifetime of the opened VC. QOS is specified in terms of Q.2931 Information Elements (IEs), including ATM Traffic Descriptor and Broadband Bearer Capability.

C.3.9.5 Use of DBR

The most likely available ATM traffic type is a constant bit rate using DBR. The use of DBR is signalled as part of the ATM broadband Bearer Capability IE (Bearer class = "BCOB-A"). Use of other ATM traffic type, such as SBR with end-to-end timing required [Bearer class = "BCOB-X" and BTC field = "SBR1 (0010011)"], is also possible.

C.3.9.6 Setting the proper cell rate

It is important to set the proper cell rate parameters in the ATM Traffic Descriptor information element. The peak cell rate can be derived from the H.245 capabilities exchange parameters and the RTP payload format packet size. For video, the **maxBitRate** field can be used from the **H261VideoCapability** or the **H263VideoCapability** to determine the ATM Cell rate. For audio, the audio capability chosen implies the bit rate to be used. For example, the use of **g711Ulaw64k** suggests the use of a 64 kbit/s audio channel, while the use of **g728** indicates the use of a 16 kbit/s channel. The RTP payload format indicates the packet size. For each packet, the subsequent AAL packet overhead and any needed padding to meet the AAL packetization rules must be added. This results in an overhead bit rate that is associated with the size of the packet and the way this packet is encapsulated in the AAL and the frequency of this overhead from this encapsulation.

The bit rate of the data to be sent and the packetization of the data according to the AAL packetization rules determine the cell rate. The packetization will determine the actual number of cells that must be sent for a given data stream at a given bit rate. The choice of MTU can affect the packetization as explained in C.3.8.

C.4 Protocol section

C.4.1 ATM signalling information elements

C.4.1.1 Generic information transport

IE Parameter	Value	Notes
Identifier related standard/application (octet 5)	00001011	ITU-T Rec. H.323
Identifier Type (octet 6)	00001011	H.245 **portNumber**
Identifier length (octet 6.1)	0000 0010	2 octets
Identifier value (octets 6.2-6.3)	H.245 **portNumber**	16-bit binary coded forward H.245 **portNumber**

H.323 Version 3 (or later) endpoints shall set the IE action indicator of the GIT information element to "clear call", according to 4.5.1/Q.2931. In this case, if the terminating endpoint does not support GIT information element coding it will reject the call with the cause value 100 for *Invalid information element content* according to 5.7.2/Q.2931. If the ATM VC setup attempt is rejected because the terminating endpoint does not understand GIT it will reject the VC call setup with cause number 99 *Information element non-existent or not implemented*, according to 5.7.2/Q.2931.

It should be noted that the **portNumber** field in H.245 is only 16 bits in length.

The H.245 **portNumber** is used by the receiving endpoint to associate the ATM VC with the proper RTP logical channel. The endpoint that initiates the **openLogicalChannel** command is the endpoint that opens the ATM VC. It is possible for the initiating endpoint to select an H.245 **portNumber** that is already in use by the receiving endpoint. This would cause a failure in the OLC procedure.

Additionally the receiving RTCP port is also specified by the initiating endpoint by implication. H.323 states that the corresponding RTCP data shall flow on a UDP port number equal to the H.245 **portNumber** plus 1. It is possible that the resulting port number for RTCP, H.245 **portNumber** plus 1, will be in use on the receiving endpoint since the H.245 **portNumber** is selected by the initiating endpoint.

Due to the above problems the receiving endpoint should have the choice of selecting the H.245 **portNumber**. If the **portNumber** is not specified in the **openLogicalChannel** the receiving endpoint shall specify a **portNumber** in the **openLogicalChannelAck** message (or **openLogicalChannel** in the case of Fast Connect). It is recommended that the transmitting endpoint does not specify the **portNumber** in the **openLogicalChannel** thereby requiring the receiving endpoint to specify one in the **openLogicalChannelAck** message (or **openLogicalChannel** in the case of Fast Connect).

The **portNumber** field of the **openLogicalChannel** message is used to select the H.245 **portNumber**. The receiving endpoint uses this H.245 **portNumber** to associate the ATM VC with the proper RTP logical channel. If the receiving endpoint finds that the given H.245 **portNumber** is inappropriate it can select a new H.245 **portNumber** and use the **portNumber** field of the **openLogicalChannelAck** message (or **openLogicalChannel** in the case of Fast Connect) to indicate the new value to the initiating endpoint. The selected H.245 **portNumber** field is conveyed in the GIT information element. This enables the receiving side to associate the ATM VC with the proper RTP logical channel.

The VC association port number is represented in network byte order in octets 6.2 and 6.3 of the GIT (i.e., octet 6.2 holds the MSB and octet 6.3 holds the LSB).

C.4.1.2 Broadband High Layer Information

IE parameter	Value	Notes
Length of B-HLI contents (octets 3-4)	3	
High layer information type (octet 5)	"0000 0001"	User-specific
High layer information (octets 5-7)	H.245 **portNumber**	16-bit binary coded forward H.245 **portNumber**

The B-HLI is only used for backward compatibility with H.323 Version 2 endpoints, as described in C.3.7.2.

C.4.1.3 ATM Adaptation Layer parameters

IE parameter	Value	Notes
AAL type (octet 5)	"0000 0101"	AAL 5
Forward maximum AAL 5 CPCS-SDU size (octets 6.1-6.2)	MTU size	The smaller **mTUsize** in the local and remote **QOSCapability.atmParms**
Backward maximum AAL 5 CPCS-SDU size (octets 7.1-7.2)	MTU size	Same as forward
SSCS type (octet 8.1)	"0000 0000"	Null SSCS

C.4.1.4 ATM Broadband bearer capability Information Element

a) In the case where the ATM traffic type in ITU-T Rec. H.245 is equal to "DBR":

IE parameter	Value	Notes
Bearer class	BCOB-A	
Susceptibility to clipping	Susceptible to clipping	
User-plane connection configuration	Point-to-point	

b) In the case where ATM traffic type in ITU-T Rec. H.245 is equal to "SBR1" with end-to-end timing required:

IE parameter	Value	Notes
Bearer class	BCOB-X	
Broadband bearer capability	"0010011" (SBR1)	SBR1 with end-to-end timing required
Susceptibility to clipping	Susceptible to clipping	
User-plane connection configuration	Point-to-point	

C.4.2 H.245 usage

The establishment of a H.323 call using AAL 5 media streams is done in a manner similar to the basic mode of H.323 on IP. The difference is that the completed **openLogicalChannel** exchange in H.245 should result in an AAL 5 VC being established. This is illustrated in Figures C.2 and C.3 for the unidirectional VC usage and the bidirectional VC usage, respectively.

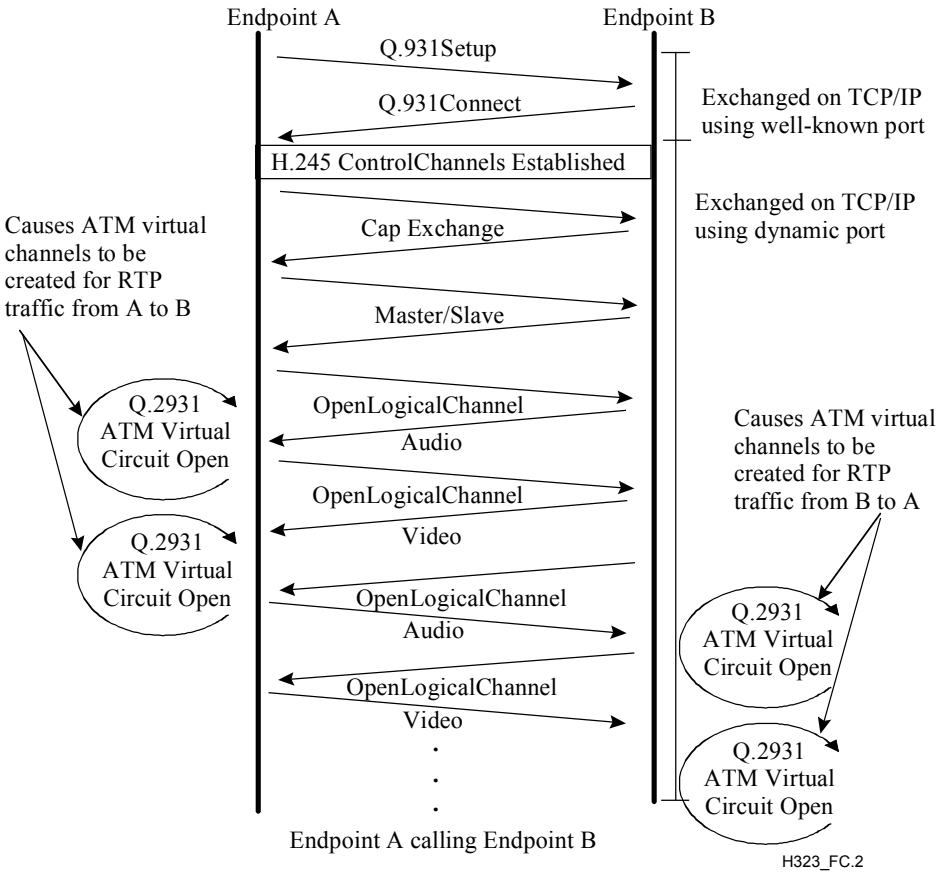


Figure C.2/H.323 – H.323 call establishment showing ATM effect – ATM VC's used unidirectionally

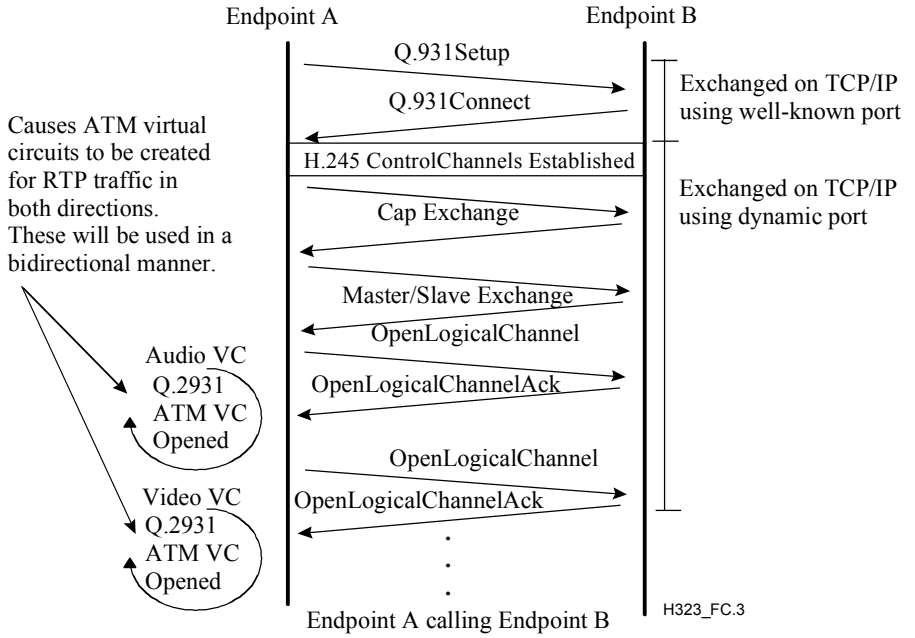


Figure C.3/H.323 – H.323 call establishment showing ATM effect – ATM VC's used bidirectionally

It should be noted that the ATM VC setups will occur in only one direction if bidirectional logical channels are used. In this case, the endpoint acknowledging the **openLogicalChannel** will merely bind the incoming ATM connection to an RTP session using the VC Association port number.

C.4.3 RTP usage

RTP and RTCP are defined in Annex A/H.225.0. RTCP is currently required for all H.323 connections and therefore is required even when using an AAL 5 VC. The RTCP is carried by UDP/IP, not directly by the AAL 5 VC.

C.4.4 Interoperation with H.323 on IP

Since the H.225.0 and H.245 communications are on IP, the endpoint will be able to receive calls from any other endpoint that is properly connected to the IP network. It is possible that H.323 endpoints will be used on ATM that do not support the methods described in this annex. They will strictly follow the basic method of using UDP/IP for the A/V streams. In this case, the endpoint will not declare the new **transportCapabilities** in H.245 and will refuse to open logical channels using ATM addressed VCs.

The protocol to **openLogicalChannel** using AAL 5 VCs for A/V streams should only be used if the received Capabilities have indicated that the method of this annex is supported. If this capability parameter is not present in the Terminal Capability Set, then the endpoint should only use **openLogicalChannel** using UDP/IP over ATM. This will ensure that the endpoint can communicate with other endpoints that support H.323, but may not support the methods in this annex.

Annex D

Real-time facsimile over H.323 systems

D.1 Introduction

Currently, facsimile and speech are typically sent using the PSTN with the same calling and addressing infrastructure. It is highly desirable to continue this approach in the context of this Recommendation. From a high level, facsimile can be viewed as another kind of real-time traffic similar to a particular speech coder. This seems appropriate, as facsimile entering the packet world via a gateway from the PSTN should logically be treated in a fashion similar to speech if the customer expects a real-time, assured end-to-end transmission service. The conversion of facsimile to email or other store-and-forward methods represents a new service that is beyond the scope of this Recommendation, which is a real-time protocol. It is recognized that manufacturers may wish to provide a gateway that falls back to a store-and-forward service when the real-time facsimile call fails. It is beyond the scope of this Recommendation when and how this decision is made, or by what means a store-and-forward facsimile service is implemented.

ITU-T Rec. T.38 [56] defines an Internet facsimile protocol consisting of messages and data exchanged between Facsimile Gateways connected via an IP network. This annex uses ITU-T Rec. T.38. Communication between the Gateways and G3/G4 Facsimile terminals is beyond the scope of ITU-T Rec. T.38. The reference model for T.38 is shown in Figure D.1 with three scenarios. In the first scenario, the two traditional Group 3 Facsimile Equipment (G3FE) terminals are virtually connected through the Gateways once the PSTN calls are established. All T.30 [55] session establishment and capabilities negotiation is carried out between the terminals. In the second scenario, the traditional Group 3 Facsimile (IAF) terminal is connected with an Internet Aware Fax terminal (IAF).

The IAF is directly connected to the IP network. In the third scenario, the two IAFs are directly connected to the IP network. In all the scenarios, T.38 packets are used on the IP network to communicate T.4/T.30 facsimile information. The transport of T.38 packets is either on TCP/IP or UDP/IP using the H.323 mechanism.

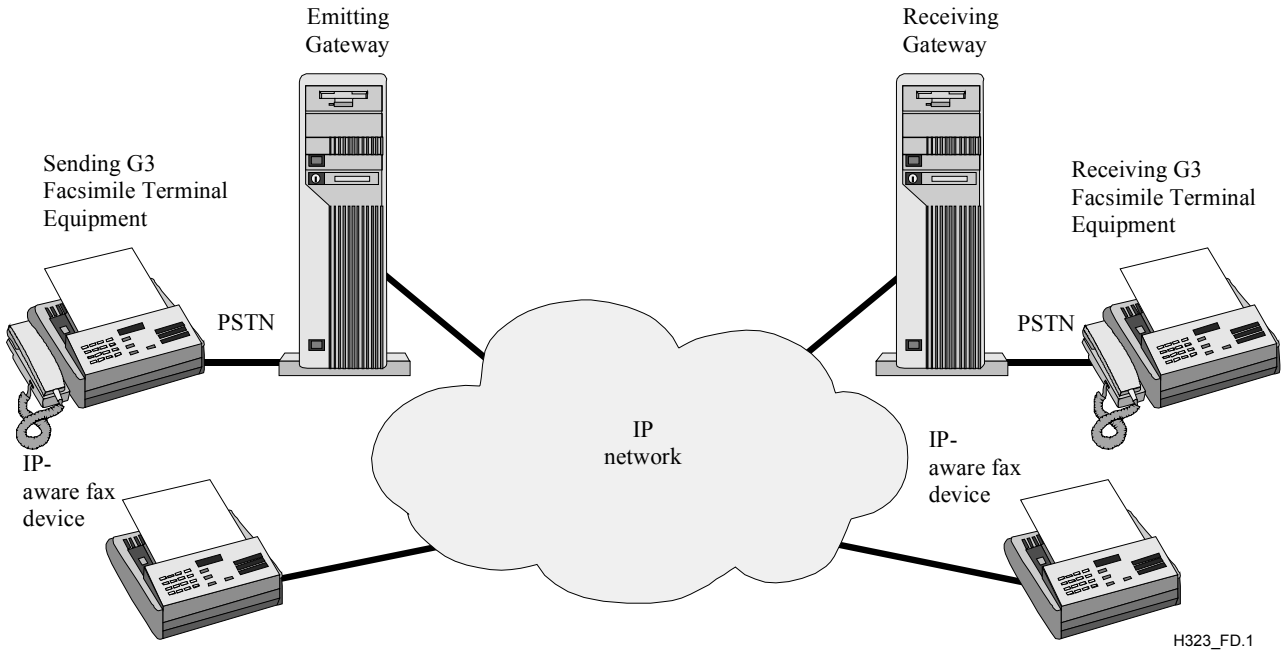


Figure D.1/H.323 – Model for facsimile transmission over IP networks

D.2 Scope

The scope of this annex is to use H.323 procedures to transfer T.38 packets in real time over the IP network. H.323 entities supporting facsimile capabilities shall use T.38 to support real-time facsimile services as described in this annex.

H.323 facsimile capable endpoints shall support the usage of TCP and UDP as described in ITU-T Rec. T.38. Annex B/T.38 describes a T.38-only capable terminal that supports a subset of H.245 messages using H.245 tunnelling. However, the T.38/Annex B terminal can interwork with an H.323/Annex D terminal using 8.1.7/H.323 "Fast Connect Procedure", and 8.2.1/H.323 "Encapsulation of H.245 Messages within H.225.0 call signalling Messages" procedures in this Recommendation. T.38/Annex B terminals interwork with H.323 terminals without being conformant to this Recommendation. An H.323 terminal that supports the procedures of this annex shall interwork with T.38/Annex B terminals.

D.3 Procedures for opening channels to send T.38 packets

Fast Connect is used to describe the H.323 procedures for opening channels for the transportation of T.38 packets. The traditional sequence can also be used, though it is not described here.

D.3.1 Opening the voice channel

Zero, one (sender to receiver channel or receiver to sender channel), or two (sender to receiver channel and receiver to sender channel) logical channels for voice may be opened depending on the capability of the sender and the receiver. If a voice channel is desired, the voice channel shall be opened as specified by the procedures in 8.1.7/H.323 "Fast Connect". Support of voice by facsimile applications is not mandatory in this annex.

D.3.2 Opening the facsimile channels

Two unidirectional reliable or unreliable logical channels (sender to receiver channel and receiver to sender channel) as shown in Figure D.2 or, optionally, one bidirectional reliable channel as shown in Figure D.3 shall be opened for the transfer of T.38 packets. T.38 packets can be transferred using either TCP or UDP. In general, the usage of TCP is more effective when the bandwidth for facsimile communication is limited. On the other hand, the usage of UDP may be more effective when the bandwidth for facsimile communication is sufficient.

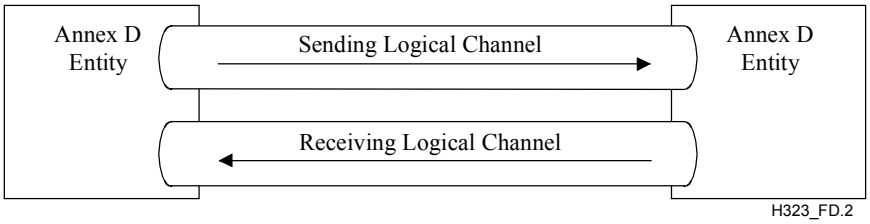


Figure D.2/H.323 – A pair of unidirectional channels

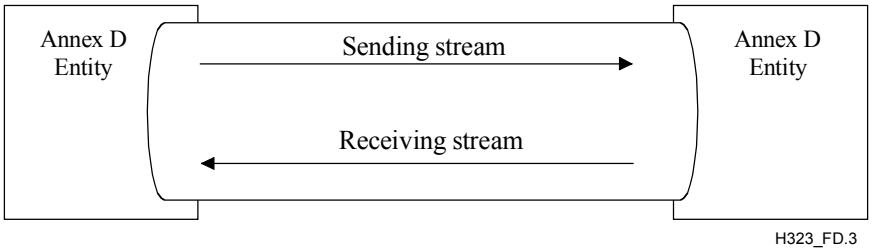


Figure D.3/H.323 – A unit of bidirectional channels

NOTE – In the first version of this annex, it was not possible to use a single bidirectional reliable channel. In order to retain backward compatibility, the endpoint may specify support for bidirectional reliable channels by including the **t38FaxTcpOptions** SEQUENCE and setting the **t38TCPBidirectionalMode** field to TRUE. If the other endpoint does not include the **t38FaxTcpOptions** SEQUENCE, the endpoint shall assume that a single bidirectional reliable channel for T.38 is not supported and shall use either two unidirectional reliable or unreliable channels.

The sender terminal specifies a TCP/UDP port in the **OpenLogicalChannel** in the **fastStart** element of *Setup*. The receiver terminal shall provide its TCP (or UDP) port in the **OpenLogicalChannel** of the **fastStart** element as specified by the procedures in 8.1.7/H.323 "Fast Connect procedure".

The receiver shall open the TCP/UDP port based on the preference of the sender. If the sender terminal has a preference for UDP or TCP, then it shall indicate its preference by ordering proposals in the **fastStart** sequence according to 8.1.7.1/H.323. The receiving terminal can select the transport, TCP or UDP, by returning the desired proposals in **OpenLogicalChannel** structures in the **fastStart** element of *Connect*.

Figures D.4 and D.5 show the signalling used to open unidirectional and bidirectional channels using Fast Connect.

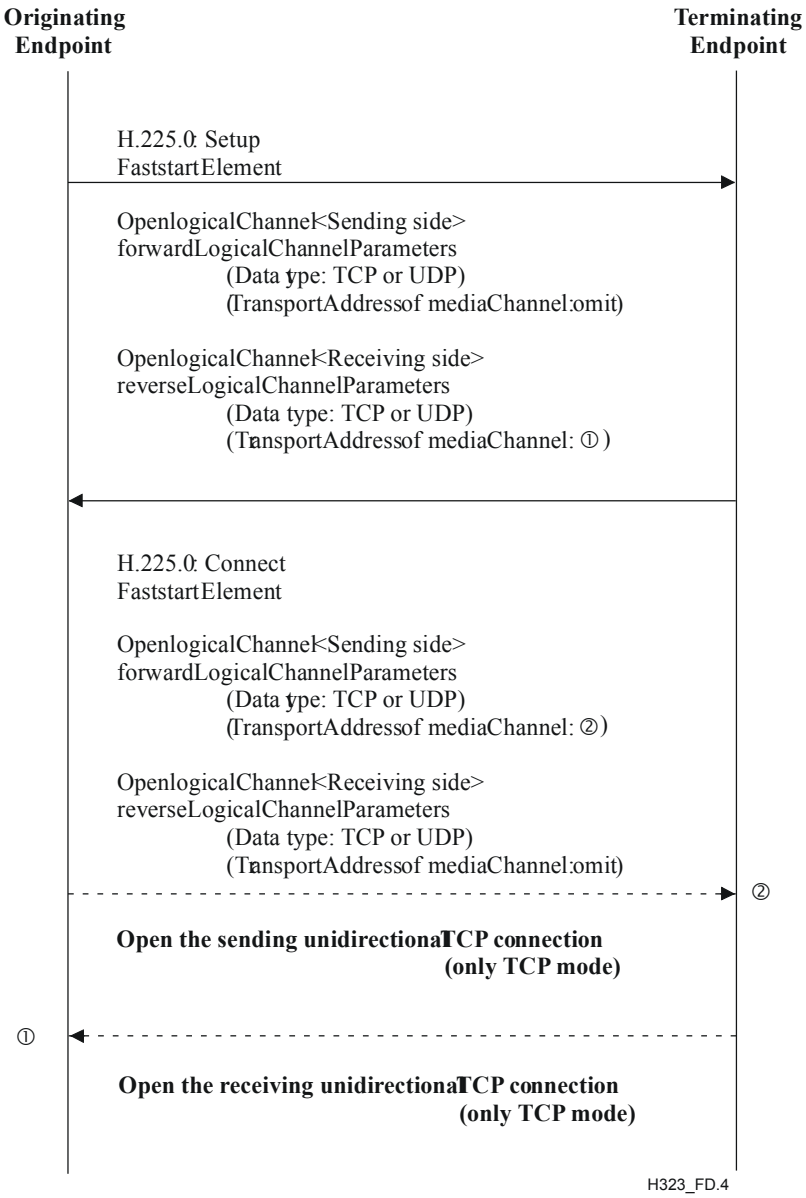


Figure D.4/H.323 – Two unidirectional channels with fast connect

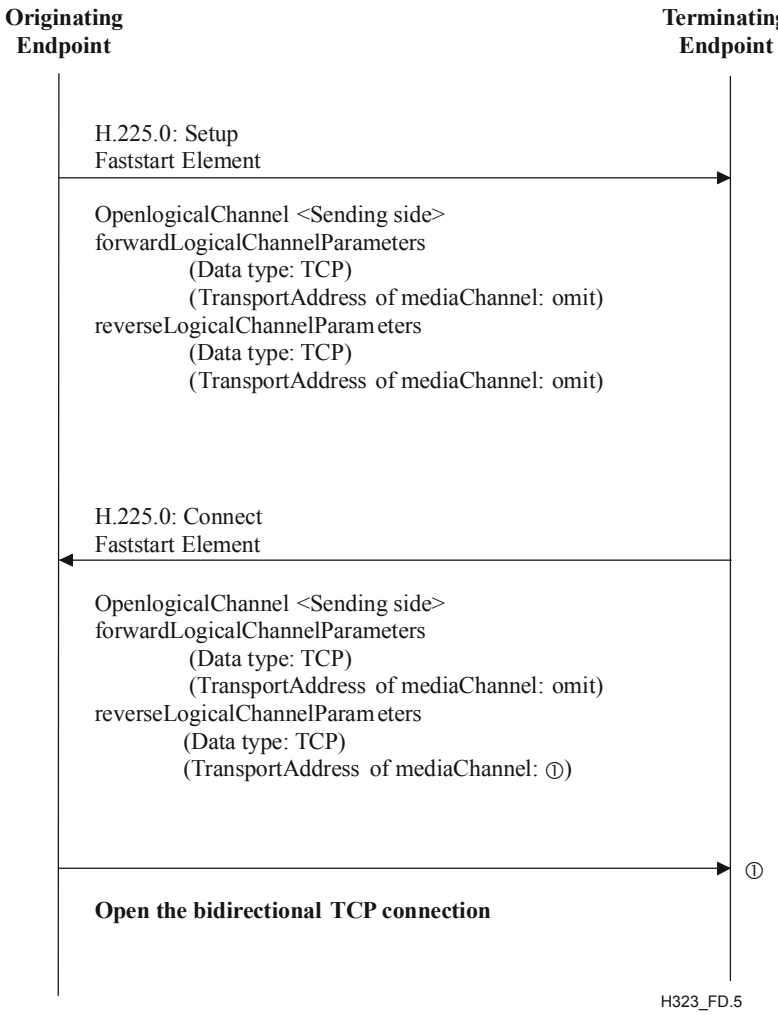


Figure D.5/H.323 – One bidirectional reliable channel with fast connect

D.3.3 DTMF transmission

DTMF tones shall be sent by H.323/Annex D terminals using **UserInputIndication** to interwork with T.38/Annex B terminals. H.323/Annex D terminals may send DTMF tones in-band with the voice when T.38/Annex B terminals are not involved in the call.

D.4 Non-Fast Connect procedures

It is noted that in Non-Fast Connect, the normal H.245-based **OpenLogicalChannel** procedures can be used to open and close both UDP and TCP fax channels (refer to 6.2.8.2/H.323). Tunnelled H.245 can also be used to open and close channels. It is also noted that non-Fast Connect and non-tunnelled H.245 procedures do not apply to interworking with ITU-T Rec. T.38.

Figures D.6 and D.7 show the signalling used to open unidirectional and bidirectional channels when not using Fast Connect.

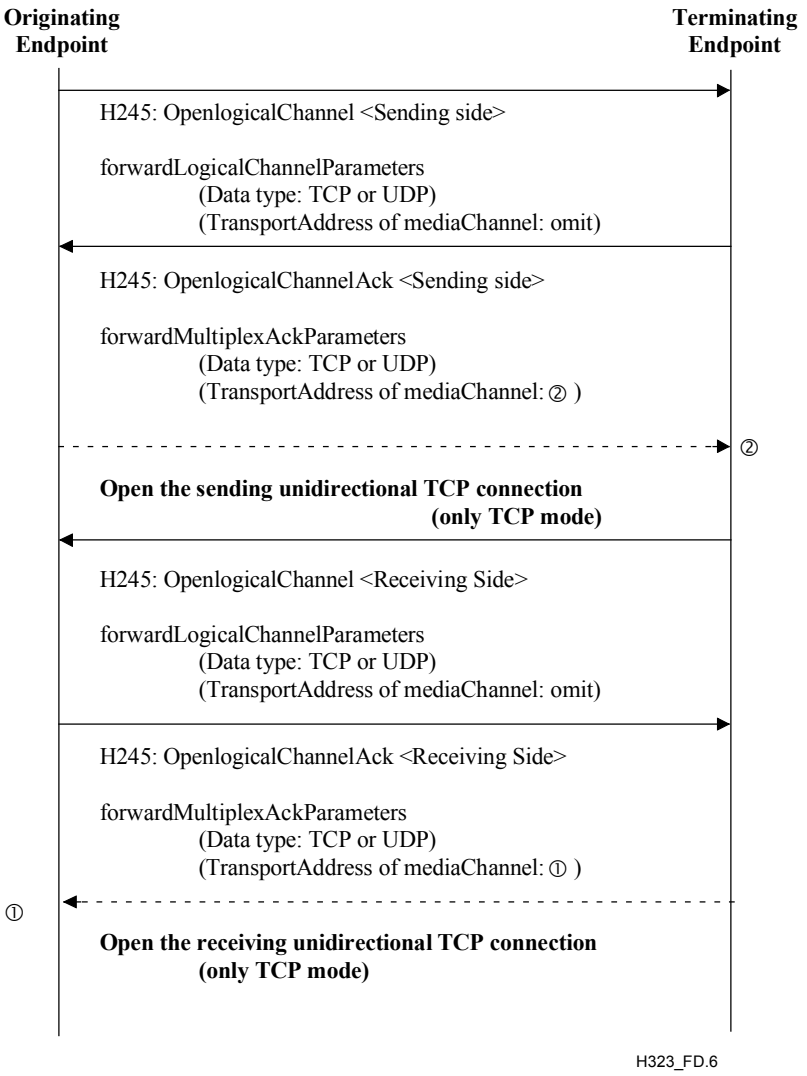


Figure D.6/H.323 – Two unidirectional channels without fast connect

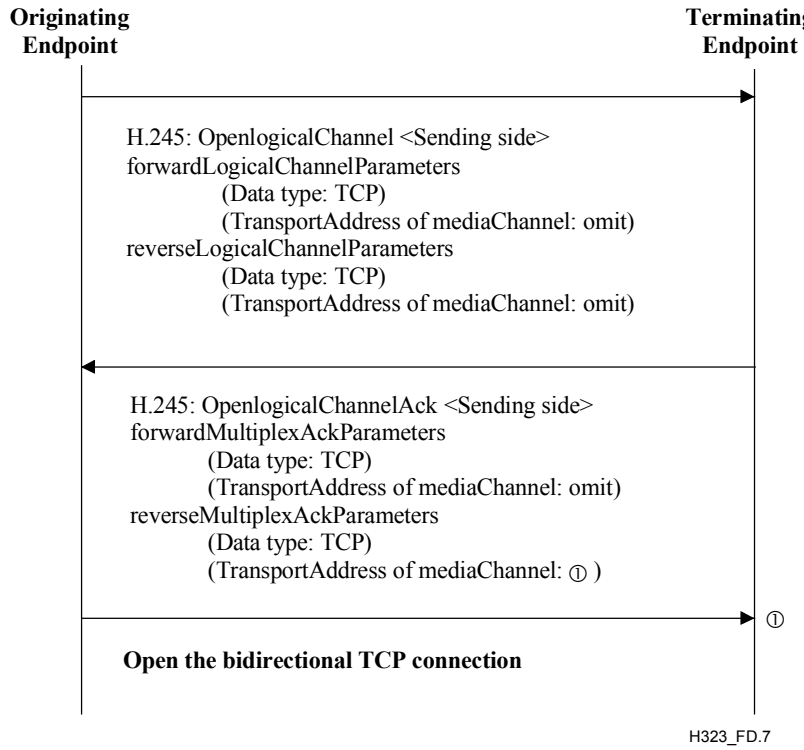


Figure D.7/H.323 – One bidirectional channel without fast connect

D.5 Replacing an existing audio stream with a T.38 fax stream

An endpoint that wishes to replace an existing audio stream with a fax stream shall use the following mechanism to achieve this goal.

Once the audio call has been established – ideally via the use of Fast Connect and prior to the receipt of the CONNECT message – the endpoint that wishes to replace the audio stream with T.38 fax shall initiate H.245 procedures via tunnelling if H.245 has not already been started.

During H.245 capability exchange, each endpoint shall express its capability of receiving and transmitting T.38 fax by including the **t38fax** field of the **DataApplicationCapability** structure. The presence of this field indicates that the remote endpoint is capable of supporting the T.38 Fax Mode.

It should be noted that the Connect message may arrive while H.245 procedures are taking place. After H.245 procedures have completed and the Connect has been received, either endpoint may detect fax tones (i.e., CNG or CED) or the presence of V.21 carrier and HDLC flags. Typical scenarios for facsimile call detection rely on the analysis of CNG calling tone and a response of the CED answer tone and/or the initiation of fax procedures using the V.21 carrier and HDLC flags. Note that in some implementations the presence of either CNG or CED are optional. Therefore, both endpoints should take an active role in order to properly detect fax.

When using two unidirectional fax channels, the endpoint that detected the tone shall initiate the standard H.245 Mode Request procedure by sending a **requestMode** message to its remote counterpart with the **t38fax** data mode as the requested mode. The endpoint that receives the **RequestMode** message shall return a **requestModeAck** message. On receiving the **requestModeAck** message, the initiating endpoint shall close its audio logical channel and open a T.38 logical channel. Similarly, the remote end shall close its audio logical channel and open a T.38 fax logical channel. After acknowledgments have been received for each of the T.38 open logical channels, fax transmission and reception takes place.

Figure D.8 illustrates a successful switchover from voice to fax when a separate H.245 channel is already open for two unidirectional media channels.

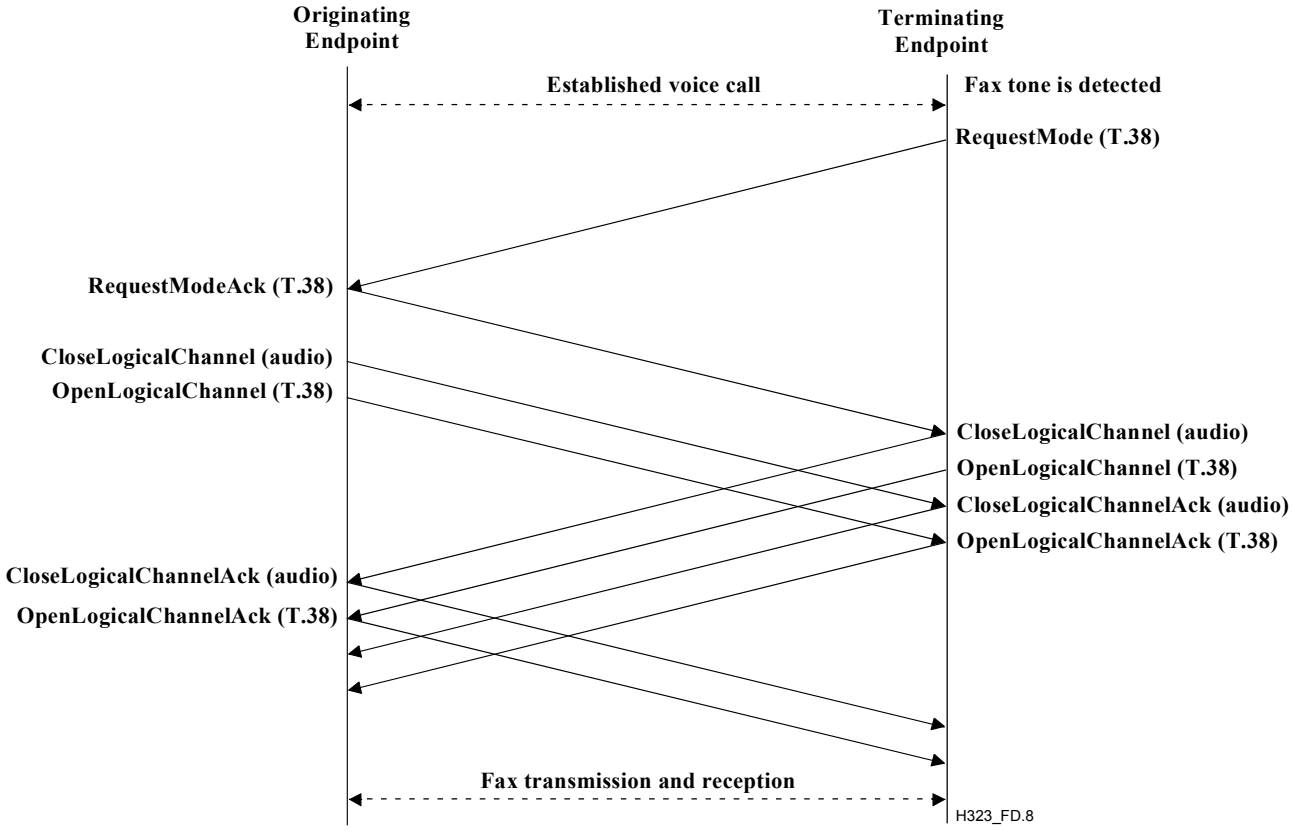


Figure D.8/H.323 – Successful switching of an existing voice call to T.38 using two unidirectional media channels without tunnelling

Figure D.9 illustrates a successful switchover from voice to fax using H.245 tunnelling for two unidirectional media channels.

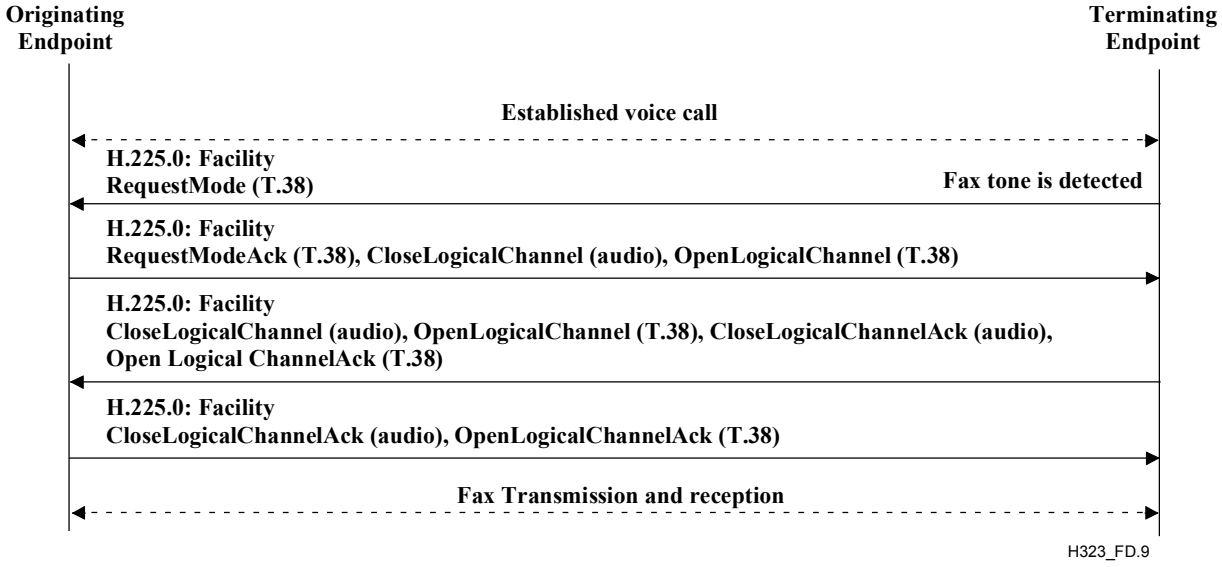


Figure D.9/H.323 – Successful switching of an existing voice call to T.38 using two unidirectional media channels with tunnelling

When using a bidirectional fax channel (for TCP only), the request mode command is not necessary: the endpoint that detected the tone shall close its open channels, request the reverse channels to be closed by the other endpoint, and open a bidirectional T.38 channel. Upon reception of the request channel close command, the remote end shall close its audio channel. After acknowledgements have been received for each of the T.38 open logical channels, fax transmission and reception takes place.

Figure D.10 illustrates a successful switchover from voice to fax when a separate H.245 channel is already open for one bidirectional media channel.

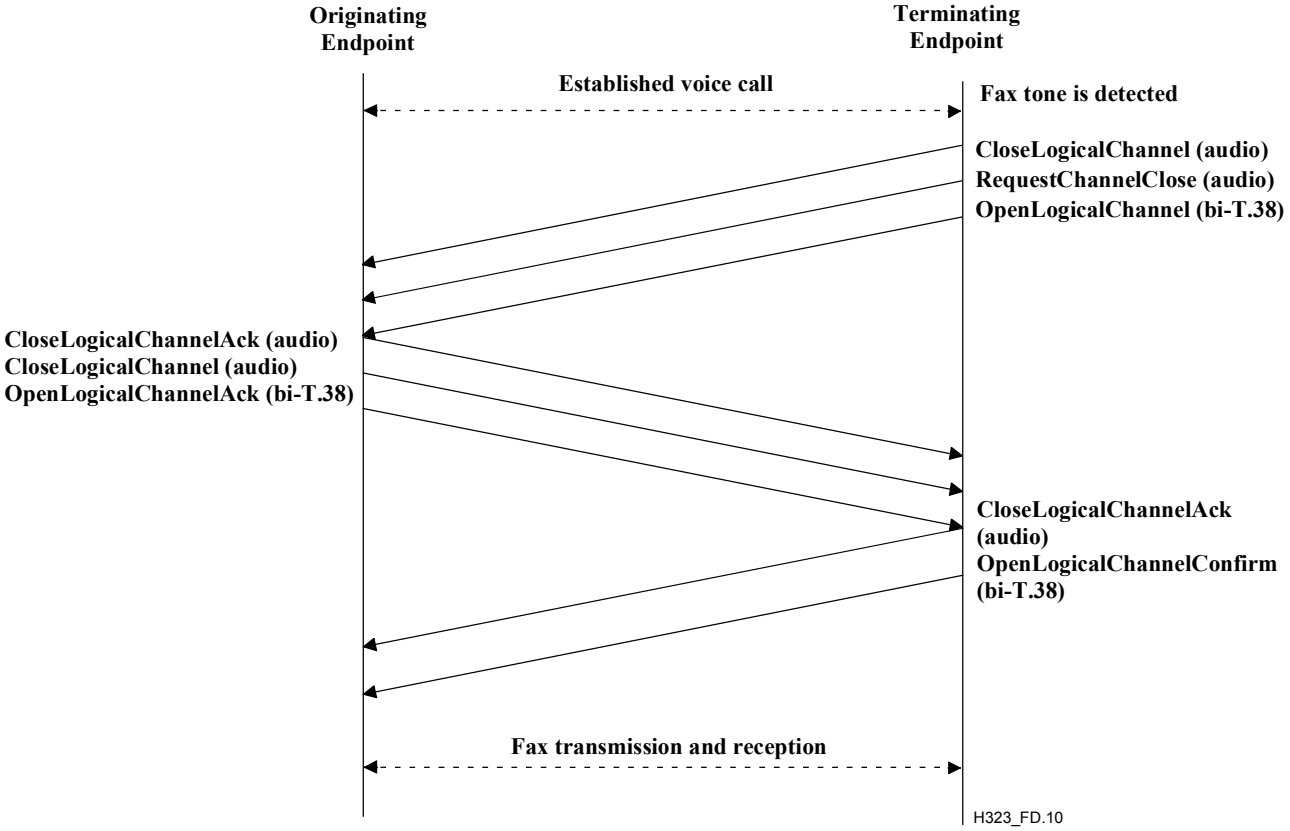


Figure D.10/H.323 – Successful switching of an existing voice call to T.38 using one bidirectional media channel (TCP) without tunnelling

Figure D.11 illustrates a successful switchover from voice to fax using H.245 tunnelling for one bidirectional media channel.

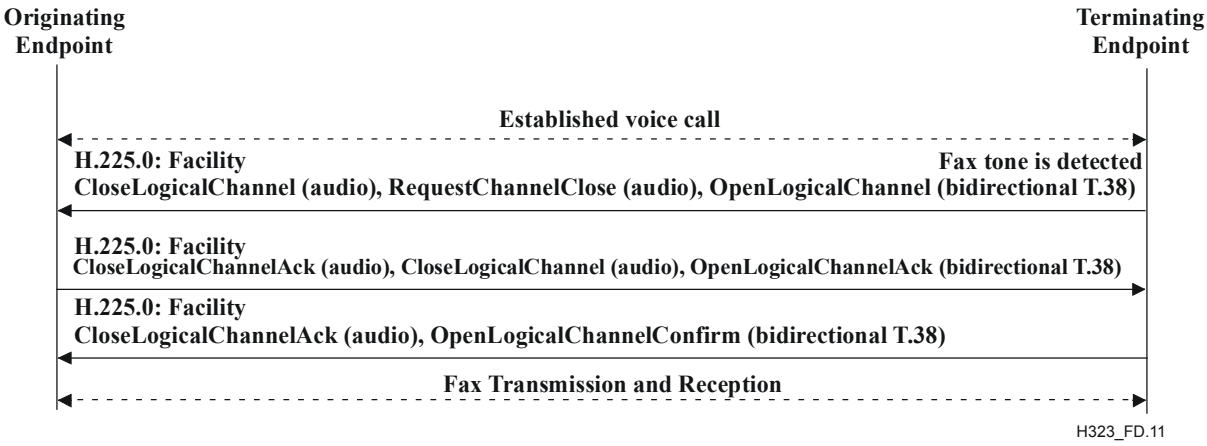


Figure D.11/H.323 – Successful switching of an existing voice call to T.38 using one bidirectional media channel (TCP) with tunnelling

Should either endpoint wish to return to an audio call after the fax transmission has ended, the Mode Request procedure shall be initiated using an audio codec as a parameter. The above procedure also applies to traditional "slow start" cases, in the event that Fast Connect cannot be established between the two endpoints.

D.6 Usage of the MaxBitRate in messages

When TCP is used for T.38 fax transmission, **maxBitRate** in the ARQ/BRQ does not include the fax data rate, and if a voice link is switched off when the fax session starts, a BRQ shall be used to indicate to the Gatekeeper that the bandwidth has changed. When UDP is used for T.38 fax transmission, **maxBitRate** in the ARQ/BRQ does include the bit rate needed for the fax session. The endpoint (terminal, gateway) shall send BRQs to the Gatekeeper as bandwidth needs change during the call. It is noted that the **maxBitRate** in the **OpenLogicalChannel** element in the Setup during Fast Connect is different from the **maxBitRate** in ARQ/BRQ, and does refer to the peak bit rate that the fax call will use.

D.7 Interactions with gateways and T.38/Annex B devices

The following case must be considered:

H.323/Annex D device (with voice) <–> T.38/Annex B device (without voice).

Note that these devices may be terminals or gateways; it does not affect the discussion. A fax call arrives from the "voiceless" side, but the voice side must generate an outgoing voice call that is not connected to anything although tones or announcements might be played. In the opposite direction, the H.323/Annex D device cannot offer a voice call to a "voiceless" device, as it cannot receive voice.

The H.323/Annex D gateway may send both a voice and fax **OpenLogicalChannel** element in the Setup message. If it encounters a T.38 device, only the fax channel will be opened if both were proposed. If the call mistakenly encounters a non-fax H.323 device, the fax port will not be opened. This is the equivalent of a fax machine calling a telephone.

An H.323/Annex D device becomes aware that it is talking to a T.38/Annex B device due to the following sequence of events:

1) The T.38/Annex B device does not supply an H.245 port in the Connect or Setup.

2) The H.323/Annex D device uses the Facility message as described in 8.2.3/H.323 and transmits a **FACILITY** message with a **FacilityReason** of **startH245** and provides its H.245 address in the **h245Address** element. The T.38/Annex B endpoint receiving a **FACILITY** message with a **FacilityReason** of **startH245** will respond with a **FACILITY** message having a **FacilityReason** of **noH245**. At this point the H.323/Annex D device should cease all attempts to open the H.245 channel.

Annex E

Framework and wire-protocol for multiplexed call signalling transport

E.1 Scope

This annex describes a packetization format and a set of procedures (some of which are optional) that can be used to implement UDP and TCP based protocols. The first part of this annex describes the signalling framework and wire-protocol, and subsequent clauses detail specific use cases. The only profile currently specified in this revision is for transporting H.225.0 call signalling messages.

This annex is designed to operate in engineered networks and use the security services provided by H.323 (e.g., H.235, IPSec). This annex should not be used over the public Internet, due to security and traffic considerations.

E.1.1 Introduction

E.1.1.1 Multiplexed transport

This annex provides a multiplexed transport layer that can be used to transmit multiple protocols (with optional reliability) in the same PDU. Often-used protocols have specific code points (also called "payload types"). Other protocols can be carried and identified using the ObjectID-typed payloads.

E.1.1.2 Multiple payloads in a single PDU

Annex E PDUs can contain multiple "payloads", each a different protocol and targeted at a different session (the definition of a "session" is protocol dependent). Note that there is no implicit relation between payloads when they arrive in the same PDU.

E.1.1.3 Flexible header options

Annex E PDU and Payload headers are configurable. Minimum header size can be as small as 8 octets, and may grow up to 20 octets when all optional fields are present.

E.1.1.4 Ack message

Messages carried using UDP can get lost. If the application needs assurance that a sent message arrived successfully, it may request an Ack message for the PDU.

A sender shall specify in the <ackRequested> field whether it wants to receive an Ack message for a PDU being sent, and the receiver shall reply with an Ack Payload if the <ackRequested> field is set.

NOTE – Ack messages shall be sent by the Annex E transport layer, not by the application using the Annex E stack. The specific Ack behaviour is mandated by the signalling model the Annex E stack is instructed to use by the application.

E.1.1.5 Nack message

A Nack message shall be used to signify some error condition. Such errors may be the inability to support a specific payload type, the arrival of a malformed PDU, and others. These messages may or may not have the effect of dropping an ongoing call.

NOTE – Nack messages are be sent by the Annex E transport layer, not by the application using the Annex E stack.

E.1.1.6 Sender sequence number policy

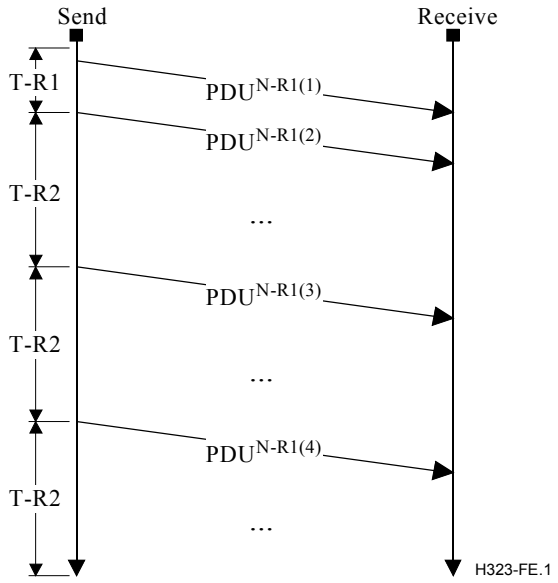
Assigned per host-address and source-port, sending Annex E layers shall start with some random value, incrementing by 1 for every PDU sent. If the sequence number reaches 2^{24} (16 777 216) it shall wrap around to 0.

E.1.1.7 Receiver sequence number policy

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

E.1.1.8 Retransmissions

When messages get lost (and an Ack was requested and not received) the sender may retransmit the message. The retransmission policy attempts to combat first-message lost by retransmitting quickly, but if that message is lost too, the sender is required to backoff the retransmission delay by a factor of more than two. See Figure E.1.



Retransmission timers and counters:

Item	Value	Comments
T-R1	500 ms	A reasonably small value is chosen here to compensate for possible 1st packet loss
T-R2	(T-R1 | T-R2) × N-R2	If the first retransmitted packet is lost, apply some back-off. If a previous T-R2 value is available, use it instead of the initial value (T-R1).
N-R1	8	Maximum number of retransmissions before abandoning the connection
N-R2	2.1	Multiplier to be used for back-off

Figure E.1/H.323 – PDU retransmission

When there is a known round-trip message interval value from a previous transmission, timer T-R1 should be set to that round-trip message interval value +10%.

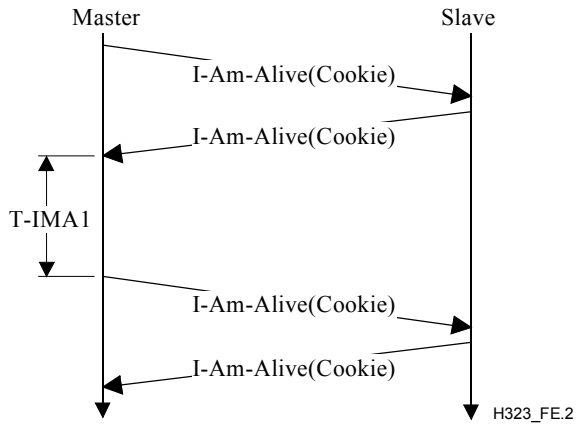
E.1.1.9 Connection keep-alive

When running over TCP, the presence of a persistent TCP connection can ensure that one side is aware of the remote side failures (by observing TCP failures). When running over UDP, there is no such "state" associated, and another procedure must be used.

The solution is for one side of the call (usually the "server" or "master" side if such classification is relevant) to send an "I-Am-Alive" message to the other side, to let the remote application know the host is still up. The remote side will answer with an I-Am-Alive message of its own as proof that it too is up. A cookie may be provided by the originator of an I-Am-Alive sequence, and if made available, shall be returned in the answer I-Am-Alive.

The retransmission timer of the I-Am-Alive messages may be reset on receiving other relevant message, as it is proof the remote end is alive. This saves bandwidth, as I-Am-Alive messages will be sent only when really needed. This capability is decided on a per-protocol basis.

Generating I-Am-Alive messages is optional, however, all entities shall support the ability to reply to I-Am-Alive messages (e.g., the ability and requirement to answer an I-Am-Alive message is not optional, and when such a message is received, it shall be answered according to the procedures defined in this annex). See Figure E.2.



I-Am-Alive timers

Item	Value	Comments
T-IMA1	6 seconds	I-Am-Alive transmission interval[a)]
N-IMA1	6	Number of consecutive I-AM-ALIVE messages not responded to after which the remote peer is declared dead
[a)] These timers should follow the recommended values in Annex R if Annex R is also being used between the two entities.		

Figure E.2/H.323 – I-Am-Alive transmission

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 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.

E.1.1.11 Reply hints

It is advisable for Annex E implementers to add a slight delay before an Ack message is sent back, to allow the application to attach a protocol payload to accompany the Ack payload. A Header option is available to allow senders to Hint to the remote transport layer that a reply is expected for a given message.

NOTE – For example, when a H.225.0 SETUP message is sent, the stack can delay the reply of the Ack payload slightly when the ReplyHint bit is set to ensure the application will have time to provide the return CONNECT payload (for example). The returning PDU will then contain both an Ack (for the SETUP) and the CONNECT payload.

E.1.1.12 Well-known port and port spawning

This annex supports one main well-known port UDP/TCP port 2517. Applications supporting Annex E operations when receiving a payload that the main well-known port does not support (identified either using the static payload type or the object-ID payload type) may reply with a Nack message that instructs the sender to send this specific payload type to a different port and IP address.

E.1.2 Signalling models

Signalling may follow many models. Each protocol implementation using this annex shall support one of the models (as described below) or choose a different signalling model that suits its requirements.

E.1.2.1 Real-time model

In the real-time model, if a PDU is lost, there is no use to re-send the PDU as the information may already be irrelevant. An example of such a protocol is RTP when used for real-time audio or video streaming. For such protocols the delay caused by retransmission is worse than losing the information.

When using this model, the Ack-flag shall always be cleared.

E.1.2.2 Serial model

In the serial-model, when a PDU is sent, the Annex E 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**).

E.1.2.3 Mixed model

The mixed model may imply that the protocol state machine and the Annex E state-machine are intertwined. Such implementations may use the Ack-bit where appropriate.

When using this model, use of the Ack-flag can be forbidden, optional, or mandatory, as prescribed by the protocol.

E.1.2.4 Annex E over TCP

This annex may be used over TCP. When used over TCP, the Ack message shall not be used. In addition, the L-bit in the PDU header shall be set, triggering the availability of the payload-count or PDU-length fields.

E.1.3 Optional payload fields

E.1.3.1 Session identifier

Annex E payloads support an optional session field that may be used to identify a session within the multiplexed transport that the payload belongs to. The session field is 16 bits long.

NOTE – This field may be used for example to carry the CRV (e.g., Call Reference Value as defined in ITU-T Rec. Q.931) in H.225.0 messages. The interpretation of the session field is protocol specific.

E.1.3.2 Source/Destination address identifier

Annex E payloads support an optional Source/Destination field that may be used to identify the source, the destination (or both) of the payload. The Source/Destination field is 32 bits long.

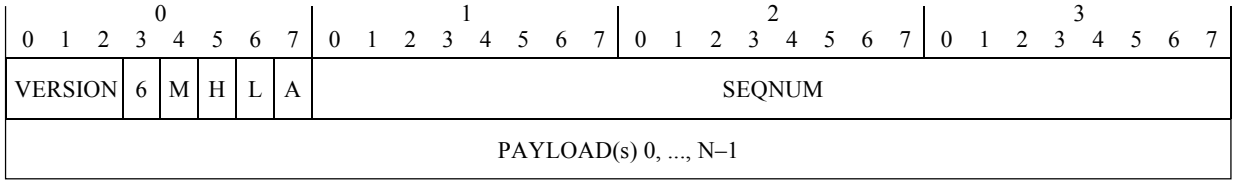
NOTE – This field may be used for example in ITU-T Rec. H.283 to express the [<M><T>] address identifying the source node of the packet, and the [<M><T>] address identifying the destination node of the packet. The interpretation of the source/destination field is protocol specific.

E.1.4 Wire-protocol

Annex E transport uses binary encoding as defined in the rest of this subclause. Structures and multi-byte fields shall use network-byte-ordering (e.g., big-endian).

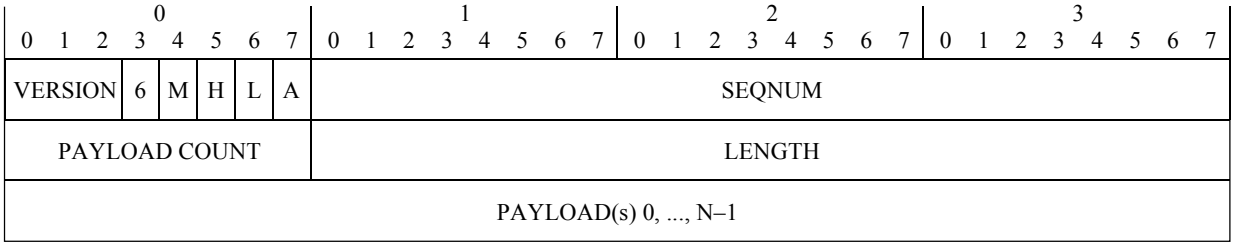
E.1.4.1 Header structure

The following structure shall be used to encode the Annex E Header. If the L-bit is cleared (hence there is no payload-count or PDU length indication), the length of the payloads within the message, and their number can be inferred from the message size as reported by the transport layer. See Figures E.3 and E.4.



Field	Content of fields	Bits
VERSION	Unsigned Integer; senders shall set this field to zero. Version number 7 is reserved for experimental use and shall be ignored by commercial implementations	3
6	When cleared, it means all IP addresses are IPv4 compliant (using 32 bits). When set, means all IP addresses are IPv6 compliant (using 128 bits)	1
M	Multicast bit. If set, the PDU was sent using Multicast, if cleared, the PDU was unicast. Senders shall set this bit if the PDU was multicast, otherwise they shall clear the bit	1
H	Reply-Hint bit – when set, this message will result in a reply, e.g., when set, the Ack message should be delayed to give the application a chance to provide an answer payload with the Ack payload	1
L	Length indicator. If present, an additional 4 OCTETs are present that contain the number of Payloads in the PDU (8 bits) and the total length (in OCTETs) of the PDU (24 bits)	1
A	Boolean: TRUE indicates that an Ack is requested for this PDU	1
SEQNUM	Unsigned Integer between 0 and 16 777 215: the sequence number of this PDU	24
PAYLOAD(s)	Sequence of payload structures	$8 \times n$

Figure E.3/H.323 – Header structure when the L-bit is cleared



Field	Content of L-bit supplementary fields	Bits
PAYLOAD COUNT	Total number of payloads in PDU –1 (e.g., 0 means there is one payload, 1 means there are two, etc.)	8
LENGTH	Total length in OCTETs of all payloads (excluding header)	24

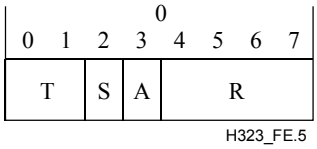
Figure E.4/H.323 – Header structure when the L-bit is set

E.1.4.2 Payload structure

The following structures shall be used to encode Annex E payloads.

E.1.4.2.1 Payload header flags

Every payload begins with a flags OCTET, that describes what optional fields are in the payload header. See Figure E.5.



Field	Content of fields	Bits
T	Two bits defining the payload identification type: **00**: Annex E Transport Messages **10**: Static-Payload typed messages **01**: OBJECT IDENTIFIER typed messages **11**: Reserved for Future Use	2
S	Signifies the presence of a Session field	1
A	Signifies the presence of a Source/Destination Address field	1
R	Reserved for future use, shall be cleared by senders	4

Figure E.5/H.323 – Payload flags

E.1.4.2.2 Annex E transport messages

Both T bits in the Payload header flags OCTET shall be set to 0 (zero) for all Annex E Transport Messages. The next octet shall signify what Annex E transport message is following. Both S and A bits shall be cleared. See Figure E.6.

Value	Interpretation
0	I-Am-Alive message
1	Ack message
2	Nack message
3	Restart Message
4..255	Reserved for future use

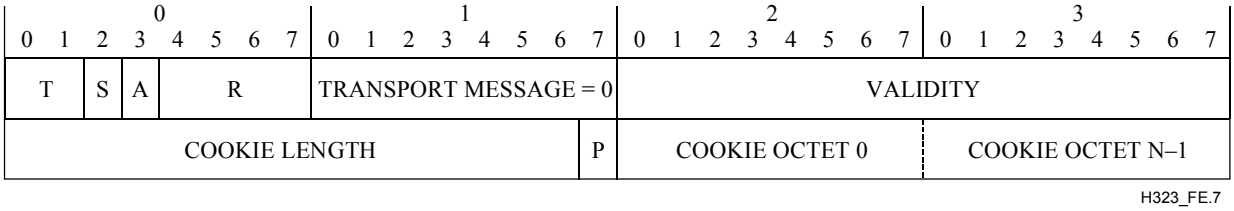
Figure E.6/H.323 – Annex E transport messages

E.1.4.2.2.1 I-Am-Alive message

The following structure shall be used to encode Annex E I-Am-Alive payloads. The transport-message octet shall be set to 0 (zero). The validity period is expressed in 100s of milliseconds.

- If the replyRequested bit (**P**) is set, the receiver shall reply with an I-Am-Alive message with the cookie (if provided).
- ReplyRequested is not the same as ackRequested in the PDU header, which results in an Ack message. replyRequested results in an I-Am-Alive message.
- If a validity period is set to ZERO (0), timer **T-IMA1** shall be used.
- PDUs that contain only an I-Am-Alive Payload shall clear the Ack-bit in the PDU header.

See Figure E.7.

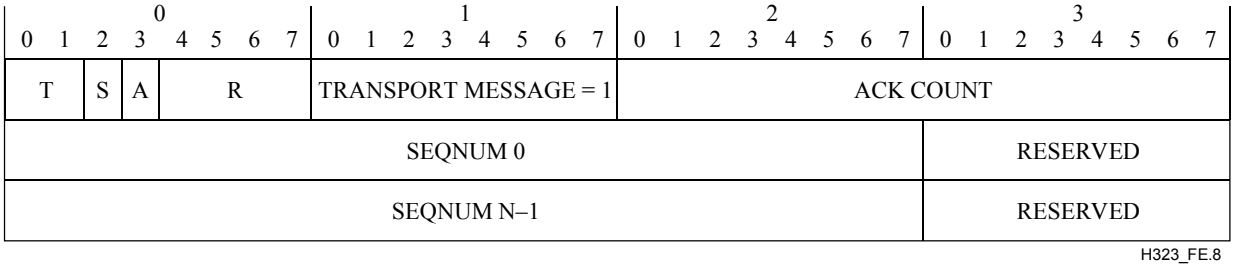


Field	Content of fields	Bits
VALIDITY	Unsigned Integer: The time in 100s of milliseconds that this I-Am-Alive is valid for	16
COOKIE LENGTH	The length (in BYTEs or OCTETs) of the COOKIE field	15
P	Reply Requested	1
COOKIE	BYTEs or OCTETs of the cookie	8 × n

Figure E.7/H.323 – I-Am-Alive message

E.1.4.2.2.2 Ack message

The following structure shall be used to encode Ack messages. The transport-message octet shall be set to 1 (one). PDUs that contain only an Ack Payload shall clear the Ack-bit in the PDU header. See Figure E.8.

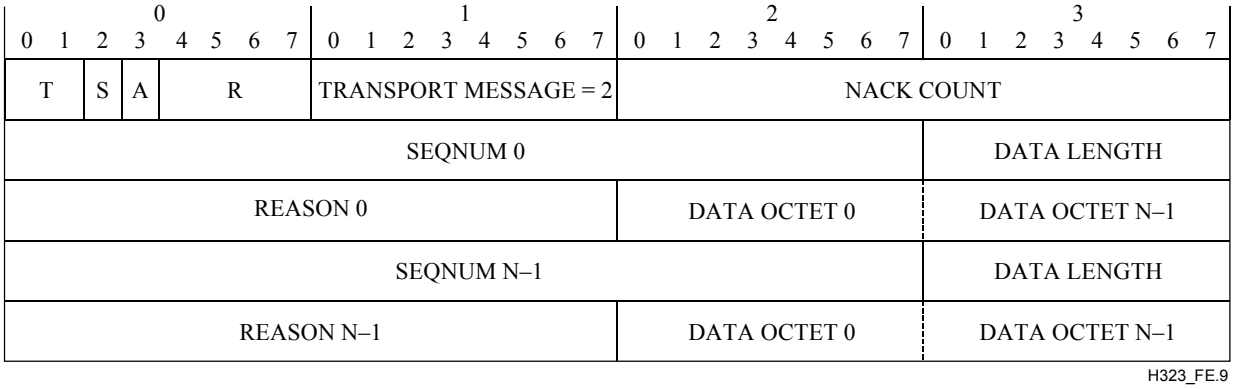


Field	Content of fields	Bits
ACK COUNT	The number of SEQNUM fields that follow	16
SEQNUM 0, ..., N–1	The Sequence Number(s) of the PDUs that are being ACKed for	24 × n
RESERVED	Reserved for future use	8 × n

Figure E.8/H.323 – Ack payload

E.1.4.2.2.3 Nack message

The following structure shall be used to encode Nack messages. The transport-message octet shall be set to 2 (two). The Nack message shall be used to signal transient errors, or more serious errors, such as the arrival of a malformed message. Unexpected Nack messages (such as ones bearing illegal sequence numbers) shall be ignored. See Figure E.9.



Field	Content of fields	Bits
NACK COUNT	The number of SEQNUM fields that follow	16
SEQNUM 0, ..., N–1	The Sequence Numbers of the PDUs that is being NACKed for	24 × n
LENGTH 0, ..., N–1	Length of Nack-specific data	8 × n
REASON 0, ..., N–1	The reason for the NACK	16 × n
OCTETs	Nack-specific data octets	8 × n

Reason value	Nack reason meaning	Length of Nack Data in octets	Data
0	Non-standard reason	1 + n	LENGTH OCTET followed by OBJECT IDENTIFIER OCTET(s)
1	Request the sender to use an alternate port for the specified static payload type	8	As defined in Figure E.10
2	Request the sender to use an alternate port for the specified ObjectID payload type	1 + n + 6	As defined in Figure E.11
3	Transport-payload not supported	1	Unsigned integer
4	Static-payload type not supported	1	Unsigned Integer; Payload as defined in the static-typed protocol that is not supported
5	Object-ID payload not supported	1 + n	LENGTH OCTET followed by OBJECT IDENTIFIER OCTET(s)
6	Payload Corrupted	1	The Payload number in the message that was corrupted
7.. 65535	Reserved for future use		

Figure E.9/H.323 – Nack message

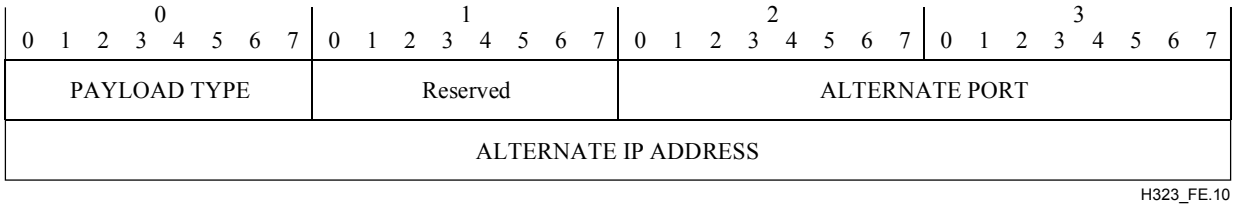


Figure E.10/H.323 – Nack reason 1 structure

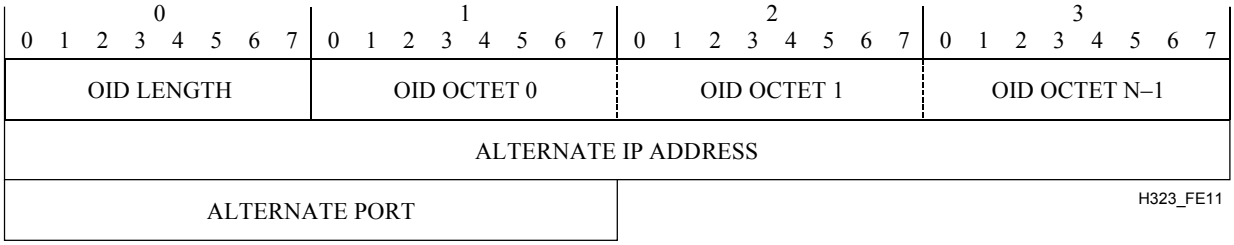


Figure E.11/H.323 – Nack reason 2 structure

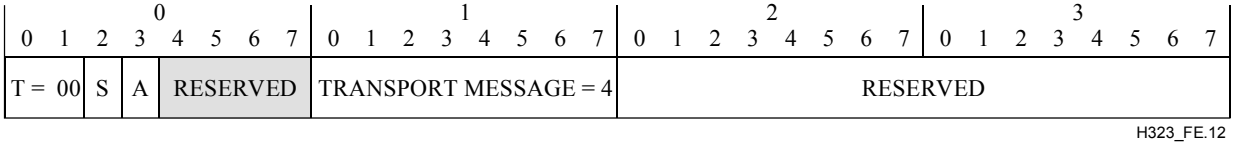
If the IP address is set to zero, the IP address of the sender shall be used (as identified by the TCP/IP layer). If the UDP port is set to zero, the port transmitted from shall be used (as identified by the TCP/IP layer).

E.1.4.2.2.4 Restart message

The following structure shall be used to encode Annex E Restart payloads. The transport-message octet shall be set to 3. Restart payloads are used to signal to the remote peer that the sender has restarted. Restart payload should be sent as a part of the first message to the remote entity. The receiver shall reset its receiver sequence number range on receiving the Restart payload. It shall consider any message arriving from the previous sequence number range as stale and shall ignore it.

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

If a restart does not affect ongoing calls, then it is invisible to the Annex E layer, and therefore shall not be signalled. See Figure E.12.



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

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

Figure E.12/H.323 – Restart message structure

E.1.4.3 Static-typed messages

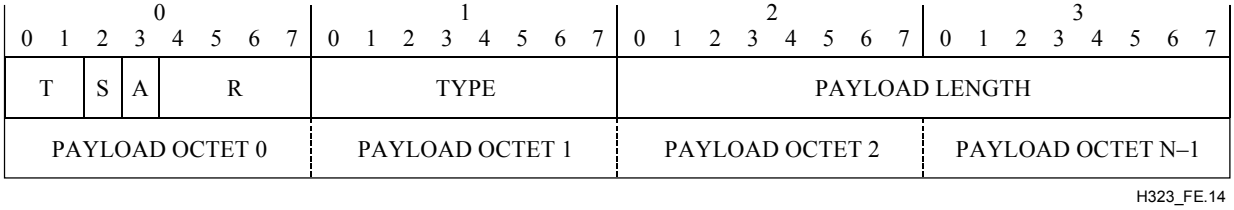
The first T bit in the Payload header flags OCTET shall be set to 1 (one) for all static-typed messages. The second T bit in the Payload header flags OCTET shall be set to 0 (zero) for all static-typed messages. The next octet shall signify what static-payload is present (see Figure E.13):

Value	Interpretation
0	Octet-stream contains a call signalling message as defined in ITU-T Rec. H.225.0
1..255	Reserved for future use

Figure E.13/H.323 – Static-typed payloads

E.1.4.3.1 Basic static-typed message (S-bit and A-bit cleared)

When both the S and A bits are cleared, the following payload format shall be used (see Figure E.14):

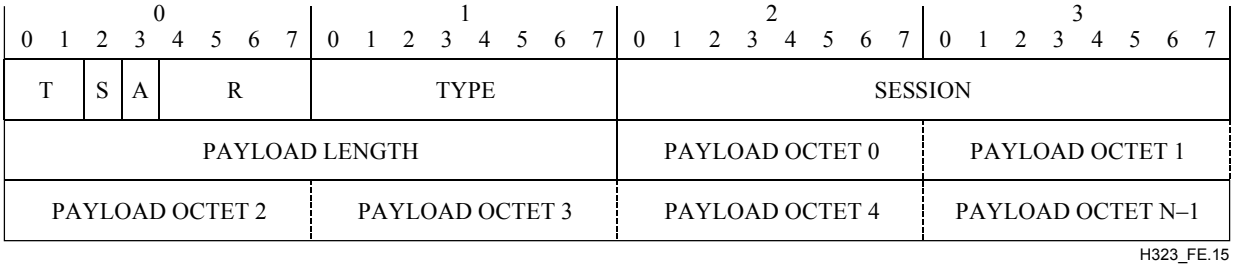


Field	Content of fields	Bits
TYPE	Unsigned Integer: the type of the payload, as defined in Figure E.13	8
LENGTH	Unsigned Integer: The length (in OCTETS or BYTES) of the payload data	16
DATA	The actual payload data OCTETs	8 × n

Figure E.14/H.323 – Basic static-typed payload

E.1.4.3.2 Extended-1 static-typed message (S-bit set and A-bit cleared)

When the S-bit is set and the A-bit is cleared, the following payload format shall be used. The S-bit signifies the presence of a SESSION field. See Figure E.15.

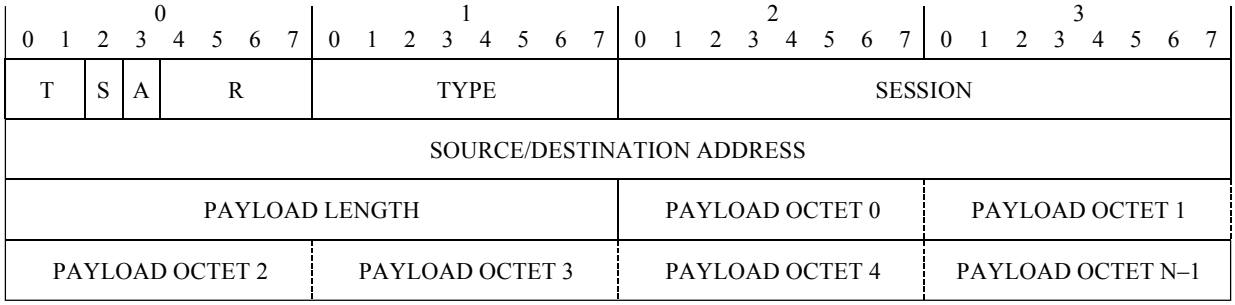


Field	Content of fields	Bits
TYPE	Unsigned Integer: The type of the payload, as defined in Figure E.13	8
SESSION	Unsigned Integer: The meaning of the session field is protocol dependent	16
PAYLOAD LENGTH	Unsigned Integer: The length (in OCTETS or BYTES) of the payload data	16
DATA	The actual payload data OCTET(s)	8 × n

Figure E.15/H.323 – Extended-1 payload format

E.1.4.3.3 Extended-2 static-typed message (S-bit and A-bit set)

When both the S-bit and the A-bit is set, the following payload format shall be used. The A-bit signifies the presence of a Source/Destination Address field. See Figure E.16.



Field	Content of fields	Bits
TYPE	Unsigned Integer: The type of the payload, as defined in Figure E.13	8
SESSION	Unsigned Integer: The meaning of the session field is protocol dependent	16
SOURCE/DESTINATION ADDRESS	Unsigned Integer: The meaning of the source/destination address field is protocol dependent	32
PAYLOAD LENGTH	Unsigned Integer: The length (in OCTETS or BYTES) of the payload data	16
DATA	The actual payload data OCTET(s)	$8 \times n$

Figure E.16/H.323 – Extended-2 payload format

E.1.4.3.4 Extended-3 static-typed message (S-bit cleared, A-bit set)

When the S-bit is cleared and the A-bit is set, the following payload format shall be used. The A-bit signifies the presence of a Source/Destination Address field. See Figure E.17.

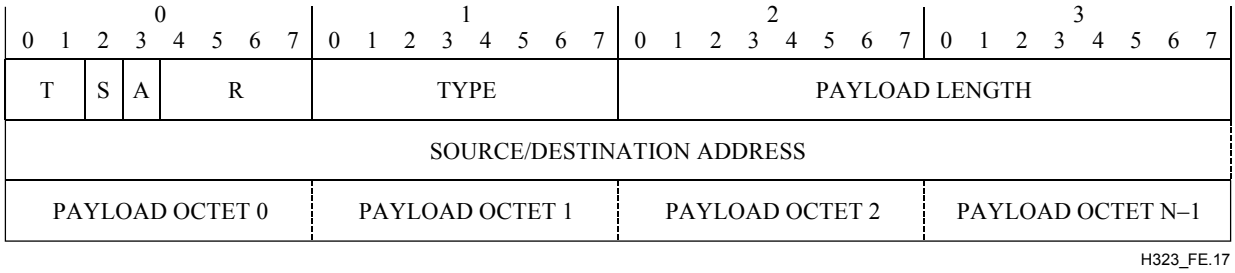


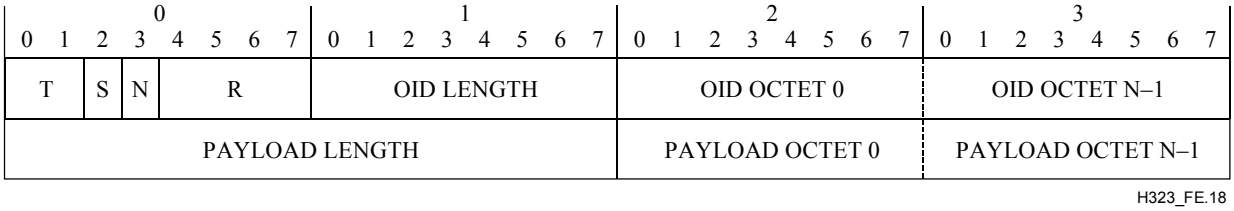
Figure E.17/H.323 – Extended-3 payload format

E.1.4.4 ObjectID-typed messages

The first T bit in the Payload header flags OCTET shall be set to 0 (zero) for all ObjectID-typed messages. The second T bit in the Payload header flags OCTET shall be set to 1 (one) for all ObjectID-typed messages. The next two octets shall signify the length of the Object-ID that follows.

E.1.4.4.1 Basic ObjectID-typed message (S-bit and A-bit cleared)

When both the S and A bits are cleared, the following payload format shall be used (see Figure E.18):

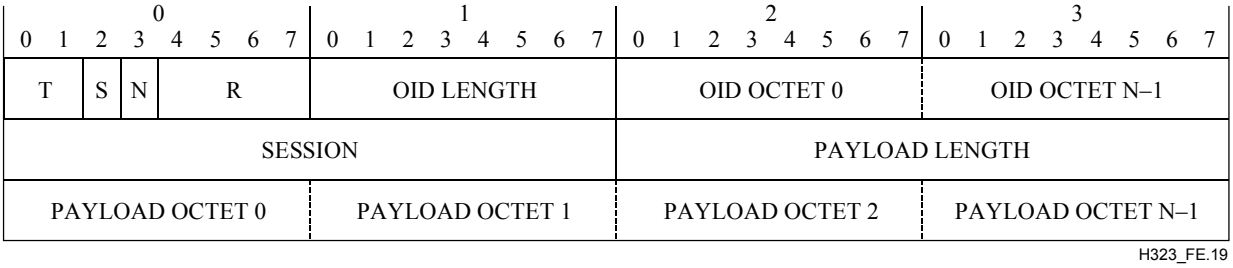


Field	Content of fields	Bits
OID LENGTH	Unsigned Integer: The length in OCTETs of the Object Identifier following	8
ObjectID	Object Identifier OCTETs	8 × n
LENGTH	Unsigned Integer: The length (in OCTETS or BYTES) of the payload data	16
DATA	The actual payload data OCTETs	8 × n

Figure E.18/H.323 – Basic ObjectID-typed payload

E.1.4.4.2 Extended-1 ObjectID-typed message (S-bit set and A-bit cleared)

When the S-bit is set and the A-bit is cleared, the following payload format shall be used. The S-bit signifies the presence of a SESSION field, which is used by the application to associate payloads with a specific session. The definition of a session is protocol specific. See Figure E.19.

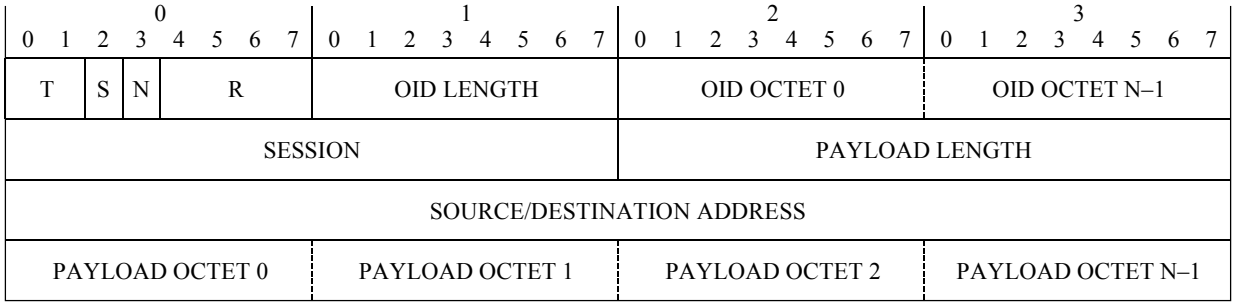


Field	Content of fields	Bits
OID LENGTH	Unsigned Integer: The length in OCTETs of the Object Identifier following	8
ObjectID	Object Identifier OCTETs	8 × n
SESSION	Unsigned Integer: The meaning of the session field is protocol dependent	16
LENGTH	Unsigned Integer: The length (in OCTETS or BYTES) of the payload data	16
DATA	The actual payload data OCTETs	8 × n

Figure E.19/H.323 – Extended-1 ObjectID-typed payload format

E.1.4.4.3 Extended-2 ObjectID-typed message (S-bit and A-bit set)

When both the S-bit and the A-bit are set, the following payload format shall be used. The A-bit signifies the presence of a Source/Destination Address field. See Figure E.20.



Field	Content of fields	Bits
OID LENGTH	Unsigned Integer: The length in OCTETs of the Object Identifier following	8
ObjectID	Object Identifier OCTETs	8 × n
SESSION	Unsigned Integer: The meaning of the session field is protocol dependent	16
LENGTH	Unsigned Integer: The length (in OCTETS or BYTES) of the payload data	16
SOURCE/DESTINATION ADDRESS	Unsigned Integer: The meaning of the source/destination address field is protocol dependent	32
DATA	The actual payload data OCTETs	8 × n

Figure E.20/H.323 – Extended-2 ObjectID-typed payload format

E.1.4.4.4 Extended-3 ObjectID-typed message (S-bit cleared, A-bit set)

When the S-bit is cleared and the A-bit is set, the following payload format shall be used. The A-bit signifies the presence of a Source/Destination Address field. See Figure E.21.

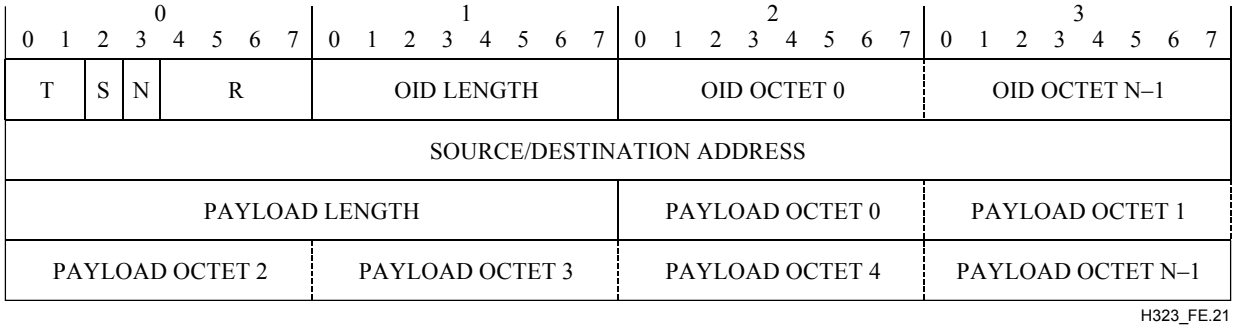


Figure E.21/H.323 – Extended-3 ObjectID-typed payload format

E.2 H.225.0 call signalling over Annex E

This clause describes how to carry H.225.0 Call Signalling messages using the Annex E transport, over UDP. Annex E is used to provide a "reliable-UDP" transport, to allow H.225.0 implementations to work over Annex E largely unchanged.

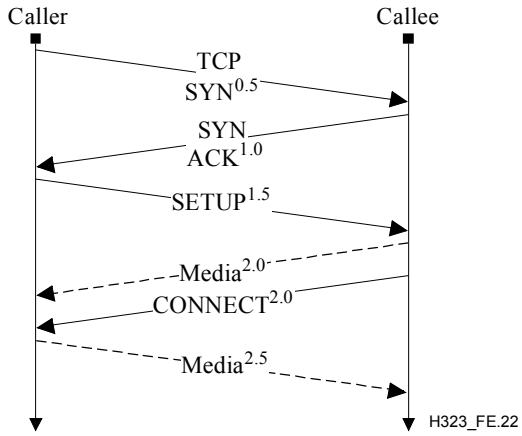
E.2.1 Rationale

ITU-T Rec. H.323 version 2 (1998) introduces the concept of "Fast Connect", which allows media cut-through in as little as in 2 round trips from callee to caller (including TCP messages), and in 2.5 round-trips from caller to callee.

This can be reduced to 1rt and 1.5rt respectively by using UDP as the transport for H.323 messages, instead of TCP. This is especially important when using the Gatekeeper-Routed-Model.

E.2.2 H.323 Call-Setup using this annex

ITU-T Rec. H.323 version 2 (1998) uses the TCP transport to carry H.225.0 messages, which means the least number of round trips possible to get media cut-through is 2 from Callee to Caller, and 2.5 from Caller to Called party. See Figure E.22.



NOTE – Some messages in the TCP handshake procedure have been omitted for clarity.

Figure E.22/H.323 – Information flow for H.323 version 2 (1998) FastConnect

E.2.2.1 UDP-based procedure

To get faster media cut-through, it is possible to use UDP for call signalling transport, which effectively enables media cut-through in a single round trip (see Figure E.23):

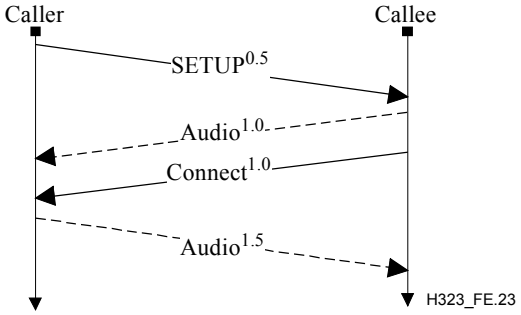


Figure E.23/H.323 – Information flow for UDP-based Call Setup

The Annex E layers 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.

E.2.2.2 Mixed TCP and UDP procedure

The procedures for TCP-based and UDP-based call setup are not mutually exclusive. If UDP-based and TCP-based call setup are carried out in parallel then the procedure in this clause shall be used. In the mixed procedure the originator transmits the SETUP message over UDP, and simultaneously establishes a TCP connection. If the originator has not received a response to the UDP SETUP when the TCP connection is established, then it also transmits the SETUP messages over the

TCP connection. If a callee receives the same SETUP message over UDP and over TCP, then it shall respond using either transport protocol (usually the one which arrived first) but not both.

If the originator receives a response over UDP then the TCP connection shall be released and communication continues over UDP. If the originator receives a response over TCP (for example because the remote peer does not support the Annex E procedures), then communication continues over TCP, UDP-based communication shall no longer be used for this call.

A callee that supports this annex shall select the transport protocol according to which arrives first: TCP Setup message, or UDP Setup message. Note these messages may be reordered in delivery. The caller is notified of the selection according to the transport protocol over which the subsequent message (e.g., Connect) has arrived. See Figure E.24.

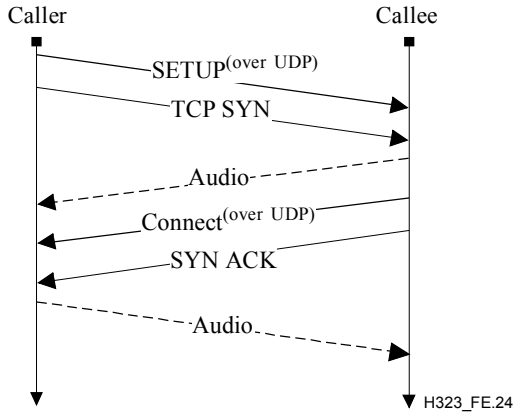


Figure E.24/H.323 – Information flow for mixed TCP and UDP procedure

This ensures that if the UDP procedure fails, usual TCP-based procedures can take over immediately (see Figure E.25):

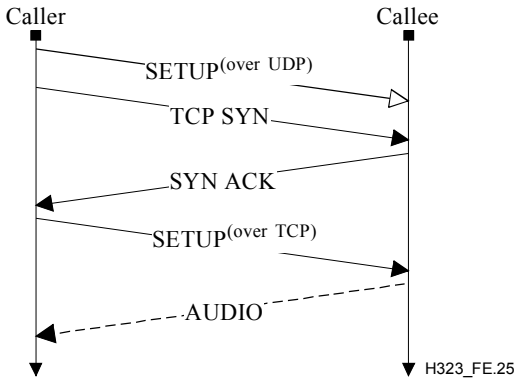


Figure E.25/H.323 – Information flow when UDP is not supported

This means that backwards compatibility when calling ITU-T Rec. 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.

NOTE – It is recommended for entities that initiate a call and do not know if the remote side supports Annex E operations to use the procedure detailed above. If the calling entity knows by some means that the remote callee supports UDP-based operations, it may use a UDP only call setup.

E.2.3 Specifics

E.2.3.1 Message identification

H.225.0 over Annex E payloads shall use static payload type **0** (zero).

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. A single H.323 entity on a physical device shall use a single, distinct UDP port as the advertised port for receiving messages. However, it may utilize a distinct port on each interface if the physical device has multiple network interfaces.

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

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

E.2.3.3 Signalling model

H.225.0 over Annex E shall use the **serial-model** as described in E.1.2.2.

E.2.3.4 Timers

H.225.0 over Annex E shall use default timers and values. The **T-IMA1** timer shall be reset upon reception of any Call Signalling message (e.g., but not when receiving RTP packets).

E.2.3.5 Session field

The session field shall be present in all payloads. The Session value shall contain the CRV from the H.225.0 call signalling messages. Specifically, the call reference flag shall be included as the most significant bit of the CallReferenceValue. This restricts the actual CRV to the range of 0 through 32 767, inclusive.

E.2.3.6 Source/destination address field

Use of the Source/Destination field is optional, but shall be present in all messages originating, or destined to an MCU or when a Gatekeeper acts as an MC.

E.2.3.7 MTU

Call-Signalling messages that require sending large amounts of data (such as certificate-based authentication and authorisation) should use TCP for call-setup, as using them over this annex may cause fragmentation due to messages being larger than path MTU.

E.2.3.8 H.245

H.245 shall be transmitted using the ITU-T Rec. H.323 version 2 (1998) H.245 Tunnelling procedures.

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

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

Annex F

Simple endpoint types

F.1 Introduction

Simple Endpoint Types, i.e., devices manufactured for a single purpose, may comprise a significant fraction of the overall set of H.323 capable end systems. In contrast to full-featured H.323 devices (many implementations of which are PC-based), the so-called Simple Endpoint Types (SETs) may be implemented in inexpensive stand-alone boxes, the most prominent example being the simple telephone.

NOTE – Sample application scenarios for such systems were found to include:

1) palmtop computer with audio communications capabilities (voice, file transfer, fax, etc.);
2) telephone with an RJ-45 connector;
3) text telephones (using ITU-T Rec. T.140);
4) cellular IP phone;
5) mobile system with integrated voice and data communications (UMTS, IMT-2000).

All these systems have in common that they support a relatively fixed set of functionality: voice and/or rudimentary (i.e., not T.120) data communication facilities. It is important to note that this functionality does not need to be extended for the respective system's purpose: a telephone set without (an elaborate) display does not need to support video functionality, neither does it require data conferencing capabilities.

All of these systems have a limited amount of resources available (e.g., processing power, communication bandwidth, memory).

This annex outlines the scope of SET devices in general and defines the procedural and protocol details of a Simple Audio Endpoint Type (Audio SET device). In particular, this annex defines the functional baseline for all types of Simple Endpoint Types; hence, further SETs are to be defined by referencing this annex and then only specifying additions to the procedures and conventions set forth in this annex.

This annex defines a subset of H.323 functionality and any deviations from Recommendation H.323 are explicitly identified. Any procedures not explicitly described in this annex are covered by the main body of this Recommendation.

The development of SET devices has potential implications on other H.323 devices: in particular, MC(U)s and gateways should be aware of their potentially minimal support for H.323 (1998) functionality in order to provide SET devices with seamless access to enhanced H.323 services such as multipoint conferences and supplementary services. Alternatively, external proxy devices may be provided to bridge the different functional ranges between SET devices and full-featured H.323 (1998) endpoints. Interoperability issues are addressed in more detail in F.9.

F.2 Specification conventions

This annex specifies only those services, procedures, protocol messages, etc. that are mandatory for the implementation of a SET device, which is a subset of the mandatory functionality of an H.323 (1998) system. This implies that a SET device shall not assume any functionality beyond what is specified mandatory in this annex from another SET device.

In addition to the mandatory components, several clauses of this annex specify conditionally mandatory services, procedures, protocol messages, etc. based on the concept of functional blocks that are optional as a whole. However, a SET device that decides to implement a particular

functional block, must support all the components defined as mandatory for this functional block; optional components may be supported.

All other features defined in ITU-T Rec. H.323 are, by definition, optional, and their implementation in a SET device is entirely at the discretion of the manufacturer.

F.3 Scope

This annex specifies rules on the use of ITU-T Rec. H.323 that enable Simple Endpoint Types to be implemented in a simple fashion. The following (non-exhaustive) list of Simple Endpoint Types is envisioned for standardization by the ITU-T:

1) **simple telephone (Simple Audio Endpoint Type)** – defined in this annex;
2) **simple telephone with security capabilities** – for further study;
3) **text conversation terminal** – for further study;
4) **fax device** – for further study.

The simple telephone is defined in this annex. Secure simple telephone, text terminal, and simple fax device are Simple Endpoint Types for further study. The profiles for Simple Endpoint Types can be categorized as follows:

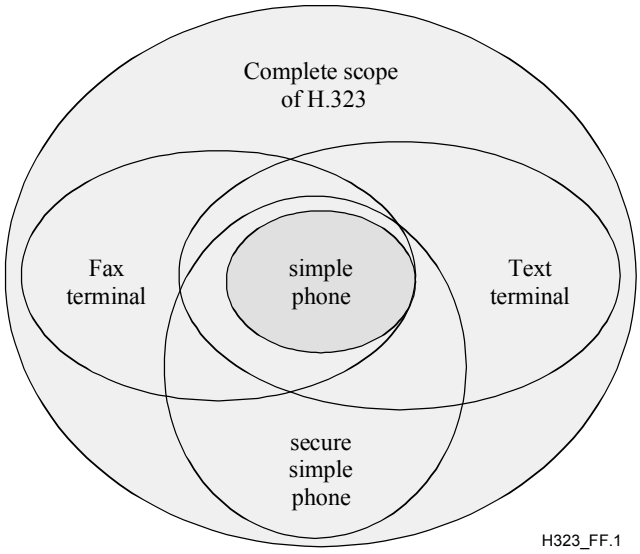


Figure F.1/H.323 – Venn diagram showing functional ranges of the various SET devices

Figure F.1 is a schematic picture of the different Simple Endpoint Types that are being defined in the context of H.323 'profiles', in a so-called Venn diagram. In this diagram, the relation between the SETs is illustrated. The wider ellipse show the context of a full H.323 compliant system. As an example, the simple telephone is put in the figure. As it is clearly a subset of the full compliant H.323 system, it lays completely within its scope. A secure simple telephone, containing additionally the security capabilities, comprises the capabilities of the simple telephone (e.g., same audio codecs, same call setup, etc.). The interoperability between a simple telephone SET implementation and a secure simple telephone will therefore be ensured.

The SET devices are defined in a way that enables them to interoperate seamlessly with one another and with H.323 (1998) devices supporting the FastConnect procedure as well as with all SET-aware H.323 endpoints.

NOTE – Although SET devices are defined with simple devices in mind, it is equally possible to build gateways based upon the respective SET specification. No additional definitions are required for other types of devices.

F.4 Normative references

All the normative references in the body of this Recommendation and in ITU-T Rec. H.225.0 (2003) apply.

F.5 Abbreviations

This annex uses the following abbreviations:

Audio SET	Simple Audio Endpoint Type
Fax SET	Simple Facsimile Endpoint Type
Secure Audio SET	Secure Simple Audio Endpoint Type
SET	Simple Endpoint Type
Text SET	Simple Text Telephony Endpoint Type

F.6 Simple (Audio) Endpoint Type – System functionality overview

The following characteristics apply to Simple Audio Endpoint Types (Audio SET devices):

Media capabilities

- Voice-capability
  - mandatory: G.711 (A-law and μ-law)
  - suggested options: G.723.1, G.729, GSM.
  - suggested options: audio redundancy encoding with any combinations of the above codecs.
- Audio SET devices shall only support symmetric audio operation.
- No data-capability.
- DTMF capability mandatory; transmission as H.225.0 Information messages mandatory; transmission as RTP payload for further study.
- No video capability.
- No T.120 capability.
- Media distribution: support for unicast mandatory.

The mandatory and optional media capabilities shall be defined separately for other Simple Endpoint Types.

Control capabilities

The following minimum control capabilities shall equally apply to all Simple Endpoint Types.

- FastConnect sequence of ITU-T Rec. H.323 (1998) mandatory.

NOTE – Audio SET devices are by default capable of participating in multipoint conferences, where they are obviously limited to audio communications.

Most other control capabilities are optional, in particular:

- UDP-based Faster-Connect Annex E/H.323 optional.
- Supplementary services (solely based upon H.450.x) optional.
- Support for H.245 messages and procedures optional.
- Support of more than a single call/conference at a time is optional.

Some control capabilities are disallowed for Audio SET devices.

- MC functionality prohibited.

F.7 Procedures for Simple Endpoint Types

This clause specifies for all the protocols required by this Recommendation, the detailed level of support by SET devices in general, and the specific requirements for Audio SET devices:

- Registration, Admission and Status (RAS) signalling (H.225.0), see F.7.1;
- Call signalling (H.225.0), see F.7.2;
- Multimedia system control signalling (H.245), see F.7.3;
- Media packetization and transport (H.225.0, RTP), see F.7.4;
- Supplementary Services (H.450.x), see F.7.5 and F.7.6;
- Multipoint conference operation, see F.7.7;
- Loosely-coupled conferences (H.332), see F.7.8;
- Management Information Bases, see F.7.9.

Security services as specified in ITU-T Rec. H.235 to create Secure SET devices are considered in F.8.

F.7.1 RAS Signalling (H.225.0 RAS)

SET devices shall comply with the RAS procedures as defined in ITU-T Recs H.323 (1998) and H.225.0 (1998) with the following modifications applying.

A SET device shall use the pre-granted ARQ procedures as specified in ITU-T Rec. H.225.0 (1998) and shall be able to determine whether an incoming call request is received from its Gatekeeper. A SET-aware Gatekeeper shall support pre-granted ARQ and shall pre-grant for placing and receiving calls with call routing through the Gatekeeper for SET devices (to be indicated in the preGrantedARQ component). If a contacted Gatekeeper does not support pre-granted ARQ or does not provide the aforementioned pre-granting configuration, a SET device shall register with another Gatekeeper.

SET devices shall at a minimum support the following RAS messages: transmission of GRQ, RRQ, URQ, UCF, and XRS and reception of GCF, GRJ, RCF, RRJ, URQ, UCF, URJ, and XRS. SET devices may support additional RAS messages.

A SET device shall include the "set" component of the H.225.0 EndpointType when communicating with a Gatekeeper and set the bits as follows.

Bit 0: = 1 if the device has Audio SET functionality.

Bit 1: = 0 if the device is not conference-aware.

Bit 1: = 1 if the device is conference-aware.

Use of the other bits will be defined by additional SET specifications.

F.7.2 Call signalling (H.225.0 Call Control)

SET devices shall comply with the call control procedures defined in ITU-T Recs H.323 (1998) and H.225.0 (1998). SET devices shall not close the call signalling channel after call establishment.

SET devices shall implement the FastConnect procedures as specified in ITU-T Rec. H.323 (1998). When originating a call, a SET device shall place a call using FastConnect.

SET devices shall support H.225.0 Information messages in the call signalling channel. Such messages should be used for, but are not restricted to, conveying user input in the Keypad Information Element.

SET devices should use the Status Enquiry and Status messages of ITU-T Rec. H.225.0 to estimate round-trip times to its peer.

SET devices may implement UDP-based call setup as outlined in Annex E/H.323. If UDP-based call setup is implemented, a SET device should attempt to call another endpoint via UDP-based call setup first.

Implementation of supplementary services based upon H.450.x is optional for SET devices. SET devices shall be able to safely ignore H.225.0 Facility messages that they do not understand.

A SET device shall include the "set" component of the H.225.0 EndpointType when exchanging call signalling PDUs with its peer. The bits of the "set" components shall be set as defined in F.7.1.

F.7.3 Multimedia system control signalling (H.245)

F.7.3.1 H.245 control channel

The FastConnect procedure shall be used for connection establishment. Repeated transmission of the fastStart element in H.225.0 call signalling messages shall be used to reconfigure or re-route media streams.

SET devices shall not open a separate H.245 connection:

a) They shall restrict H.245 signalling to the **OpenLogicalChannel** structure in the FastConnect sequence along with implicit Master Slave Determination.

b) If further H.245 signalling is required, they shall perform tunnelling as defined in ITU-T Rec. H.225.0 (1998).

SET devices shall use the syntax of ITU-T Rec. H.245 (1998) or later versions.

No specific procedures are defined for H.245 messages. If SET devices implement H.245 functionality, they shall adhere to the procedures defined in ITU-T Recs H.323, H.225.0 and H.245.

F.7.3.2 Master-Slave Determination

SET devices shall implicitly assume the slave role in any communication relationship without an H.245 control channel.

In case an H.245 tunnel is established, following the rules of 6.2.8.4/H.323 (1998), the SET device shall indicate a value of 40 for the **terminalType**. This ensures that in case a SET device connects to a full H.323 (1998) device, the latter will win the Master-Slave-Determination.

F.7.3.3 Terminal capability exchange

Although SET devices are by definition restricted in their supported functional range, a capability exchange procedure cannot be circumvented to allow for a minimum of diversity in the devices. However, the range of possible capabilities that may be signalled by a SET endpoint is restricted to what is defined in the following, and the capability exchange procedures shall adhere to the rules set forth in this subclause.

The Capability Exchange procedure for media types and transmission modes shall be carried out following the rules of the FastConnect procedure using multiple Open Logical Channel structures as a selection of possibilities offered by the caller out of which the callee chooses a subset to send and receive.

The following subclause list which capabilities need to be understood on the receiving (called) side and which may be transmitted on the sending (calling) side for Audio SET devices.

F.7.3.3.1 Audio capability

- G.711 (μ-Law, A-Law, 56 kbit/s, 64 kbit/s)
  The following alternatives shall be supported:

```
AudioCapability.g711Alaw64k         ≥ 20     number of frames
AudioCapability.g711Alaw56k         ≥ 20     number of frames
AudioCapability.g711Ulaw64k         ≥ 20     number of frames
AudioCapability.g711Ulaw56k         ≥ 20     number of frames
```

- G.723.1 (silence suppression or not, low and high rate)

  A SET supporting G.723.1 must at a minimum support:

```
AudioCapability.g7231
   maxAl-sduAudioFrames             ≥ 1      number of frames
   silenceSuppression                        True/False as appropriate
```

- G.729 (plain or Annex A)

  A SET supporting G.729 must at a minimum support:

```
AudioCapability.g729                ≥ 1      number of frames
AudioCapability.g729AnnexA          ≥ 1      number of frames
```

- GSM (full rate, enhanced full rate, half rate).

  A SET supporting GSM must at a minimum support:

```
AudioCapability.gsmFullRate         GSMAudioCapability,
AudioCapability.gsmHalfRate         GSMAudioCapability,
AudioCapability.gsmEnhancedFullRate GSMAudioCapability
```

  with GSMAudioCapability defined as appropriate for each of these rates:

```
GSMAudioCapability.audioUnitSize    ≥ 1      number of frames
GSMAudioCapability.comfortNoise              True/False as appropriate
GSMAudioCapability.scrambled                 True/False as appropriate
```

F.7.3.3.2 Video Capability

Audio SET devices do not support video.

F.7.3.3.3 Data Capability

Audio SET devices do not support data.

F.7.3.3.4 Conference Capability

SET devices are assumed to be proxied into centralized conferences with centralized data distribution (see F.7.7).

F.7.3.3.5 User Input Capability

SET devices shall support transmission of DTMF as Keypad Information Elements in the H.225.0 call signalling connection (e.g., using Information messages).

F.7.3.3.6 Security Capability

Security for SET devices, i.e., the definition of Secure SET devices, is for further study. Refer also to F.8.

F.7.3.3.7 maxPendingReplacementFor

Shall be supported by Audio SET devices. A value equal to '1' shall be implicitly assumed:

```
maxPendingReplacementFor = 1
```

Hence, the **maxPendingReplacementFor** parameter shall not be signalled explicitly.

F.7.3.3.8 nonStandardCapability

Use of non-standard capabilities, on the top level of the capability structure as well as within the aforementioned capability categories, should be avoided as far as possible.

F.7.3.3.9 Additional rules for the use of capabilities

For Audio SET devices, audio capabilities shall only be signalled via the FastConnect procedure and repeated exchange of **OpenLogicalChannel** structures using the FastConnect.

Video capabilities, data capabilities, conference capabilities, security capabilities, and h233encryption capabilities shall not be used.

The values of the MultiplexCapability table entry of an Audio SET device shall be assumed as follows:

maximumAudioDelayJitter	≥ 250 ms
receiveMultipointCapability, transmitMultipointCapability, and receiveAndTransmitMultipointCapability	
multicastCapability	TRUE/FALSE as appropriate, default FALSE[1]
multiUnicastConference	TRUE/FALSE as appropriate, default FALSE[1]
mediaDistributionCapability	
centralizedControl	TRUE
distributedControl	FALSE
centralizedAudio	TRUE
distributedAudio	TRUE/FALSE as appropriate, default FALSE[1]
centralizedVideo	FALSE
distributedVideo	FALSE
centralizedData	ABSENT
distributedData	ABSENT
mcCapability	
centralizedConferenceMC	FALSE
decentralizedConferenceMC	FALSE
rtcpVideoControlCapability	ABSENT
mediaPacketizationCapability	ABSENT
...,	
transportCapability	ABSENT
redundancyEncodingCapability	Audio redundancy encoding only (if any)
logicalChannelSwitchingCapability	FALSE
t120DynamicPortCapability	FALSE

Capabilities signalled from the remote side that are not understood shall be ignored.

---

[1] Multicast, multi-unicast, and distributed audio may be supported by Conference-aware Audio SET devices.

F.7.3.4 Logical Channel Signalling Messages

The opening of logical channels shall adhere to the FastConnect specifications of ITU-T Rec. H.323 (1998).

In addition, SET devices shall support reconfiguration of media streams at any time during a call. Open Logical Channel structures shall be tunnelled in H.225.0 call signalling messages following the procedures defined in ITU-T Recs H.225.0 (1998) and H.323 (1998) reusing the fastStart element of the H.225.0 call signalling message. Open Logical Channel structures outside the FastConnect procedure shall be used to alter media stream parameters – to provide a basis for supplementary services. Such Open Logical Channel structures shall be interpreted upon reception as follows.

- If the logical channel number matches a currently open logical channel, the respective channel shall be reconfigured following the principles of the FastConnect procedure if the **dataType** component is not "null". If the **dataType** component is "null" – indicating a "NullChannel" – the respective logical channel shall be considered closed and media transmission on this logical channel shall cease.
- If the logical channel number does not match a currently open channel, a new logical channel shall be opened following the princicples of the FastConnect procedure.

In the following, the restrictions on Open Logical Channel request are outlined:

```
OpenLogicalChannel
    forwardLogicalChannelNumber         LogicalChannelNumber
    forwardLogicalChannelParameters
        portNumber                      ABSENT
        dataType                        a valid audio data type
                                        (see F.7.3.3.1)
        multiplexParameters             CHOICE:h2250LogicalChannelParameters
        forwardLogicalChannelDependency ABSENT,
        replacementFor                  used if another Logical Channel is
                                        to be replaced

    reverseLogicalChannelParameters
        dataType                        a valid audio data type
                                        (see F.7.3.3.1)
        multiplexParameters             CHOICE:h2250LogicalChannelParameters
        reverseLogicalChannelDependency LogicalChannelNumber OPTIONAL,
        replacementFor                  used if another Logical Channel is
                                        to be replaced
    separateStack                       ABSENT
    encryptionSync                      ABSENT for Audio SET devices; FFS.
```

To the **H2250LogicalChannelParameters** structure, the following restrictions apply:

```
H2250LogicalChannelParameters
    nonStandard                     should be ABSENT
    sessionID                       INTEGER(0..255)
    associatedSessionID             ABSENT
    mediaChannel                    TransportAddress - should be a unicast
                                    address
    mediaGuaranteedDelivery         ABSENT
    mediaControlChannel             PRESENT - reverse RTCP channel
    mediaControlGuaranteedDelivery  FALSE
    silenceSuppression              as appropriate
    destination                     typically ABSENT
    dynamicRTPPayloadType           as appropriate,
    mediaPacketization              as appropriate; may only specify the
                                    payload format used
```

```
    rtpPayloadType
    payloadDescriptor           should refer to an rfc-number
    payloadType                 (dynamic) payload type value to be used

transportCapability
    nonStandard                 should be ABSENT
    qOSCapabilities             should be ABSENT (may only contain RSVP
                                parameters)
    mediaChannelCapabilities    should be ABSENT (may indicate "ip-udp")

redundancyEncoding              optional; only audio redundancy is
                                allowed
source                          typically ABSENT
```

F.7.4 Media exchange

For media exchange, SET devices shall follow the H.323 and H.225.0 procedures using RTP/UDP/IP to convey the media streams. The appropriate media packetization formats shall be used.

F.7.5 Supplementary services (H.450.x)

Support of any of supplementary services according to the H.450.x series of Recommendations is optional.

NOTE – If H.450.x functionality is not provided by a SET device, the SET device should implement the message rejection functionality (Interpretation APDU) of H.450.1 to enable its peer to quickly determine non-availability of supplementary services on side of the SET device. If H.450.1 message rejection is not implemented, the peer has to rely on a timeout.

A baseline for supplementary services to be supported by SET devices is for further study.

F.7.6 Third-party initiated pause and re-routing

Support for third-party initiated pause and re-routing is similar to the procedures outlined in 8.4.6/H.323 (1998), with the following modifications applying.

F.7.6.1 Initiating side

To re-route a call connecting to a SET device its peer (typically a Gatekeeper) shall transmit a NullChannel specification in the fastStart element in a message of the call signalling channel.

Subsequently, the initiating entity shall again transmit the (for the new peer) appropriate **OpenLogicalChannel** structures, similar to the capability negotiation and media stream establishment in the FastConnect procedure, and include the new transport addresses to redirect the media stream sourced by the SET device. The **OpenLogicalChannel** structures are carried in an H.225.0 call signalling message.

The **OpenLogicalChannel** structure should offer the same audio encodings that were offered in the initial call.

F.7.6.2 Receiving side (SET device)

Upon reception of a NullChannel specification in a fastStart element, a SET device shall stop transmitting the media stream(s) immediately and shall be prepared to handle interruptions in the received media stream(s). The SET device shall expect a repeated exchange of capability and transport addresses following the principles of the FastConnect procedure.

Upon reception of an **OpenLogicalChannel** structure carried in an H.225.0 call signalling message, the SET device shall select an acceptable media encoding from the selection offered by the initiating entity, following the rules of the FastConnect procedure. The SET device shall then start transmitting its media stream(s) to the transport address(es) newly indicated in the **OpenLogicalChannel** structures.

F.7.7 Conference-mode operation

SET devices may participate in multipoint conferences in either of two ways:

- by being proxied into a conference through a dedicated external device, such as a SET-aware MC combined with a suitable MP or a SET-specific proxy as outlined in F.7.7.1 as the default mode of operation for SET devices; or
- by implementing the necessary procedures of the H.225.0 and H.245 protocols as outlined in this clause. This mode of operation is defined in F.7.7.2.

F.7.7.1 Conference-unaware SET devices

The default mode of operation for SET devices does not require any awareness of conferencing functionality in a SET device itself. Instead, an external entity is assumed that bridges between a full-featured H.323 device and the SET device. This logical entity may be a stand-alone proxy device or may be part of an MC(U), a Gateway, or a Gatekeeper.

NOTE – The functionality of a logical bridging entity may include the following:

- concealing the existence of conference-related H.245 commands and responding appropriately in the direction of the full-featured H.323 device;
- adapting H.245 capability and logical channel signalling including multipoint mode commands;
- mixing several incoming audio streams and providing a single stream to the SET device;
- translating transport addresses for the audio stream;
- transcoding audio streams; and
- offering access to conference control functions via simple input means (such as DTMF signalling) to the SET device.

F.7.7.2 Conference-aware SET Devices

The specification of conference-aware SET devices is for further study.

Nevertheless, SET devices may follow the full procedures for conference-mode operation defined in the H.323 series of Recommendations.

F.7.8 Support for loosely-coupled conferences (ITU-T Rec. H.332)

Support for loosely-coupled conferences according to ITU-T Rec. H.332 is optional:

- Participation as a member of the panel is optional; it is provided either if conference-mode operation and media distribution via multicast are supported, or if an appropriate MC/MP combination hides all the conference commands from the SET device and only presents a single audio-stream.
- Participation as a member of the audience is optional; it is possible if the SET device supports multicast reception of information and is capable of receiving and interpreting H.332 session announcements.

F.7.9 Management Information Bases (MIBs)

Implementation of Management Information Bases is optional for SET devices. If MIBs are included in the implementation, the following H.323-related MIBs should be implemented:

- Call signalling;
- Terminal entity;
- RAS;
- Real time Protocol (RTP).

Details are for further study.

F.8 Security extensions

Plain SET devices are not capable of supporting H.235 security services. Secure SET devices, however, define a simple extension to SET devices covering security functionality using a subset of the mechanisms specified in ITU-T Rec. H.235.

The details of Secure SET devices are covered in Annex J.

F.9 Interoperability considerations

This annex specifies a SET device as a well-defined subset of the total H.323 functionality.

SET devices should always be used in conjunction with SET-aware Gatekeepers. The SET-aware Gatekeeper shall perform pre-granted ARQ and shall employ the Gatekeeper-routed call model to ensure full interoperability with other H.323 (1996) and H.323 (1998) devices.

In addition, SET-awareness may be built into MC(U)s or gateways to achieve seamless interoperability.

Table F.1 presents an overview of interoperability achieved between Audio SET devices and other H.323 endpoints.

Table F.1/H.323 – Interoperability of SET devices with other H.323 devices

	H.323 (1996)	**H.323 (1998)**	**H.323 (1998) with Fast Connect**	**SET device**
H.323 (1996)	√	√	√	√ (GK)
H.323 (1998)	√	√	√	√ (GK)
H.323 (1998) with Fast Connect	√	√	√	√ a)
SET device	√ (GK)	√ (GK)	√ a)	√

(GK) Indicates that a SET-aware Gatekeeper is needed for interoperation.

a) Optional redirection of media channels requires repeated execution of FastConnect in both endpoints.

F.10 Implementation notes (Informative)

This clause provides informative text on simple encoding of most of the necessary H.245 messages without requiring specific ASN.1 encoders/decoders.

NOTE – All these messages are transmitted as tunnelled H.245 messages; i.e., the resulting bit patterns are encoded as a single OCTET STRING of the SEQUENCE in the fastStart component of a H323-UU-PDU. In the tables shown below, the leftmost octet (octet #0) of the first row (word #0) is placed in the first octet of the octet string, followed by octet #1 of the first row, and so on. Octet #3 of word #n is followed by octet #0 of word #(n+1).

If numbers are to be encoded, 2-complement encoding is used for numbers that may be negative. Otherwise, simple binary encoding is used. Encoding of numbers spanning multiple octets is done in a way that the most significant bit of the encoded value is located in the first octet of the value (network byte order).

F.10.1 Open Logical Channel

The **OpenLogicalChannel** structures are used by SET devices during the FastConnect procedure to indicate their capabilities and simultaneously open media channels in both directions and to reconfigure media streams during a conference. By definition, the **OpenLogicalChannel** structures contain only either forward logical channel parameters or backward logical channel parameters.

F.10.1.1 Forward Logical Channel parameters

An Open Logical Channel structure containing only **ForwardLogicalChannel** parameters may be coded in three different ways, depending on the audio type (AuType) and the X bit.

F.10.1.1.1 ITU-T Recs G.711 and G.729

The most common structure is the following (ITU-T Recs G.711, G.729 and Annex A/G.729):

	Octet #0								Octet #1								Octet #2								Octet #3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0x00								Logical Channel Number								0	0	0	0	1	1	X											
4	AuType		0	0	0	0	0		# samples								0x80								length = 0x0A							
8	0x04								0x00								session id								0	M	0	0	0	0	0	0
12	RTCP: IP address																															
16	RTCP: UDP port number																															

Logical Channel Number: This field contains the number of the H.245 logical channel – 1.

X bit: Used to distinguish between basic and extended audio types. If X = 0, AuType (see next field) applies; otherwise (X = 1), the extended audio types described below apply (primarily GSM) along with a different packet structure.

AuType: Identifies the audio codec to be used. The following values are acceptable for AuType. The leftmost bit is placed in bit 1 of octet #3 above, the rightmost in bit 5 of octet #4.

No.	Codec description	AuType value
1	G.711 A-law 64 kbit/s	0001
2	G.711 A-law 56 kbit/s	0010
3	G.711 μ-law 64 kbit/s	0011
4	G.711 μ-law 56 kbit/s	0100
5	G.723.1	1000
6	G.729	1010
7	Annex A/G.729	1011
8	GSM and others (see below)	X = 1

samples: For codecs 1, 2, 3, 4, 6, and 7 this component contains the number of samples – 1 per audio packet as defined in ITU-T Rec. H.245.

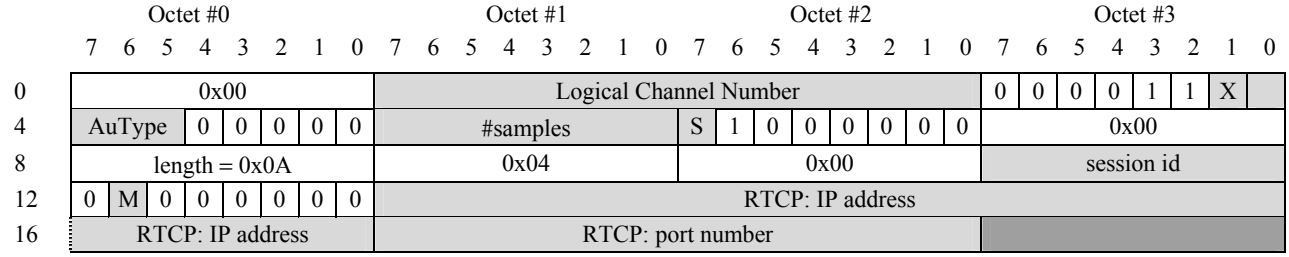
session id: Contains the session id parameter to be used in conjunction with RTP/RTCP.

M bit: Multicast address bit: indicates that the following address is a multicast address. While many address types are defined besides IPv4 (including IPv6 and IPX), the structures shown here are only valid for IPv4 addresses.

RTCP IP address/port: Contains the transport address for the RTCP receiver reports to be sent to.

F.10.1.1.2 G.723.1 codec

For ITU-T Rec. G.723.1, the structure is slightly modified as follows:

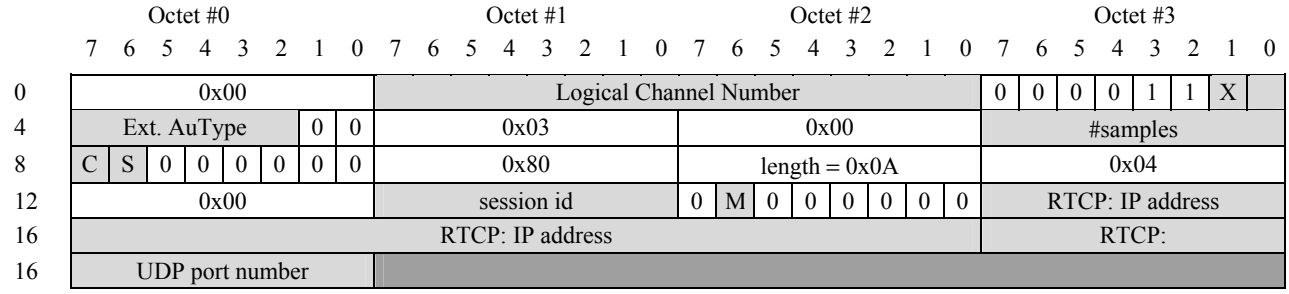


The meaning of the fields is identical to the meaning defined for the above format. In addition, the following fields are relevant:

S bit: Indicates support for silence suppression if S = 1.

F.10.1.1.3 GSM

For GSM, identified by bit #1 of octet #3 set to X = 1, the structure looks as follows:



The fields have the same meaning as in the above packet formats. In addition, the following fields are defined for GSM:

Ext. Audio Type: Identifies the extended audio codec:

GSM Full Rate = 000 0011

GSM Half Rate = 000 0100

GSM Enhance Full Rate = 000 0101

C bit: C = 1 indicates support/use of comfort noise

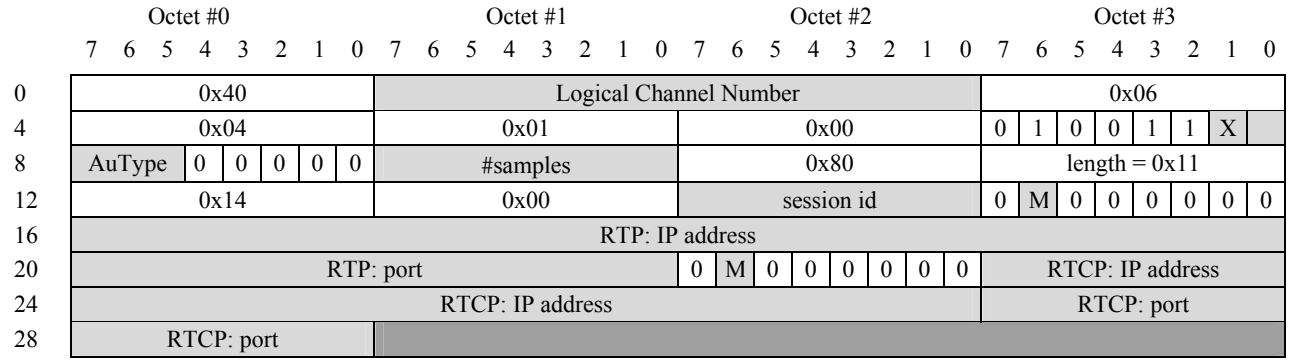
S bit: S = 1 indicates support/use of scrambling

F.10.1.2 Reverse Logical Channel Parameters

Open Logical Channel Message containing **ReverseLogicalChannel** parameters are encoded as described in this subclause.

F.10.1.2.1 ITU-T Recs G.711 and G.729

The most common structure is the following (ITU-T Recs G.711, G.729 and Annex A/G.729):



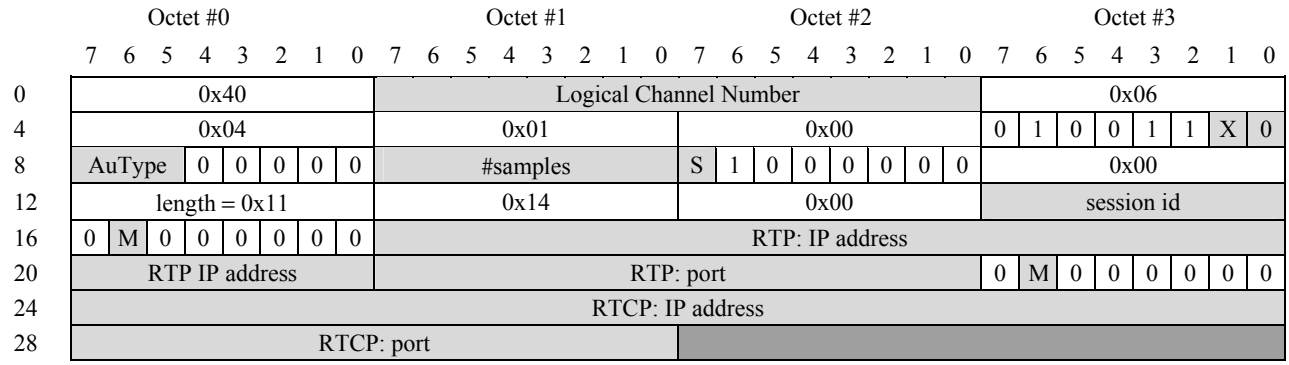
The fields have the same meaning as above. In addition, the following fields are defined:

RTP IP address/port: Target transport address for the RTP audio stream to be sent to.

RTCP IP address/port: Target transport address for RTCP sender reports to be sent to.

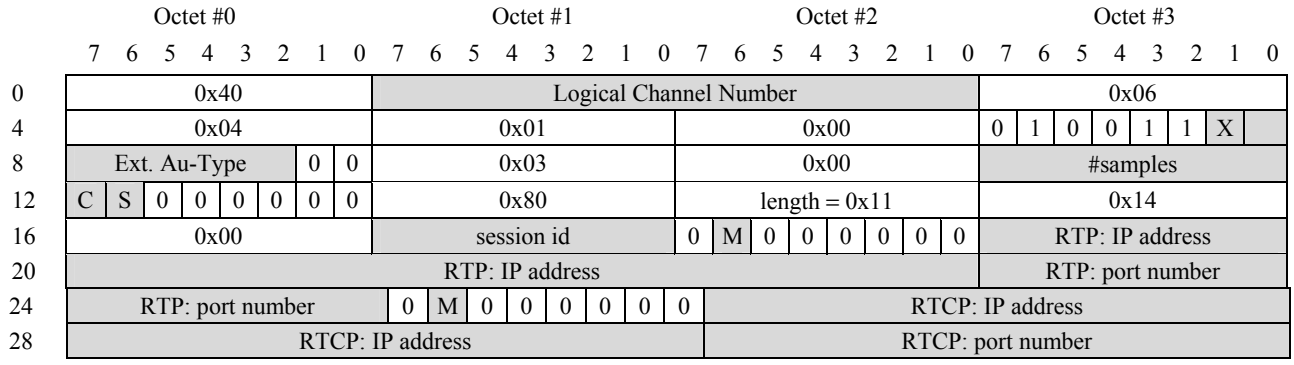
F.10.1.2.2 ITU-T Rec. G.723.1

For ITU-T Rec. G.723.1, the structure differs slightly from the above as follows:



F.10.1.2.3 GSM

For GSM, identified by bit #1 of octet #7 set to X = 1, the structure looks as follows:



Ext. Au Type: Identifies the extended (GSM) audio codec to be used as follows:

GSM Full Rate = 000 0011

GSM Half Rate = 000 0100

GSM Enhance Full Rate = 000 0101

Annex G

Text conversation and Text SET

G.1 Introduction

Standardized, character-oriented text conversation facilities are needed in all networks. When building text conversation facilities on multimedia protocols, an opportunity is created to use any combination of text, video and voice in a conversation. The initiative to standardize this combination comes from the needs of people with communication-related disabilities. The availability of the three media in a conversation offers communication opportunities over any one of the media alone. Anyone may find a commonly available, standardized text conversation addition to multimedia conversation services valuable, enhancing videotelephony to "Total Conversation".

Since H.323 is a framework, where components can be included when required, single function text terminals as well as text and voice terminals can be useful subsets of the full Total Conversation terminal. These subsets correspond to text telephones available for the PSTN.

ITU-T Rec. T.140 [G1] specifies a text conversation protocol. It is a common presentation level suitable for straightforward real-time text conversation in multimedia services and in text telephony. It is based on the ISO/IEC 10646-1 character code so as to be suitable to any language. It is introduced throughout the H-series multimedia protocols.

This specification describes how text conversation facilities are added to the H.323 multimedia environment in packet networks.

The text conversation facility is established in a data channel identified by the H.245 **OpenLogicalChannel** message. The same identification is used for opening text conversation channels in H.324. Only the protocol and procedures of the data channel to carry T.140 data differ.

Thereby, Total Conversation gets a uniform implementation across different networks. The complexity of gateways and other network components can be kept low.

G.2 Scope

The scope of this annex is to specify H.323 procedures to establish and carry text conversation sessions in real time over packet networks in the H.323 multimedia environment. It also specifies rules on the use of H.323 that enable Text Conversation Simple Endpoint Type Devices (Text SET) to be created as supersets of the Audio Simple Endpoint type devices specified in Annex F/H.323. The Text SET specification describes a device that can be used for real-time conversations in voice and text simultaneously over packet networks.

G.3 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[G1] ITU-T Recommendation T.140 (1998), *Protocol for multimedia application text conversation,* plus amendment.

[G2] HELLSTRÖM (G.): RTP Payload for Text Conversation, *RFC 2793*, *Internet Engineering Task Force*, 2000.

G.4 Definitions

This annex defines the following terms:

G.4.1 total conversation: Conversational services offering real-time communication in video, text and voice.

G.4.2 T140PDU: Protocol Data Unit from T.140 = a collection of data submitted in T.140 format for transmission.

G.5 Procedures for opening channels for T.140 text conversation

The session requirements of T.140 are reflected in the following specification for the channel setup using the H.245 Open Logical Channel Message structure in the H.323 environment.

A reliable or unreliable channel may be selected to carry the T.140 session. The unreliable channel shall always be supported. The unreliable channel may be selected for cases when the terminal is expected to participate in sessions where a reliable channel is unfavourable or impossible to use. The reliable channel is a preferred option.

- In the capabilities exchange, when using a reliable channel, specify:

  ```
  DataApplicationCapability.application = t140
  DataProtocolCapability = tcp
  ```

- In the capabilities exchange, when using an unreliable channel, specify:

  ```
  DataApplicationCapability.application = t140
  DataProtocolCapability = udp
  ```

- In the Open Logical Channel procedure, specify:

  ```
  OpenLogicalChannel.forwardLogicalChannelParameters = dataType
  DataType = data
  ```

  And select a reliable or unreliable channel for the transfer of T.140 data by specifying the `DataApplicationCapability` and the `DataProtocolCapability` as above.

The fast-start or the normal procedures may be used.

The destination node and originating node concepts of ITU-T Rec. T.140 are mapped to the two H.323 endpoints.

The T.140 user identity is an alias for the far H.323 endpoint.

G.6 Framing and buffering of T.140 data

Transmission of T.140 data shall be done according to the following specifications, different for the reliable and the unreliable channel.

G.6.1 Common considerations

T.140 data may be collected in a buffer before transmission in the channel. On low bit-rate channels, such buffering is recommended in order to reduce packet overhead. Buffering of data in 0.3-second intervals is recommended as default.

On reception, the data contents of the data channel is retrieved and used as T.140 data.

G.6.2 Usage of reliable channels

When a reliable channel is selected for T.140 transmission, TCP is used, and T.140 data is transmitted in the channel without further framing.

G.6.3 Usage of unreliable channels

When an unreliable channel is specified for the T.140 transmission, RTP is used. The details of the RTP payload format "T140" is found in [G2]. The recommended procedures described in [G2] should be used. The payload type allocation is dynamic. For the plain "T140" payload format, Payload Type 96 is used. For the payload type "RED" with redundancy, Payload Type 98 is used.

The procedures offer the possibility to include a number of already transmitted T140PDUs in the packet. This is done in order to include redundant data to reduce the risks of data loss.

The transmitting station may select a number of T.140 PDU generations to retransmit in each packet. A higher number introduces better protection against loss of text. If network conditions are not known, it is recommended to use two generations. It is recommended to use not more than six generations.

RTCP should be used to monitor packet loss, so that a decision can be made on the number of generations of redundant data to transmit.

G.7 Interaction with text conversation facilities in other devices

The information in this clause is not normative and is provided for information only, beyond the scope of this annex.

ITU-T Rec. T.140 is established as the text conversation protocol throughout a series of H-series multimedia protocols, T.120 data conferencing and for ITU-T Rec. V.18 text telephones. The data channels are specific to each environment.

When gateways to these different environments are established, the T.140 channel in the H.323 environment is mapped into the T.140 channel in the other environment. The T.140 channel data can be transparently transferred through the gateway.

When gateways to other text conversation protocols are established, the data and protocol mechanisms of that protocol shall be mapped into a T.140 text conversation channel in the gateway. Such mapping functions can be called T.140 equalizers. Gateway functions to the different text telephone systems involve T.140 equalizers.

Figure G.1 gives an overview of text conversation protocols and gateway services.

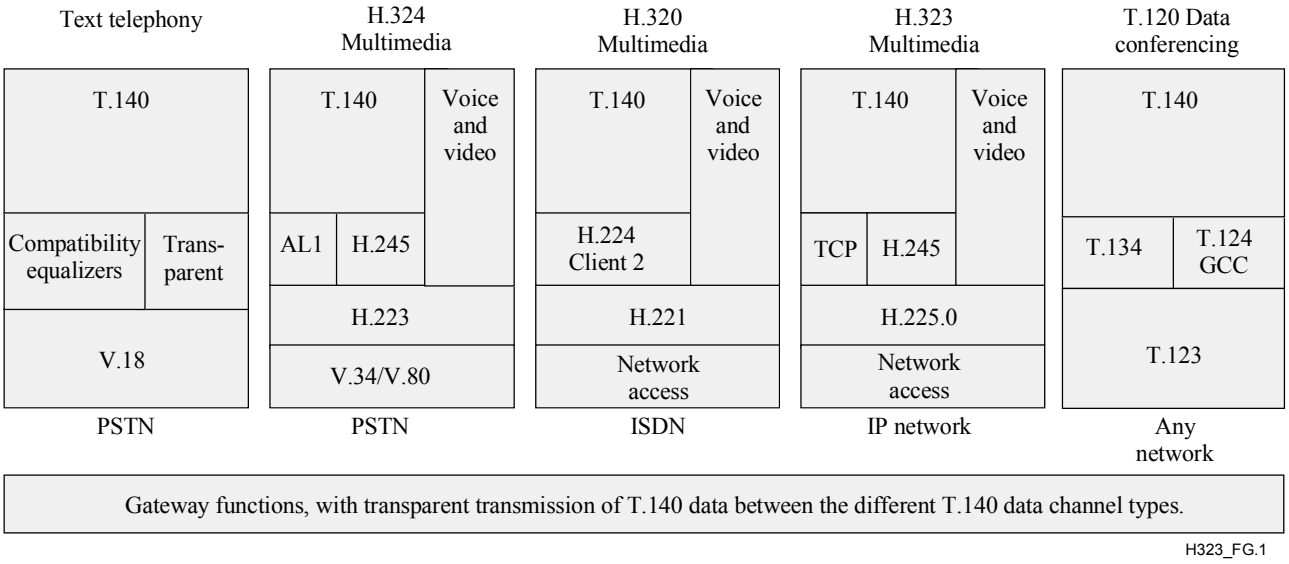


Figure G.1/H.323 – Multimedia real-time text conversation Recommendations and interworking needs

G.8 Multipoint considerations

Without further specification, three alternative options exist for H.323 endpoints with T.140 text conversation to participate in multipoint text conversations.

Alternatives:

- One separate T.140 channel is set up for each remote H.323 endpoint. The text streams can be coordinated for display through a multipoint-aware user interface, that also transmits T.140 data to all connected endpoints.
- An MCU coordinates the T.140 data stream to the H.323 endpoint to contain data from a number of remote endpoints.
- Instead of the procedures described in this annex, the T.134 application member of T.120 data conferencing is used as the channel for T.140 data. Multipoint sessions are coordinated through the T.120 concepts.

G.8.1 Situations for multipoint text conversation

In order to clarify the use of text conversation, and especially the different multipoint cases, the following examples of possible setups and applications are given without being normative.

G.8.1.1 One-to-one

The one-to-one case represents a direct conversation in text between two parties, where the text entered at one endpoint is displayed character by character or in small groups of characters as they are entered at the other end. Typical examples are situations like the traditional text telephony in PSTN and multimedia conversation applications with video, text and data used for person-to-person calls. See Figure G.2.

Anne	Eve
Hi, this is Anne. Have you heard that I will come to Paris in November?	Oh, hello Anne, I am glad you are calling! No, that was new to me. What brings you here?

Figure G.2/H.323 – Possible display of a one-to-one text call

G.8.1.2 Many-to-many

All users have write permission, forming an unmanaged conference.

The display can be arranged as specified in ITU-T Rec. T.140 with one window for each participant. See Figure G.3.

Anne	Eve
Hi, this is Anne. Have you heard that I will come to Paris in November?	Oh, hello guys! How are you Steve?
Steve	**Bill**
Hi there! This is Steve, I'm fine.	Hello Anne! I am happy that you are on the big Internet!

Figure G.3/H.323 – Possible display of an unmanaged four-to-four text session

The display of a many-to-many conference can also be ordered in one window with labels for each participant's entries (IRC style) (see Figure G.4):

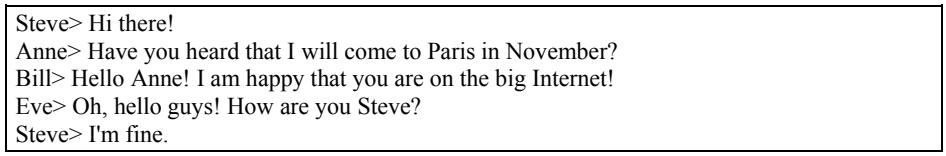


Figure G.4/H.323 – Possible display of an unmanaged four-to-four text session IRC style

G.8.1.3 One-to-many with managed right to type

One writer at a time is given the right to transmit text to many readers. The right to type may be passed to other writers, in a managed meeting.

Typical application is in distance education when the teacher normally has the right to type, but can hand it over to a participant.

G.8.1.4 One-to-many with fixed right to type

One writer types text in the session from one fixed endpoint, the other endpoints display the text in a receiving window. The right to write cannot be transferred.

Typical application is found in subtitled speeches.

The user terminals may be H.332 loosely coupled endpoints.

See Figure G.5.

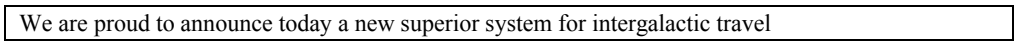


Figure G.5/H.323 – Example of one-to-many text session

G.9 Text SET: Text Conversation Simple Endpoint Type

This part of the annex specifies Text Conversation Simple Endpoint Type Devices that operate using a well-defined subset of H.323 protocols. They are well suited for IP Text Telephony applications while retaining the interoperability with regular H.323 Version 2 (1998) devices. The specification adds real-time text conversation facilities as specified in ITU-T Rec. T.140 to the simple IP-voice telephone as specified in Annex F/H.323, to form the IP-text telephone with simultaneous voice and text functionality.

G.9.1 Introduction to Text SET

The procedural and protocol details of a Simple Endpoint Type Text Telephone Device for IP networks is defined in terms of modifications and additions to the Audio SET specification found in Annex F/H.323. The device here is called Text SET.

The general SET concepts are described in Annex F/H.323. This is a set of modifications to the Audio SET specification that comprises what is needed to add text conversation functionality to the Audio SET. This annex indicates the clause numbers of the original.

G.9.2 Text SET System Functionality Overview (F.6/H.323)

In **Media capabilities**; modify:

- Data-capability mandatory; T.140.

G.9.3 Procedures for Text SET devices (F.7/H.323)

Modify the Media packetization and transport to:

- Media packetization and transport (H.225.0, RTP, TCP, T.140) – See F.7.4/H.323.

G.9.4 RAS Signalling (H.225.0 RAS – F.7.1/H.323)

As for Audio SET, but a SET H.225.0 endpoint type code booked for Text SET is used.

Bit 2 = 1 Indicates that the device has Text SET capabilities.

Bit 2 = 0 Indicates that the device has no Text SET capabilities.

NOTE – The Gatekeeper protocols must be designed so that they will allow voice-only sessions with a Text SET device.

G.9.5 Call Signalling (H.225.0 Call Control – F.7.2/H.323)

SET H.225.0 endpoint type code bit 2 is used to indicate a Text SET function.

G.9.6 Data Capability (F.7.3.3.3/H.323)

Data capability T.140 shall be specified.

DataApplicationCapability.application = t140.

G.9.7 Additional rules for usage of capabilities (F.7.3.3.9/H.323)

Audio and data capabilities shall only be signalled via the FastConnect procedure and repeated exchange of **OpenLogicalChannel** structures using the FastConnect.

Video capabilities, conference capabilities, security capabilities, and h233 encryption capabilities shall not be used.

The values of the **MultiplexCapability** table entry shall be assumed as for Audio SET with the following exceptions:

```
mediaDistributionCapability
centralizedDataTRUE
distributedDataTRUE/FALSE as appropriate, default FALSE
```

G.9.8 Logical channel signalling messages (F.7.3.4/H.323)

Add in the **OpenLogicalChannel** request.

```
OpenLogicalChannel.forwardLogicalChannelParameters.DataType.data = t140
MultiplexParameters     as appropriate for the selected reliable or
                        unreliable channel type.
```

G.9.9 Media exchange (F.7.4/H.323)

For text exchange, SET terminals shall follow the procedures specified in this annex.

G.9.10 Initiating side (F.7.6.1/H.323)

Add:

The **OpenLogicalChannel** structure should offer the same data encoding for text that were offered in the initial call.

G.9.11 Conference-unaware Text SET terminals (F.7.7.1/H.323)

Add the following functionality points:

- Merging several incoming text sessions to the Text SET device.
- Translating the transport addresses for the text stream.
- Transferring and possibly transcoding text data streams.

G.9.12 Support for loosely-coupled conferences (ITU-T Rec. H.332) (F.7.8/H.323)

A Text SET device can participate in a Loosely-coupled Conference using the H.332 procedures provided that the conference is expanded to include text, and that the channel for text transmission is selected to use an unreliable channel.

Annex J

Security for H.323 Annex F

J.1 Introduction

This annex describes security for H.323 Annex F simple endpoint types. The specified security profile is based upon H.235v2 and uses the featured baseline security profile of H.235 Annex D. The shown security profile in H.323 Annex J adopts ITU-T Rec. H.235 for the purpose of simple endpoint types and their specific security requirements. The security profile selects appropriate security features from H.235 with its rich set of options.

The described text provides an overview on the security profile; H.235v2 Annex D provides all the technical and implementations details.

Basically, a **security simple endpoint type** (**security SET**) is a SET as defined by H.323 Annex F that implements additionally certain security features of this annex.

Currently, this annex focuses only on a "secure audio SET (SASET)" and leaves any other security simple endpoint types (e.g., secure FAX SET, secure text terminal, secure Video SET, etc.) for further study.

J.2 Specification conventions

Some explanation is useful for understanding the terms used in this annex:

The annex applies the **baseline security profile** for a SASET (**secure audio simple endpoint type**). The baseline security profile provides basic security by simple means using secure password-based cryptographic techniques; the functionality provided should be implemented by each SASET. The baseline security profile may use the **voice encryption security profile** for achieving voice confidentiality if necessary. It is for further study, whether there will be other, more sophisticated security profiles for SASETs.

In order to avoid references to a trademark (RC2®), this annex actually references an "RC2-compatible" encryption algorithm.

This annex uses well-known security terms as key, key management and SET, which have different meanings in other contexts (e.g., touch key pad, Q.931/Q.932 feature key management, and Secure Electronic Transaction protocol).

J.3 Scope

This annex describes security for simple endpoint types. As shown in F.3, this currently includes:

- **Secure simple telephone terminal** (Secure Audio Simple Endpoint Type) – Defined in this annex (see J.6).

Any other security SETs are for further study.

J.4 Abbreviations

This annex uses the following abbreviations:

DES	Data Encryption Standard
GK	Gatekeeper
HMAC	Hashed Message Authentication Code
ITU	International Telecommunication Union
MAC	Message Authentication Code
RAS	Registration, Admission & Status
RTP	Real Time Protocol
SASET	Secure Audio Simple Endpoint Type
SET	Simple Endpoint Type
SHA	Secure Hash Algorithm

J.5 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

– 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.245 (2003), *Control protocol for multimedia communication*.

– IETF RFC 2268 (1998), *A Description of the RC2® Encryption Algorithm*.

J.6 Secure Audio Simple Endpoint Type (SASET)

This annex describes a baseline for **secure audio simple endpoint types (SASETs)**. An example of a SASET is a secure simple phone.

J.6.1 Assumptions

The baseline security profile mandates the GK-routed model for secure H.323 Annex F SETs. SASETs and other H.323 entities that implement this security profile (e.g., GKs) are assumed to implement the fast connect procedure.

In accordance to Annex F the baseline security profile mandates the fast connect procedure with integrated key management elements but does not support H.245 tunnelling. Thus, the baseline profile does not provide means for key update and synchronization using (tunnelled) H.245 messages. SASETs implementing only the baseline security profile but still need some key-update mechanism should hangup the call and reconnect and thereby obtain a new session key.

J.6.2 Overview

The baseline security is applicable in administered environments with symmetric keys/passwords assigned among the entities (SASETs-gatekeeper, gatekeeper-gatekeeper).

Table J.1 summarizes all the procedures defined in H.235v2 Annex D.

Table J.1/H.323 – Summary of Secure Audio Simple Endpoint Types (see H.235v2 Annex D)

Security Services	Call functions						
	RAS	H.225.0		H.245 (Note)	RTP		
Authentication	*Password HMAC-SHA1-96	*Password HMAC-SHA1-96		*Password HMAC-SHA1-96			
Non-Repudiation							
Integrity	*Password HMAC-SHA1-96	*Password HMAC-SHA1-96		*Password HMAC-SHA1-96			
Confidentiality					♦56-bit DES	♦56-bit RC2-compa-tible	♦168-bit Triple-DES
Access Control							
Key Management	*Subscription-based password assignment	*Subscrip-tion-based password assignment	♦authen-ticated Diffie-Hellman key-ex-change	♦Integrated H.235 session key management (key distribution, key update using 56-bit DES/56-bit RC2-compatible/ 168-bit Triple-DES)			

* Blue area: Password-based scheme

♦ Green area: Voice encryption security profile

NOTE – Embedded H.245 inside H.225.0 fast connect.

For authentication and integrity, the user shall use a password-based scheme (blue area in Table J.1). The password-based scheme is highly recommended for authentication due to its simplicity and ease of implementation. Hashing the fields in the H.225.0 messages is the recommended approach for integrity of the messages (also using the password scheme). SASETs realize authentication in conjunction with integrity using the same common security mechanism.

SASETs when deploying the voice encryption security profile (green area in Table J.1) shall implement 56-bit DES as the default encryption algorithm; SASETs may implement 168-bit Triple-DES while SASETs implementing exportable encryption may implement 56-bit RC2-compatible.

For voice confidentiality, the suggested scheme is encryption using RC2-compatible, DES or Triple-DES based on the business model and exportability requirement. Some environments that are offering already a certain degree of confidentiality may not require voice encryption. In this case, Diffie-Hellman key agreement and other key management procedures are not necessary as well.

Access control means are not explicitly described; they can be implemented locally upon the received information conveyed within H.235 signalling fields (ClearToken, CryptoToken).

This Recommendation does not describe procedures for subscription-based password/secret key assignment with management and administration. Such procedures may happen by means that are not part of this annex.

SASETs may use back-end services according to the procedure described in H.235v2 Appendix I.4.6.

Annex K

HTTP-based service control transport channel

K.1 Introduction

This annex describes an optional way of controlling supplementary services in an H.323 environment. By opening a separate connection conveying a service independent control protocol, new services may be developed and deployed without updates to the H.323 endpoints.

This service control channel is intended to be used for a wide range of services, some which require the use of H.450 or proxy signalling (e.g., as in Appendix III) for invocation/execution. As this channel is service independent, no specific services are defined or advocated. The data exchanged on this channel are meant to be informative (user interface) and should be followed by appropriate actions (e.g., H.450 invocations) in the call signalling plane when needed. Although some serverside applications need to support H.450 services for interworking, this annex is totally independent of the H.450.x Recommendations.

The service control channel may be utilized for both call-related and non-call-related services. It may be opened between the terminal and the network, or between two endpoints (in a call or with a call independent connection).

While several protocols might be used, this annex describes the use of the hypertext transfer protocol (HTTP) for this purpose. HTTP is open, flexible, firewall friendly and well known. Any device claiming to support Annex K shall support HTTP as a transport for service control, optionally also S-HTTP for applications requiring security. The actual service application protocol is dynamic, and is indicated using MIME types in the HTTP signalling. Example applications may include XML pages possibly including Java™ and scripts, download of tones and announcements to be played to the client, upload of Call Processing scripts from client to a gatekeeper, etc. While this annex focuses on user directed supplementary services, this service control channel could also be used for other means. It could, for example, be used for software upgrades or for pushing commercials to the clients.

Clause K.2 describes the use of H.323 for providing the HTTP connection's URL between the service provider and the client, clause K.3 shows the use of HTTP, and clause K.4 shows some examples of possible services and the corresponding signalling.

The interface between the service control plane and the call control plane on the client or the service provider is not within the scope of this annex, but could include HTML or XML tags such as mailto or H.323 URLs. See Figure K.1.

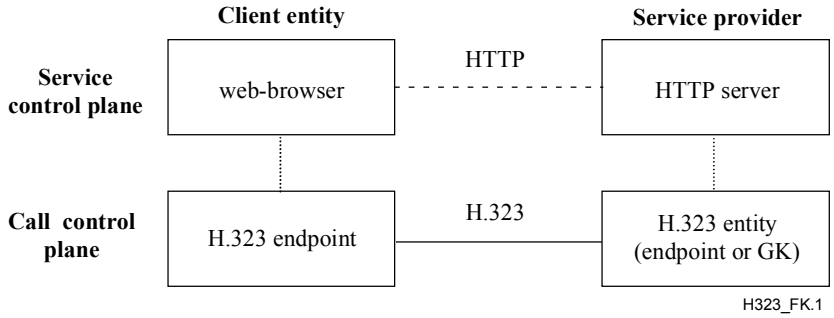


Figure K.1/H.323 – System overview for HTTP-based service control

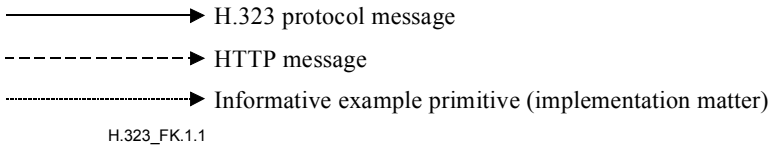
It is generally up to the provider of the URL to define and implement the control functions and services that are being presented by the given URL (standard or non-standard services may be supported). If the service control interacts with H.323 call processing, the provider of this URL should make the binding between the HTTP service and the H.323/H.450 services that are being supported by the gatekeeper or endpoint.

As the HTTP service control channel is stateless and unaware of the services in scope, it cannot take service interaction problems into account. An application that utilizes this service control channel should, however, consider this carefully.

Any sequence charts or references to H.323 signalling in this annex are informative examples for describing possible interactions with service control and call control. They do not redefine H.323 signalling rules, as most are greatly simplified for briefness.

K.1.1 Notation

The following notation is used:



HTTP and RAS messages are capitalized (HTTP:GET, RAS:ARQ), while H.225.0 call signalling messages are written with the first letters capitalized (Setup). ASN.1 codepoints in H.225.0 are written in bold (**ServiceControlAddress**).

K.2 Service control in H.323

This clause describes how H.323 messages are used for maintaining the service control sessions.

K.2.1 Service control session

A service control session is a one way relation between the client entity and the service provider, in this case an HTTP session. It is initiated from the client after the receipt of a **ServiceControlAddress** URL in H.225.0 messages. The URL may be received through two different H.323 signalling channels:

- A **ServiceControlSession** structure containing a URL is received in a message over the RAS channel. If there is no appropriate message to send, the **ServiceControlIndication** (SCI) message may be sent to the endpoint at any time.
- A **ServiceControlSession** structure containing a URL is received in a message on the H.225.0 call signalling channel.

The service control session is identified with a **sessionId**, a unique number for the signalling channel. The **sessionIds** received through RAS and call signalling may overlap, as the senders of these may not be aware of each other.

A service provider wishing to initiate a new service control session does so by sending a **ServiceControlSession** structure to the client. It contains a new **sessionId**, the URL for the service, and the reason field set to "**open**". The client may open a connection to this address and request the resource from the URL, but no acknowledgement is given from the client in the call signalling plane. If the user wishes to end the session at any time, e.g., by closing a pop-up window for the session, this is done without any notification to the provider.

If a service provider needs to notify an endpoint about new services or events relating to a previously opened session, it may do so by issuing a new **ServiceControlSession** structure on the RAS or call signalling channel (as was used in the "open" sequence). The structure shall contain the same **sessionId** as before (to reuse the same resource, e.g., screen window), a new URL to be loaded, and the reason set to "**refresh**".

If the service provider needs to terminate the session, it may send a **ServiceControlSession** structure with the same **sessionId** and reason set to "**close**". The client should, if it still has the session open, close any resources such as windows dedicated to the session.

The reason for the support of multiple sessions is that unrelated service provider nodes may use the same notification mechanisms, e.g., the call signalling channel. Service applications utilizing this annex should take care not to overuse the number of sessions, as many notifications quickly will confuse an end user. Clients supporting this annex are not required to support more than two sessions, one call related and one non-call related.

K.2.2 Non-call-related service control

To provide services relating to the registration session, and not a given call, the gatekeeper may return a **ServiceControlSession** structure containing a URL in the RCF message. The returned URL should be complete in terms of defining protocol, server and resource, i.e., <protocol>://<server-address>/<resource>. The endpoint may load this URL and display the services and service control functions as provided by the data given by this URL (e.g., a web-page with menus and links).

If the network needs to notify the endpoint about service related events, during a call or as part of the registration, it can issue a Service Control Indication (SCI) with a URL to this endpoint. To indicate that this URL relates to an already active non-call-related service control session, the **sessionId** shall be the same as previously and the **callSpecific** field shall not be present. The endpoint may then load this URL and be provided with updated services and service control functions. An endpoint that receives such a SCI shall respond with a Service Control Response (SCR) message to avoid retransmissions of the SCI from the provider. The SCR message is only an acknowledgement of the receipt of the SCI message, and not necessarily an application level response.

The Service Control Indication message may also be used to open a new session or to close the session.

If an entity other than the local gatekeeper wishes to open a call unrelated service control session towards an endpoint, this can be done by opening a call independent signalling connection towards the endpoint, and sending a Setup message with a **ServiceControlSession** structure including an URL. The **conferenceGoal** parameter shall be set to **callIndependentSupplementaryService** and the bearer capability information element of the Setup shall be set as defined for call independent connection in 7.2.2.1.2/H.225.0. Otherwise the same procedures as in K.2.2 with the **ServiceControlSession** transported in call signalling messages applies, with the absence of media on the connection.

K.2.3 Call-related service control

Two methods are provided to open service control session related to a specific call:

1) A service control session is opened between an endpoint and its gatekeeper with a URL carried in a call-related RAS message, especially for gatekeepers using direct endpoint call signalling. If the SCI message is used the **callSpecific** field of the SCI shall contain the **callIdentifier**, the **conferenceId** and the **answerCall** field as used in previous signalling for this call. A new **sessionId** shall be used. This session should not affect the call unrelated service control session as in K.2.2.

2) Service control sessions are opened between an endpoint and a gatekeeper or between two endpoints with a **ServiceControlSession** field containing a URL in the call signalling messages.

If a service provider needs to notify an endpoint about new services or events in an existing session, it may do so by means of refreshing data on an URL that has been previously loaded (e.g., applet/servlet dialogues), or it can issue a H.225.0 message (Facility or SCI) with a new URL, the reason set to "**refresh**" and the same **sessionId** as earlier for the session. An endpoint that receives such a Facility message should load this URL and render the data presented by it to the same resource (e.g., screen window) as was first used for this session.

If a service providing entity wishes to initiate a new session after the call is connected, it may also use the Facility/SCI message with a **ServiceControlSession** containing a new **sessionId**, the URL in scope and the reason set to "**open**". H.225.0 messages without the **ServiceControlSession** present does not influence the HTTP session, except Release Complete, which without a URL indicates that all sessions for this call are ended. This signalling should be seen separate for all sessions in use (non-call related, call related with SCI and in-call signalling messages).

Gatekeepers that use the HTTP service control should be careful not to interact with end-to-end service control. This is in particular the case for non-gatekeeper routed calls where the gatekeepers are unaware of the call control messages and states. To alleviate this problem, it is recommended that endpoints use separate browser-windows for the different service control sessions. Intermediate devices such as gatekeepers or MCUs utilizing this annex must be aware of the possibility for conflict with other service providing entities along the call signalling path. Messages (call signalling or other, e.g., an LCF with service control data that can be sent to the client in an ACF) may arrive towards the client with a **ServiceControlSession** using the same **sessionId** as already used between the intermediate provider and the service client. If the intermediate device decides to pass on the **ServiceControlSession**, it must be able to map the **sessionId** to a unique number for the client. Another possibility is to multiplex these two sessions into the same presentation level protocol.

To provide call-related services between different zones or domains, a terminating entity may return a **ServiceControlSession** structure containing a URL in other messages than on the call signalling channel (e.g., LCF/LRJ). It is up to the local gatekeeper to forward the **ServiceControlSession** received in corresponding messages (e.g., ACF/ARJ) toward the client. Applications needing detailed call state information, the possibility to perform actions in the call control plane or the possibility to update the session later should not use this mechanism, but rather use the call signalling channel to convey the **ServiceControlSession** structure.

K.3 Usage of HTTP

K.3.1 Non-call-related services control channel

The HTTP protocol is defined in RFC 2068. This clause provides an informative indication of how the HTTP protocol could be employed for the purpose of providing the described service control protocol.

For non-call-related services, the endpoint is provided with an URL that it could retrieve by means of the standard GET method. The data is collected and rendered according to normal procedures for an HTTP user-agent[2]. The following example (Figure K.2) illustrates the flow:

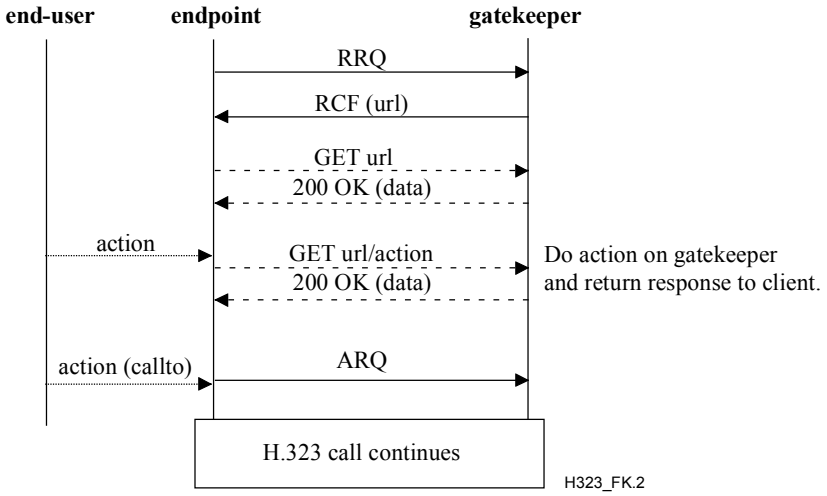


Figure K.2/H.323 – Example of non-call-related service control

K.3.2 Call-related services control channel

In order to support call-related service control a URL is conveyed in different H.225.0 messages as in K.2.3. An endpoint that supports this annex should when it receives such an URL request a standard HTTP user-agent to open and render that URL.

The HTTP user-agent should render the given URL and support style-sheets, scripts, links and images according to that defined for HTTP in RFC 2068. Actions defined and executed by the contents of this URL could be executed locally (e.g., mailto links) or remote on any linked HTTP server, e.g., being implemented or related to an endpoint or a gatekeeper. An example with the endpoint as service provider is given below (see Figure K.3), and gatekeeper service provider is in K.4 – Example 2.

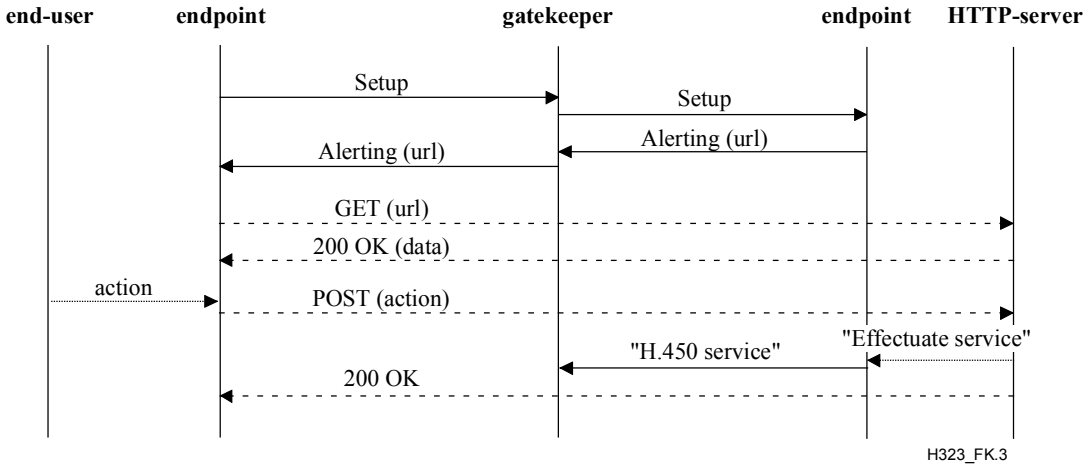


Figure K.3/H.323 – Example of call-related service control using URL in H.225.0 call signalling messages

---

[2] The term "HTTP user agent" used within this annex refers to a process that implements the client part of the HTTP protocol (normally represented with a web-browser).

1) The client sends a Setup that is routed via the gatekeeper to the endpoint representing the called party.

2) The called party may be in a state where specific call processing is programmed, e.g.:
   - Decide to reject the call by sending a Release Complete. The Release Complete could contain a URL to be displayed by an HTTP user-agent on the calling party. The URL could, for example, be a reference to the home page of the called party.
   - Decide to return a list of options for call setup options. In this case it returns Alerting with an URL that defines the options given to the calling party, for example, divert call to operator, secretary, voice-mail, email or intrusion on existing call session.

3) The calling party H.323 endpoint requests an HTTP user-agent to open the URL and the data is then rendered on the web interface of the calling party. The end-user can then dismiss the browser-window or interact with it by selecting a link/action.

4) Actions defined and executed by the contents of this URL could be executed locally (for example, mailto links) or remote on any linked HTTP server, e.g., being implemented or related to the endpoint or the gatekeeper. The remote endpoint or gatekeeper should analyse the given action and effectuate it by means of standard H.323/H.450 services. The result could, for example, be to divert the call to a voice-mail server.

K.4 Example scenarios

To illustrate the usage of the open service control a set of examples are given. These are:

- a simple example of usage of non-call-related service control;
- an example of call-related service control for gatekeeper routed calls;
- an example of call-related service control for non-gatekeeper routed calls;
- an example of non-call-related service control for script upload.

All examples here are only using one simultaneous service control channel. For simplicity, messages containing a **ServiceControlSession** structure are indicated with only the "url".

Example 1: Non-call-related service control

The example illustrates the control signals when a user registers with a gatekeeper, receives back an URL referencing a phone-book, updates the phone-book with a friends contact (alias) and then uses this updated phone book for making a call (these are not necessarily the same friends) by selecting an entry with a H.323 URL. See Figure K.4.

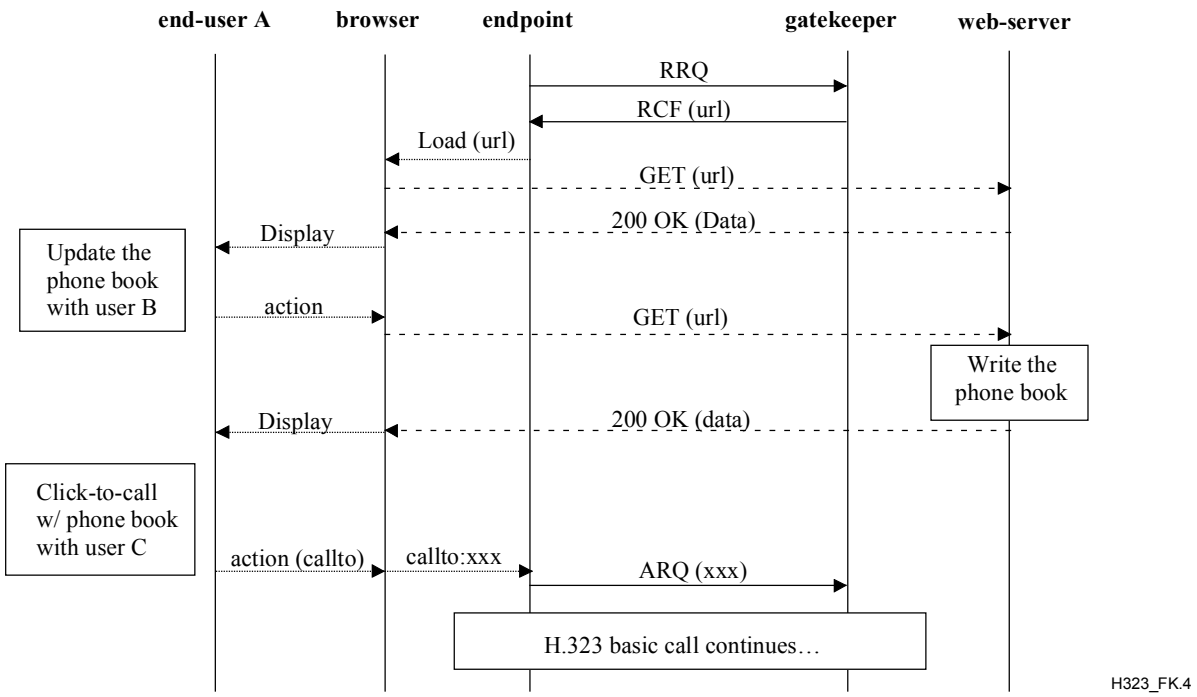


Figure K.4/H.323 – Non-call-related service control

Example 2: Call-related service control, gatekeeper-routed call

The example illustrates a variation of a "Call Waiting Service" with options for the calling party. The gatekeeper detects that the called party is busy and provides an URL to the calling party in an Alerting message (to prevent a timeout at the calling endpoint). The URL references a web page containing a set of options for the further processing of the call.

The user hears the audio alert and a web page with options is presented. The options could be divert to voice-mail, email or operator. The user selects the voice-mail and this selection is signalled to the HTTP-server that informs the endpoint about this.

The gatekeeper effectuates the diversion request as call forwarding on no reply (as Alerting have been sent) and informs the HTTP-server about the successful diversion. The HTTP server then responds to the browser with a new web-page saying, for example, that diversion was completed successfully and giving it some new options. See Figure K.5.

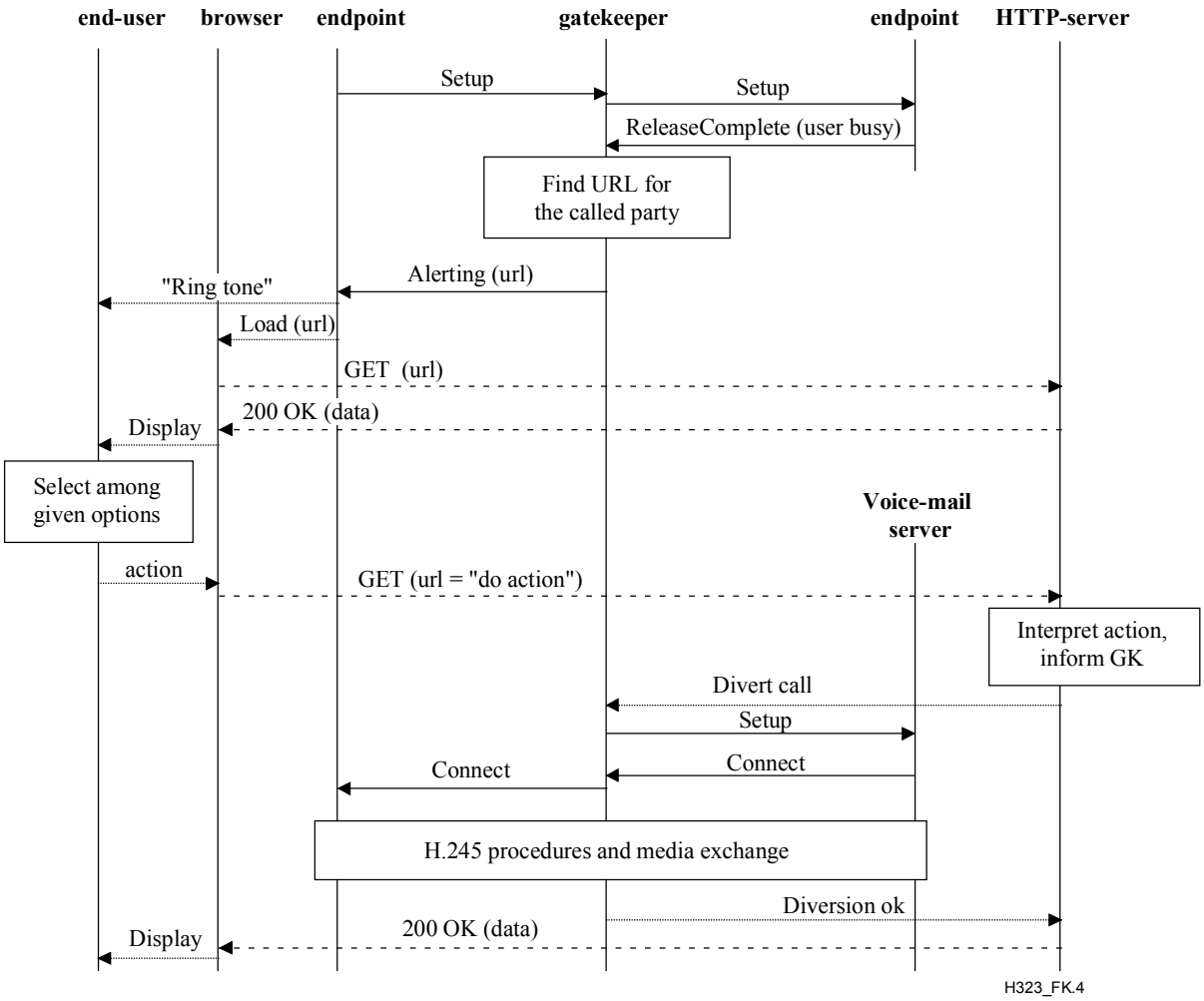


Figure K.5/H.323 – Call-related service control, gatekeeper-routed call

Example 3: Call-related-service control, non-gatekeeper-routed call

The example illustrates the same service as in Example 2, executed at the called endpoint. The called endpoint is busy in a call and returns an URL to the calling party in an Alerting message (to prevent a timeout at the calling endpoint). The URL references a web page containing a set of options for the further processing of the call.

The user hears the audio alert and a web page with options is presented. The options could be divert to voice-mail, email or operator. The user selects the voice-mail and this selection is signalled to the HTTP-server that informs the endpoint about this.

The endpoint effectuates the diversion request as call forwarding on no reply (as Alerting has been sent) and informs the HTTP-server about the successful diversion. The HTTP server then responds to the browser with a new web-page saying, for example, that diversion was completed successfully and giving it some new options. See Figure K.6.

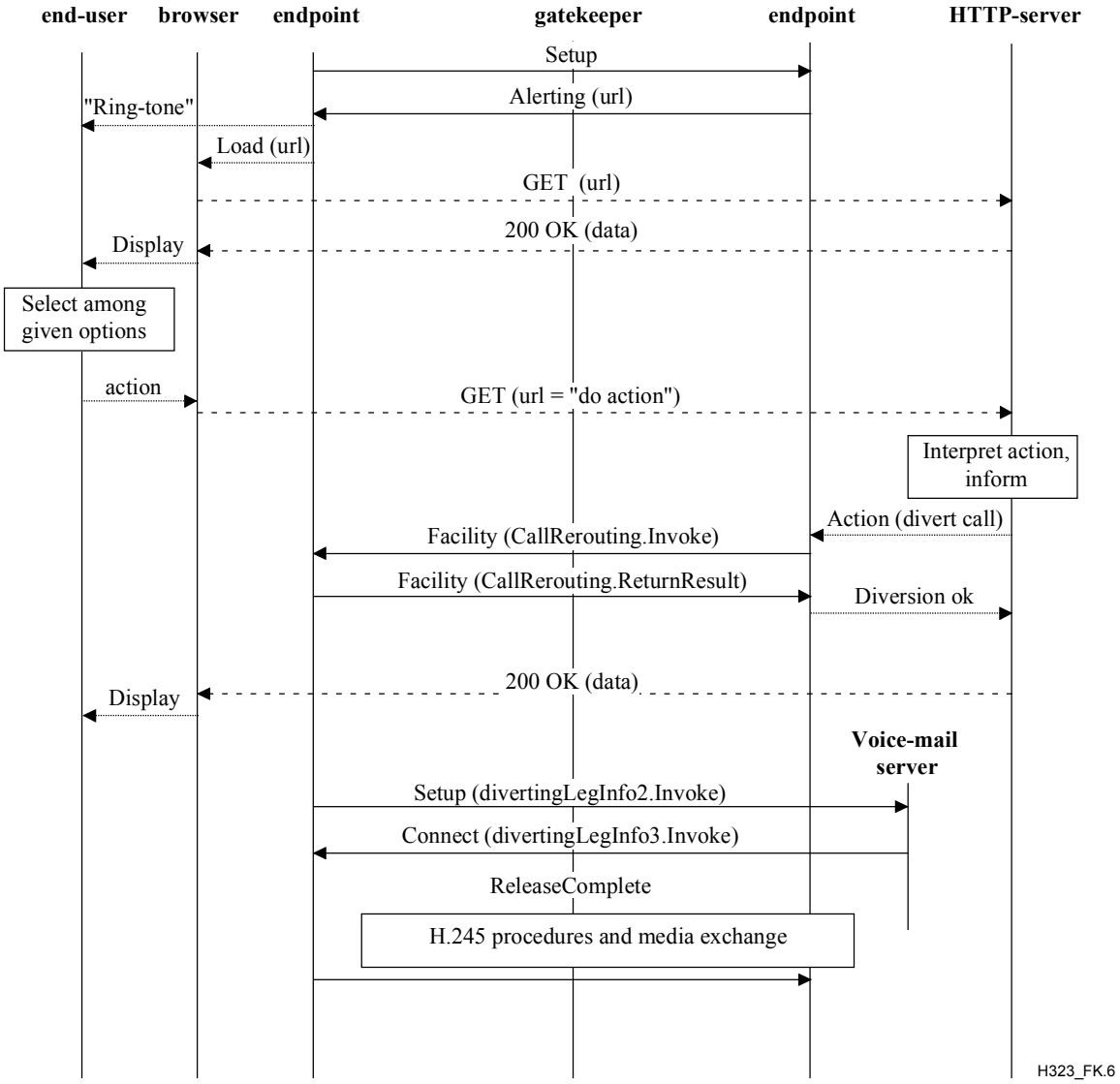


Figure K.6/H.323 – Call-related-service control, non-gatekeeper-routed call

Example 4: Non-call-related service control, script upload

Call processing scripts are also a form of service control. The example shows a terminal uploading a script after registration. The user prepares the script by a graphical builder in the endpoint or by other means, and decides to upload this to the server.

In this case the endpoint knows, when the user decides to upload the script, that it must utilize the POST scheme. The details of the script and impacts on further call signalling is dependent on the script. See Figure K.7.

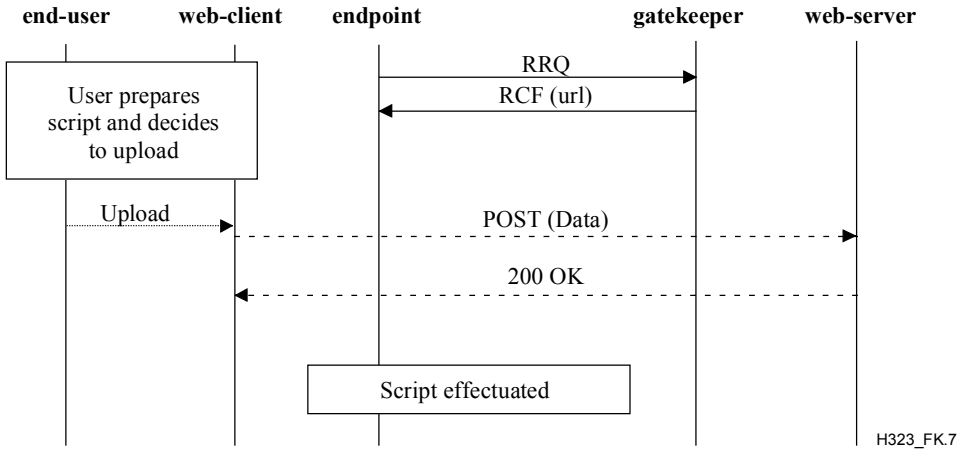


Figure K.7/H.323 – Non-call-related service control, script upload

K.5 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

K.5.1 Normative references

[H2250] ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

[URL] BERNERS-LEE (T.) *et al.*: Uniform Resource Locators (URL), *RFC 1738, Internet Engineering Task Force*, December 1994.

[HTTP] FIELDING (R.) *et al.*: Hypertext Transfer Protocol – HTTP/1.1, *RFC 2068, Internet Engineering Task Force*, January 1997.

K.5.2 Informative references

[S-HTTP] RESCORLA (T.) *et al.*: The Secure HyperText Transfer Protocol, *RFC 2660, Internet Engineering Task Force*, August 1999.

[HTML] BERNERS-LEE (T.): Hypertext Markup Language – 2.0, *RFC 1866, Internet Engineering Task Force*, November 1995.

[MIME] FREED (N.), BORENSTEIN (N.): Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, *RFC 2045, Innosoft, First Virtual*, November 1996.

Annex L

Stimulus control protocol

L.1 Scope

This annex describes stimulus signalling procedures between H.323 terminals and a Feature Server functional entity. This stimulus method allows the network service provider to implement new supplementary services for the terminals without changes in the terminal software, which results in easier maintenance. An example of such terminals is a LAN attached feature phone. A Feature Server may be colocated with the Gatekeeper.

The H.323 stimulus protocol allows services to be provided by one or more Feature Servers. For interoperability, standard H.225.0 signalling is used for basic call control, and all manipulations of media streams are done using standard H.245 or Fast Connect procedures. Mechanisms based on ITU-T Rec. H.248 are used to manipulate physical terminations such as speaker or handset.

The protocol described by this annex can support both the direct signalling model and the Gatekeeper routed model.

The typical configurations illustrated by Figures L.1 and L.2 show functional signalling entities that may be involved in a call from an H.323 stimulus terminal to another endpoint in a different H.323 zone. Figure L.1 shows the Feature Server acting as a signalling proxy for the Annex L terminal. Figure L.2 shows the Feature Server colocated with the Annex L terminal's Gatekeeper. In both cases, the Feature Server has access to the H.323 signalling, which provides it with call state information which may be useful for particular services, as well as enabling the Feature Server to affect media streams using H.245 or Fast Connect signalling.

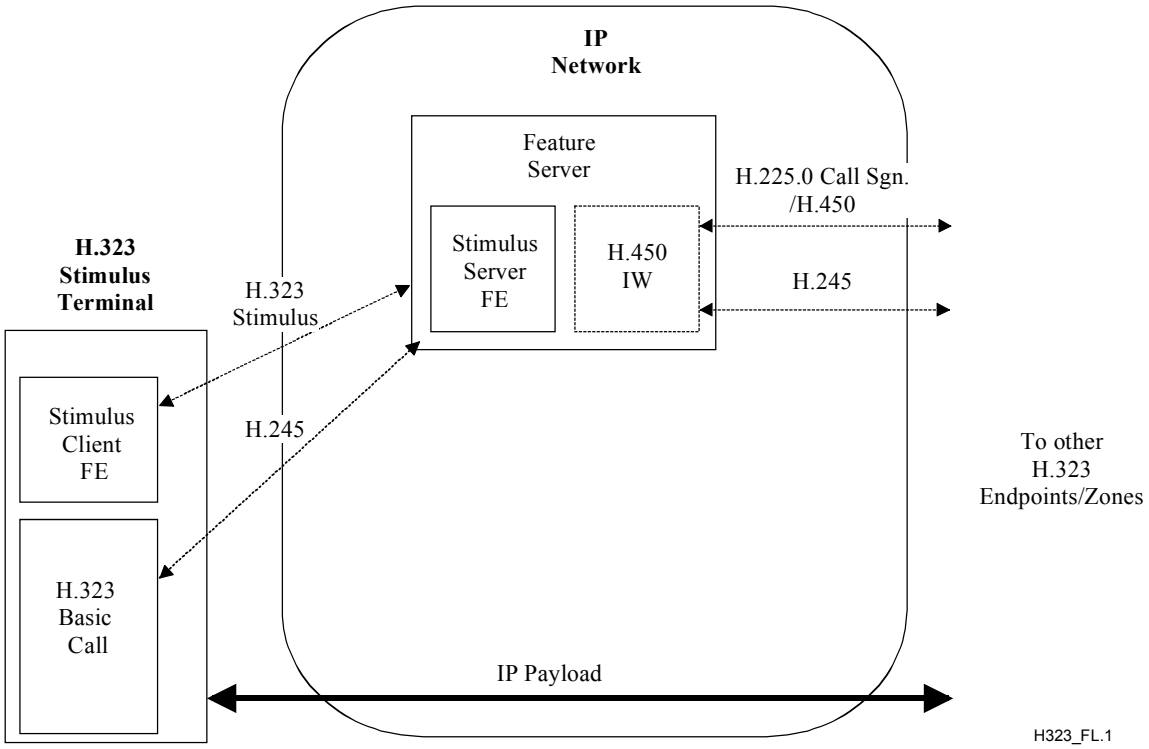


Figure L.1/H.323 – Example of Annex L in conjunction with direct signalling model

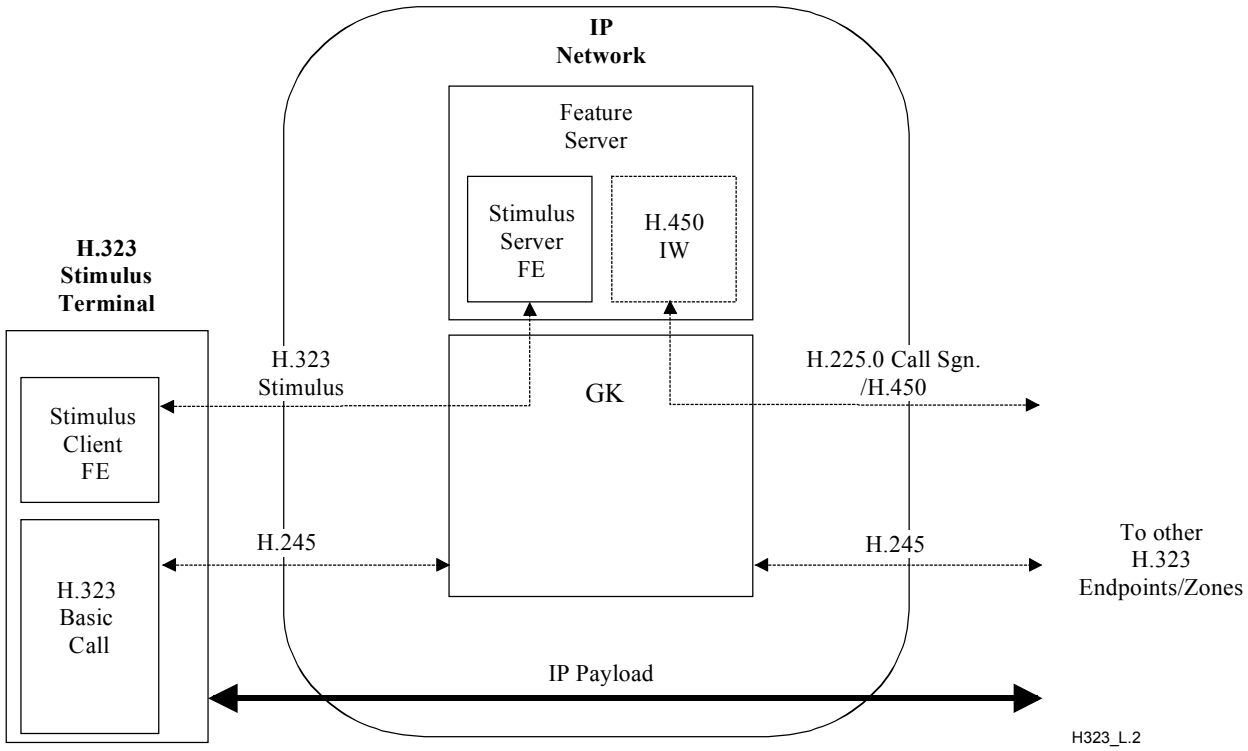


Figure L.2/H.323 – Example of Annex L in conjunction with GK-routed signalling model

L.1.1 Terminology

L.1.1.1 feature server: A Functional Entity that uses the method described in this annex to provide features to an Annex L Endpoint. A Feature Server may reside anywhere in the network. It may be colocated with a Gatekeeper, or reside on a gateway or other H.323 callable entity. A Feature Server may provide interworking between the stimulus protocol and H.450 services.

L.1.1.2 Annex L endpoint: An H.323 callable entity that can be controlled using the method described in this annex.

L.1.1.3 feature: A transaction that can affect the user interface and which may alter media streaming.

L.1.2 Relationship of H.323 stimulus to H.248

Since H.248 was developed for control of media gateways, it implies tight relationship between controller and the media gateway. Endpoints, such as telephones and residential gateways, can be included as controlled devices and treated as single line media gateways. However, these are tied to exactly one controller, which provides all connection control, features and services to the H.248 endpoints. A user can subscribe to features from one controller at a time.

This annex adopts the controller/endpoint model of H.248 for control of stimulus supplementary services, so that these procedures need to be defined only once. This annex explicitly excludes all parts of H.248 that are related to the control of media connections, which is done using standard H.245 or Fast Connect.

L.1.3 Relationship of H.323 stimulus to HTTP

Annex K allows third party control of an H.323 call based on a separate hypertext connection (using HTTP) for user interaction. There is no fixed set of capabilities for the user interface, as various types of text formats, images, and sounds will be utilized dynamically. The service provider (the HTTP server) is responsible for the mapping between HTTP events and call control actions (H.450 or other messages) for supplementary services, so the H.323 endpoint is unaware of the HTTP

application. The service provider may be associated with the local gatekeeper, or the remote endpoint, or remote gatekeeper within a call.

L.1.4 Relationship to H.450 supplementary services

Because a stimulus terminal does not perform H.450 supplementary services, the feature server or the gatekeeper is responsible for providing a proxy function for handling of the H.450 procedures over the network on behalf of the terminal.

In this case the feature server becomes an endpoint for all H.450 operations and implements all supplementary services and the state machines involved. The interaction with the user happens through the telephone user interface, which the gatekeeper is able to control via the H.323 stimulus signalling.

L.2 Introduction

The essential requirement for an H.323-based stimulus protocol is to provide a set of capabilities that allow supporting endpoints access to a potentially unlimited set of supplementary services. There are many benefits to such a protocol, such as allowing endpoints to remain relatively lightweight, and providing a degree of isolation from the effects of new feature introduction. These services themselves are typically controlled by a Gatekeeper, a proxy, or other network entity. This annex uses the term "Feature Server" to generically designate any network entity providing configuration or stimulus control of endpoints according to the protocol described.

The goals of the protocol described in this annex are:

- support for arbitrary (standard and non-standard) supplementary services;
- interoperability of these services between Feature Server and endpoint;
- backwards compatibility with endpoints using H.323 (version 2 or later).

This protocol achieves these goals by incorporating significant portions of the protocol described in ITU-T Rec. H.248. H.248 describes a purely stimulus model of endpoint control, whereas this annex must necessarily be a hybrid of both stimulus and H.323-based functional models. Annex L entities use H.248 PDUs in addition to standard H.323 messages to support this hybrid model.

This annex describes a framework that eases delivery of services into both H.323 and H.248-based systems, by allowing a high degree of commonality between an Annex L Feature Server and the components of an H.248 Media Gateway Controller (MGC) not related to media control. This framework enables the reuse of H.248-based packages in H.323-based systems, often with little or no modification. For example, suitably designed packages can allow a Feature Server to control various user interface elements of a compliant terminal, such as:

- write to a text display;
- provide hardware-independent indications to the endpoint, from which the endpoint may control its own indicators, such as message waiting or line lamps;
- receive user input such as digits, text, special keys (such as hookswitch and function keys);
- assign functions to soft keys and into an endpoint resident directory;
- request application of specific tones;
- specify tones dynamically.

Annex L terminals possess the above-listed control capabilities in common with H.248 terminals; the two types differ only in the means of managing media streams and their association with one or more calls or "contexts".

Use of the protocol described in this annex is suggested for, but is not restricted to Annex F Simple Endpoint Types.

L.3 Stimulus framework

L.3.1 Overview

Annex L terminals use the standard H.323 mechanisms for registration and signalling channel establishment. Normal H.225.0 call signalling is used for call establishment and termination. Media control may use the H.323 fast connect procedures (including repeated fastStart) or, optionally, H.245 signalling using procedures described in ITU-T Recs H.245, H.323 and its annexes. Use of these mechanisms may result in the creation of analogues to H.248 ephemeral terminations (which are not directly controllable using this annex).

Stimulus signalling capabilities of Annex L Endpoints will be specified in packages as in H.248. For example, an Annex L terminal might be described by a basic set package (for switchhook changes, etc.), a keypad package, an alerting package, a key package, and a display package. Additional packages might be included to permit modification of operational parameters and/or collect performance statistics.

As Annex L terminals are principally H.323 endpoints, H.323 procedures shall always apply and cannot be disabled by any H.248 signalling. For example, if an H.248 command results in the termination of a call, standard H.245 and H.225.0 signalling for call termination is still required.

L.3.2 Protocol signalling

The only form of signalling that all H.323 entities must support is H.225.0 Call Signalling. This is the most appropriate transport for the stimulus protocol as it allows a Feature Server to be co-located with a Gatekeeper or any other type of H.323 endpoint.

Annex L entities should support encapsulation of H.248 messages in the **StimulusControl** field, which is available in all H.225.0 call signalling messages. On every call on which it participates, an Annex L endpoint which supports H.248 encapsulation shall include a **StimulusControl** field in the first H.225.0 call signalling message that it sends to any other H.323 entity (the **StimulusControl** field may be empty).

When an endpoint registers with a Gatekeeper, the Gatekeeper may indicate an alias for the Feature Server in the **featureServerAlias** field of the RCF. When this alias is present, it should be used by an Annex L endpoint as the server destination for non-tunnelled H.248 signalling which is constrained to the functionality defined by this annex. Use of this alias address allows the Gatekeeper to associate or route the call to the Feature Server. Upon reception of a valid **featureServerAlias** in an RCF, a supporting endpoint shall immediately send an H.248 **ServiceChange** command containing the Root TerminationId to the indicated Feature Server address.

This allows two models of interaction between a Feature Server and an Annex L Endpoint:

- the Feature Server is present in the call signalling path for all H.225.0 call signalling messages for all calls originating and terminating on an Annex L Endpoint;
- a separate call signalling connection between the Annex L Endpoint and the Feature Server is established only when this feature is invoked.

L.3.3 Use of H.248

Annex L endpoints shall support the transaction level procedures of 7.2/H.248. Annex L signalling may include any of the commands defined in clause 7/H.248.

Because Annex L terminals do not use H.248 for media control, use of the following H.248 descriptors is not applicable to Annex L entities: ModemDescriptor, MuxDescriptor, StreamDescriptor, LocalControl Descriptor, Local Descriptor, Remote Descriptor, and TopologyDescriptor. These descriptors shall not be used for Annex L signalling and shall be ignored if received. Note that this annex cannot be used to explicitly address different media

streams; if an Annex L terminal supports multiple media streams (e.g., audio and video), the assignment of a termination to the call context (0xFFFFFFFD, see L.3.4, below) is implicitly assumed to refer to the stream carrying the appropriate medium.

The packages supported by the endpoint should be listed in the **supportedH248Packages** field of the RRQ when the endpoint registers with a Gatekeeper. If this field is present, but empty, a Feature Server can use an AuditCapabilities query to determine the supported packages.

L.3.4 H.225.0 encapsulation

All H.225.0 encapsulated Annex L related signalling uses a **StimulusControl** structure. Use of its fields is described in this clause. The use of Annex L by an endpoint is inferred from the presence of this structure in the first call signalling message sent by the endpoint to the feature server. If no H.248 message is encapsulated in this structure, then all of its optional contained fields may be omitted.

Encapsulated Annex L stimulus control shall be signalled using the **stimulusControl** field in the H323-UU-PDU element that is used for call signalling in H.323.

The H.248 message to be sent shall be encapsulated in the **h248Message** field in the **stimulusControl** sequence. The encapsulated message is a full MegacoMessage data type as defined in ITU-T Rec. H.248.

When an Annex L Feature Server becomes active within the context of an existing call, it may need to determine the state of that call, and/or the endpoint. This can be accomplished with the use of the H.248 AuditValue command.

The assignment of TerminationIds for physical terminations on the endpoint may be provisioned on the Feature Server and endpoint, predefined in a package, or obtained via AuditCapabilities.

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. Annex L Endpoints may support only one form of encoding, and shall use the same form of encoding for all Annex L signalling to a Feature Server. Annex L Feature Servers should support both forms of encoding; communication from a Feature Server to an endpoint shall use only the form for which the endpoint has indicated support.

For H.225.0 encapsulated Annex L signalling, the special value "ANNEX-L", defined as 0xFFFFFFFD, shall be used as the ContextId for all call-related transactions. All commands shall apply to the current H.323 call (as represented by the **callIdentifier** of the H.225.0 call signalling message encapsulating the H.248 command). Commands not related to the call represented by the encapsulating H.225.0 message shall be associated with a ContextId value of NULL, as defined in ITU-T Rec. H.248.

Encapsulated Annex L transactions shall not use ContextId values other than NULL (as defined in ITU-T Rec. H.248) or ANNEX-L (as defined above).

Certain H.248 activities may not be associated with active H.323 calls. In this case, any existing call signalling channel between the endpoint and the Feature Server may be used, and the procedures of H.248 shall be used to associate the activity with the correct H.248 objects. For these activities, H.323 call independent signalling procedures may be used. For call independent signalling, the procedure of 7.2/H.450.1 shall be used.

For H.248 activities that can be associated with an active call with the desired Feature Server in the call signalling path, any appropriate H.225.0 call signalling message may be used to communicate between Feature Server and endpoint.

L.4 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ITU-T Recommendation H.248 (2000), *Gateway control protocol*.

– ITU-T Recommendation H.248 Annex G (2000), *User interface elements and actions packages*.

– ITU-T Recommendation H.450.1 (1998), *Generic functional protocol for the support of supplementary services in H.323*.

Annex M1

Tunnelling of signalling protocols (QSIG) in H.323

M1.1 Scope

The purpose of this annex is to give guidance on how the generic tunnelling mechanism described in 10.4 can be used to tunnel QSIG over H.323 networks. Other groups such as ISO/IEC are ultimately responsible for the QSIG procedures themselves. Information on QSIG (also known as PSS1) can be found in references [M1-1] and [M1-2] below.

M1.2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[M1-1] ISO/IEC 11572:2000, *Information technology – Telecommunications and information exchange between systems – Private Integrated Services Network – Circuit mode bearer services – Inter-exchange signalling procedures and protocol.*

[M1-2] ISO/IEC 11582:2002, *Information technology – Telecommunications and information exchange between systems – Private Integrated Services Network – Generic functional protocol for the support of supplementary services – Inter-exchange signalling procedures and protocol*.

[M1-3] ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

M1.3 Endpoint procedures

Endpoints supporting tunnelling of QSIG information shall use the procedures in 10.4, with the following OBJECT IDENTIFIER used as the TunnelledProtocol:

- **{iso (1) identified-organization (3) icd-ecma (0012) private-isdn-signalling-domain (9)}**

H.225.0 messages tunnel the entire QSIG message, unchanged, starting with the Protocol discriminator field, and ending with the other information elements. The binary content of the QSIG messages is encoded as an OCTET STRING in the **H323-UU-PDU.tunnelledSignallingMessage.messageContent**. Since the binary encoding of QSIG messages is what is tunnelled, the integrity of the QSIG messages is fully preserved, including any BER encoding of ASN.1 in Facility or Notification indicator information elements.

QSIG messages can, but need not, be tunnelled in the corresponding H.225.0 messages. For example, the QSIG SETUP message can be tunnelled in a H.225.0 SETUP message, and the QSIG RELEASE COMPLETE message can be tunnelled in an H.225.0 RELEASE COMPLETE message. For other messages, it is possible that there is no corresponding H.225.0 call signalling message (e.g., in the case of a QSIG DISCONNECT message) or the corresponding message is not available because it has already been sent. In those cases, the QSIG message may be tunnelled in an H.225.0 FACILITY message. A QSIG CALL PROCEEDING message should be tunnelled in an H.225.0 FACILITY message since the H.225.0 CALL PROCEEDING message does not have end-to-end significance. Also since the NOTIFY and PROGRESS messages are optional, they might not be delivered end-to-end and should be tunnelled in a FACILITY message unless tones or announcements are provided by the called side and no Progress indicator has been sent to the calling side so far. In this case a PROGRESS message (with Progress descriptor #1 or #8) should be used to tunnel a QSIG PROGRESS message. QSIG call clearing procedures may be supported by tunnelling the QSIG DISCONNECT and RELEASE messages in the H.225.0 FACILITY message. In the special case where a tunnelled QSIG RELEASE message is interpreted as a tunnelled QSIG RELEASE COMPLETE message (this happens when a QSIG RELEASE message is received when a RELEASE COMPLETE was expected), the H.323 call may be released by the side receiving the QSIG RELEASE message by sending an H.225.0 RELEASE COMPLETE with no tunnelled QSIG message.

A single QSIG call can be tunnelled in a single H.323 call. The relationship between QSIG call references and H.225.0 call references is outside the scope of this Recommendation.

Table M1.1 is indicative only and illustrates an example of the mapping between QSIG messages and H.225.0 messages.

Table M1.1/H.323 – Mapping between QSIG messages and H.225.0 messages

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

M1.4 Tunnelling of QSIG connection oriented call independent signalling

For QSIG call independent signalling connections, no H.245 control channel and no media channels are required.

The call signalling procedures of H.225.0 may be used to establish a call independent signalling connection between the peer endpoints, as described in 10.4.

M1.5 Gatekeeper procedures

A gatekeeper participating in a call where QSIG tunnelling is used between the endpoints should pass along tunnelled QSIG messages unchanged unless it intends to terminate the tunnel. This may be the case when a gatekeeper is offering emulated QSIG services.

Annex M2

Tunnelling of signalling protocols (ISUP) in H.323

M2.1 Scope

The purpose of this annex is to give guidance on how the generic tunnelling mechanism described in 10.4 can be used to tunnel ISUP over H.323 networks. Other groups such as ITU-T are ultimately responsible for the ISUP procedures themselves. Information on ISUP can be found in references [M2-1] and [M2-2] below.

M2.2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[M2-1] ITU-T Recommendation Q.763 (1999), *Signalling System No. 7 – ISDN user part formats and codes*.

[M2-2] ITU-T Recommendation Q.764 (1999), *Signalling System No. 7 – ISDN User Part signalling procedures*.

[M2-3] ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

M2.3 Endpoint procedures

Endpoints supporting tunnelling of ISUP information shall use the procedures in 10.4. Endpoint shall identify the ISUP variant by either using the **tunnelledProtocolObjectID** or the **TunnelledProtocolAlternateIdentifier** structure. The **subIdentifier** may be used to identify the revision of the ISUP variant, e.g., "1988". See Table M2.1.

Table M2.1/H.323 – Examples of tunnelled protocols identified by tunnelledProtocolObjectID

Standard	tunnelledProtocolObjectID	subIdentifier
ITU-T Rec. Q.763 (1988)	{itu-t (0) recommendation (0) q (17) 763}	"1988"
ITU-T Rec. Q.763 (1992)	{itu-t (0) recommendation (0) q (17) 763}	"1992"

When using the **TunnelledProtocolAlternateIdentifier** structure the **protocolType** shall be set to "isup". The **protocolVariant** shall be a string identifying the ISUP specification used, e.g., a document number. See Table M2.2.

Table M2.2/H.323 – Examples of tunnelled protocols identified by TunnelledProtocolAlternateIdentifier

ISUP specification (Note)	protocolType	protocolVariant	subIdentifier
ANSI T1.113-1988	"isup"	"ANSI T1.113-1988"	"1988"
ETS 300 121	"isup"	"ETS 300 121"	"121"
ETS 300 356	"isup"	"ETS 300 356"	"356"
BELLCORE GR-317	"isup"	"BELLCORE GR-317"	"317"
JT-Q761-4 (1987-1992)	"isup"	"JT-Q761-4 (1987-1992)"	"87"
JT-Q761-4 (1993)	"isup"	"JT-Q761-4 (1993)"	"93"
NOTE – The ISUP specification may be a standard, a Recommendation or any other document specifying the ISUP protocol, e.g., an ISUP interconnection specification for a specific country.			

- **{ itu-t (0) recommendation (0) q (17) 763 }**

H.225.0 messages tunnel the entire ISUP message, unchanged, starting with the Message type code parameter, and ending with the other parameters. The binary content of the ISUP messages is encoded as an OCTET STRING in the **H323-UU-PDU.tunnelledSignallingMessage.messageContent**. Since the binary encoding of ISUP messages is what is tunnelled, the integrity of the ISUP messages is fully preserved.

For example, the ISUP IAM message can be tunnelled in a H.225.0 SETUP message, and the ISUP ANM message can be tunnelled in an H.225.0 CONNECT message. For other messages, it is possible that there is no corresponding H.225.0 message (e.g., in the case of an ISUP IDR message) or the corresponding message is not available because it has already been sent. In those cases, the ISUP message may be tunnelled in an H.225.0 FACILITY message.

A single ISUP call can be tunnelled in a single H.323 call.

Some information elements in the H.225.0 message may have been modified by the H.323 network and the gateway receiving the tunnelled ISUP message may need to override the corresponding ISUP parameters.

The **tunnellingRequired** flag shall be included in the Setup message when the ISUP required parameter in the IAM message indicates 'ISUP required'.

Table M2.3 is indicative only and illustrates an example of the mapping between ISUP messages and H.225.0 messages.

Table M2.3/H.323 – Mapping between ISUP messages and H.225.0 messages

ISUP message	H.225.0 message
IAM	SETUP
SAM	INFORMATION
CPG	CALL PROCEEDING, ALERTING, PROGRESS, NOTIFY or FACILITY
ACM	CALL PROCEEDING, ALERTING, PROGRESS, NOTIFY or FACILITY
ANM, CON	CONNECT
REL	RELEASE COMPLETE
All other messages	FACILITY

M2.4 Gatekeeper procedures

A gatekeeper participating in a call where ISUP tunnelling is used between the endpoints should pass along tunnelled ISUP messages unchanged unless it intends to terminate the ISUP tunnel. This may be the case when a gatekeeper is offering ISUP services.

A gatekeeper shall not select an endpoint that does not support ISUP when the **tunnellingRequired** flag is included the Setup message.

Annex M3

Tunnelling of DSS1 through H.323

M3.1 Scope

The purpose of this annex is to give guidance on how the generic tunnelling mechanism described in 10.4 can be used to tunnel DSS1 (Q.931) over H.323 networks. Other groups may adapt this procedure to accommodate national variants of DSS1.

M3.2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[M3-1] ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*.

[M3-2] ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.

[M3-3] ITU-T Recommendation H.450.1 (1998), *Generic functional protocol for the support of supplementary services in H.323*.

M3.3 Endpoint procedures

Endpoints supporting tunnelling of DSS1 information shall use the procedures in 10.4, with the following OBJECT IDENTIFIER used as the **TunnelledProtocol.id. tunnelledProtocolObjectID** in a H.225.0 call signalling message or in the H.225.0 RAS message:

- **{itu-t (0) recommendation (0) q (17) 931}**

  Endpoints supporting tunnelling of DSS1 information and then acting as DSS1 User entity shall use the procedures in 10.4, with the following value used as the **TunnelledProtocol.subIdentifier**:

- **"User"**

  Endpoints supporting tunnelling of DSS1 information and then acting as DSS1 network entity shall use the procedures in 10.4, with the following following value used as the **TunnelledProtocol.subIdentifier**:

- **"Network"**

  When sending a H.225.0 RAS message requesting a specific tunnelled protocol (see 10.4.2) in the **desiredTunnelledProtocol** field an endpoint has to include the OBJECT IDENTIFIER and subidentifier of the protocol it expects from the other side to ensure proper gatekeeper functionality.

  DSS1 is an asymmetrical protocol and can only be used between one user and one network entity. By using different OBJECT IDENTIFIERS for user and network entities, the H.323 endpoints can ensure that no DSS1 tunnelling takes place between two user or two network entities.

  H.225.0 messages tunnel the entire message, unchanged, starting with the Protocol discriminator field, and ending with the other information elements. The binary content of the DSS1 messages is encoded as an OCTET STRING in:

- **H323-UU-PDU.tunnelledSignallingMessage.messageContent**

  Since the binary encoding of DSS1 messages is what is tunnelled, the integrity of the DSS1 messages is fully preserved, including any BER encoding of ASN.1 in Facility or Notification indicator information elements.

  DSS1 messages can be tunnelled in the corresponding H.225.0 message or in H.225.0 FACILITY messages. For example, the DSS1 SETUP message may be tunnelled in a H.225.0 SETUP message, and the DSS1 RELEASE COMPLETE message may be tunnelled in an H.225.0 RELEASE COMPLETE message. For other messages, it is possible that the corresponding H.225.0 message may not be supported (e.g., a DSS1 CONNECT ACK message), not available because it has already been sent or not transparently transported end-to-end. In those cases, the DSS1 message shall be tunnelled in an H.225.0 FACILITY message. In particular, the H.225.0 SETUP ACKNOWLEDGE or CALL PROCEEDING messages shall not be used for tunnelling of a DSS1 message, because it may not reach the originating H.225.0 endpoint, if an intermediate Gatekeeper has already sent such a message. Instead, for tunnelling of a DSS1 SETUP ACKNOWLEDGE or CALL PROCEEDING message, first a H.225.0 SETUP ACKNOWLEDGE or CALL PROCEEDING message without a tunnelled DSS1 message shall be sent, followed by a H.225.0 FACILITY message tunnelling the DSS1 SETUP ACKNOWLEDGE or DSS1 CALL PROCEEDING message. Also, DSS1 STATUS and STATUS ENQUIRY messages shall be tunnelled in a H.225.0 FACILITY message, to ensure, that the DSS1 messages reach the H.225.0 endpoint.

  DSS1 call clearing procedures may be supported by tunnelling the DSS1 DISCONNECT and RELEASE messages in the H.225.0 FACILITY message.

A single DSS1 call may be tunnelled in a single H.323 call. The DSS1 call reference is selected by the ingress endpoint and shall be the same in all tunnelled DSS1 messages for an H.323 call. However, the DSS1 call reference value in a TDM network is unique on a peer DSS1 entity basis. In an H.323 system, there is no peer DSS1 entity basis since any H.323 call may terminate on any endpoint. To ensure uniqueness, the H.323 call reference value should be used for identifying the H.323 call only.

The DSS1 tunnelling procedures shall not be used in conjunction with the H.450.1 procedures in the same call.

Table M3.1 illustrates the relationship between tunnelled DSS1 messages and enveloping H.225.0 messages.

Table M3.1/H.323 – Relationship between tunnelled DSS1 messages and enveloping H.225.0 messages

Q.931/Q.932 message	H.225.0 message	Remark
Call establishment messages		
ALERTING	ALERTING	
CALL PROCEEDING	FACILITY	
CONNECT	CONNECT	
CONNECT ACKNOWLEDGE	FACILITY	
INFORMATION	FACILITY	Support of H.225.0 INFORMATION message is optional
PROGRESS	FACILITY	Support of H.225.0 PROGRESS message is optional
SETUP	SETUP	
SETUP ACKNOWLEDGE	FACILITY	
Call clearing messages		
DISCONNECT	FACILITY	
RELEASE	FACILITY	
RELEASE COMPLETE	RELEASE COMPLETE	
Call Information messages		
RESUME	For further study	
RESUME ACKNOWLEDGE	For further study	
RESUME REJECT	For further study	
SUSPEND	For further study	
SUSPEND ACKNOWLEDGE	For further study	
SUSPEND REJECT	For further study	
USER INFORMATION	FACILITY	
Miscellaneous messages		
CONGESTION CONTROL	FACILITY	
NOTIFY	FACILITY	Support of H.225.0 NOTIFY message is optional
STATUS	FACILITY	
STATUS ENQUIRY	FACILITY	

Table M3.1/H.323 – Relationship between tunnelled DSS1 messages and enveloping H.225.0 messages

Q.931/Q.932 message	H.225.0 message	Remark
FACILITY	FACILITY	
HOLD	FACILITY	
HOLD ACKNOWLEDGE	FACILITY	
HOLD REJECT	FACILITY	
RETRIEVE	FACILITY	
RETRIEVE ACKNOWLEDGE	FACILITY	
RETRIEVE REJECT	FACILITY	
NOTE – DSS1 messages with global call reference e.g., RESTART, RESTART ACK and STATUS may be treated by the endpoints and therefore they may not be tunnelled.		

M3.4 Tunnelling of bearer-independent DSS1 signalling

For tunnelling of the bearer-independent transport mechanisms of DSS1 as described in 6.3.2/Q.932, no H.245 control channel and no media channels are required.

The call signalling procedures of H.225.0 may be used to establish a call independent signalling connection between the peer endpoints, as described in 10.4. For details on this call independent signalling connection, see also 6.2/H.450.1.

M3.4.1 DSS1 connectionless transport

The DSS1 connectionless transport mechanism as described in 6.3.2.2/Q.932 is based on FACILITY messages using the dummy call reference value.

Each such DSS1 FACILITY message shall be transported in a separate H.225.0 connection, which shall be cleared immediately after reaching the terminating side.

In particular, a DSS1 FACILITY message shall be transported in a H.225.0 SETUP message, as described in 10.4 and in 6.2/H.450.1. The terminating side (but no intermediate Gatekeeper) shall clear this connection immediately with a H.225.0 RELEASE COMPLETE message. Additionally, the entity sending the H.225.0 SETUP message shall clear the call after receiving expiry of an appropriately chosen timer which has been started after sending the H.225.0 SETUP message.

M3.4.2 DSS1 bearer-independent connection-oriented transport

The DSS1 bearer-independent connection-oriented transport mechanism as described in 6.3.2.1/Q.932 is based on connections initiated with REGISTER messages.

Here, the following message mapping shall apply:

Q.931/Q.932 message	H.225.0 message	Remark
REGISTER	SETUP	The H.225.0 SETUP message shall be used to set up a call-independent signalling connection as described in 6.2/H.450.1. The H.225.0 SETUP message shall be acknowledged with a H.225.0 CONNECT message in order to prevent call clearing after T303 expiry.
FACILITY	FACILITY	
RELEASE COMPLETE	RELEASE COMPLETE	

M3.5 Gatekeeper procedures

A gatekeeper participating in a call where DSS1 tunnelling is used between the endpoints should pass along tunnelled DSS1 messages unchanged unless it intends to participate in the DSS1 procedures and terminate the DSS1 protocol. This may be the case when a gatekeeper is offering DSS1 services.

Annex O

Usage of URLs and DNS

O.1 Scope

This Recommendation defines a means for building multimedia communication services over an arbitrary packet-based network, including the Internet. It is useful to take advantage of such services as the Domain Name System (DNS) [O-1] and ENUM [O-9] in order to help facilitate the completion of multimedia calls, especially when using H.323 over the Internet. This Recommendation defines the procedures for using DNS to locate gatekeepers and endpoints and for resolving H.323 URL aliases. This annex also defines parameters for use with the H.323 URL.

O.2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[O-1] IETF RFC 1034 (1987), *Domain names – concepts and facilities*.

[O-2] IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.

[O-3] IETF RFC 2782 (2000), *A DNS RR for specifying the location of services (DNS SRV)*.

O.3 Informative references

Note that this is an informative material and it is not required to implement this annex.

[O-4] ITU-T Recommendation E.164 (1997), *The international public telecommunication numbering plan*.

[O-5] IETF RFC 768 (1980), *User datagram protocol*.

[O-6] IETF RFC 793 (1981), *Transmission control protocol*.

[O-7] IETF RFC 1006 (1987), *ISO transport services on top of the TCP: Version 3*.

[O-8] IETF RFC 2806 (2000), *URLs for Telephone Calls*.

[O-9] IETF RFC 2916 (2000), *E.164 number and DNS*.

[O-10] IETF RFC 2960 (2000), *Stream Control Transmission Protocol*.

O.4 H.323 URL

The H.323 Uniform Resource Locator (URL) describes a location for an H.323 entity or service reachable using standard H.323 procedures. The H.323 URL can include optional parameters that specify services and transport protocols that facilitate H.323 communications. The URL is suitable for usage within web pages, as input provided by the user, as output of an ENUM procedure, etc.

The H.323 URL has the general form of *user@hostport* where either both of the parts (i.e., *user* and *host*) or only one of the parts (i.e., *user* alone or *@host* alone*)* is present. The *user* part corresponds to an H.323 user or service name. The *host* part is a legal numeric IP address or a fully qualified domain name, thus providing means for address resolution using the DNS infrastructure.

Refer to 7.1.4 for the specific syntax of the H.323 URL.

This annex defines the H.323 URL parameters and the procedures for the H.323 URL usage.

O.5 Encoding of H.323 URL in H.323 messages

In general, procedures defined by this annex apply to an H.323 URL that is encoded with its scheme name. Processing of an URL/URI without an encoded scheme name is left for future study unless specified explicitly by this Recommendation.

An endpoint shall encode the H.323 URL with its scheme name in the **url-ID** field of **AliasAddress**.

During an address resolution procedure, a gatekeeper shall try to retrieve an H.323 URL from the **url-ID** field of **AliasAddress**. If unsuccessful, the gatekeeper should try to retrieve an H.323 URL from the **h323-ID** of **AliasAddress**. The latter is in order to support URL addressing even if an URL interface is not exposed to a user of earlier endpoint implementations. As a result the user will be able to convey the destination URL by inserting it with its scheme name manually as if it was a free format **url-ID**.

O.6 Non-H.323 URLs and URIs within the context of H.323

Non-H323 standard URL and URI schemes (such as *mailto*, *tel*, and *sip*) may be embedded into H.323 messages.

Non-H.323 URIs shall be inserted into H.323 messages in their full form (including the scheme name) in **url-ID** field of **AliasAddress** type.

An H.323 entity (such as a Gatekeeper) shall process any URI (embedded in an H.323 message) according to its syntax and semantics as pointed to by its scheme name.

O.7 H.323 URL parameters

The following table summarizes optional standard *url-parameters* of an H.323 URL. The valid parameter combinations are implied from the main text of Recommendation H.323.

Parameter	Brief description
user	Indicates that the *user* part of the H.323 URL contains a phone number.
service	Specifies the recommended type of service (i.e., one of the H.323 protocols) to be invoked first to reach the particular entity.
transport	Indicates the transport protocol to be used for the service above.

O.7.1 ABNF syntax

This annex specifies the following standard values for the **url-parameter** that is defined in 7.1.4:

```
user-parameter      =   "user=phone"
service-parameter   =   "service="("ls" | "rs" | "cs" | "be")
transport-parameter =   "transport="("udp" | "tcp" | "h323mux" | "sctp")
```

NOTE – These parameters may take on additional values in subsequent revisions of this Recommendation.

O.7.2 User parameter

Currently a single standard value for the *user* parameter is defined: *phone*.

Using *user=phone* allows to explicitly express that the user part of the H.323 URL carries a telephone number.

When encoding the *tel* URL scheme [O-8] within the H.323 URL, its scheme name (i.e., "tel:") shall be omitted and any of its used attributes (starting with ";") shall be placed in the *user* part of the H.323 URL. Note that each character occurring in the *tel* URL but is not allowed in the user part of the H.323 URL shall be escaped.

O.7.3 Service parameter

The *service-parameter* can have one of four values: *ls*, *rs*, *cs*, or *be* specifying RAS LRQ, RAS RRQ, call signalling messages of H.225.0, or the inter/intra-domain protocol defined in Annex G/H.225.0 correspondingly.

The value of *service-parameter* is that of the preferred service. In the process of connection establishment, the originating side may try using other services than the one specified by the *service-parameter*.

If *service-parameter* is absent, the H.323 entity may attempt each of the services in user-defined order. For specific guidelines, refer to O.9.

O.7.4 Transport Parameter

Signalling protocols defined in this Recommendation may be carried over different transports. Values *udp, tcp, h323mux,* and *sctp* specify UDP [O-5], TCP [O-6], Annex E/H.225.0, and SCTP [O-10] respectively. For each H.323 protocol, there are default values for both the transport protocol and the listening port (i.e., the well-known TSAP identifier) specified by ITU-T Recs H.323, H.225.0 and their corresponding Annexes. Default values may be specified by *transport-parameter* and/or *port* of an H.323 URL. Values, different from the defaults, shall be specified by the *transport-parameter* and/or *port* of an H.323 URL.

Note that inclusion of *port* parameter (including the default value) has special meaning. It is an indication to the resolving entity that the *host* points to a specific H.323 entity rather than a remote DNS domain containing H.323 SRV RRs. For details see O.9.

The value of *transport-parameter* is that of the preferred transport. In the process of connection establishment, the originating side may try using other transport protocols than the specified by the *transport-parameter*.

O.8 Usage of the H.323 URL

Currently there are two primary reasons for using an H.323 URL: to locate a callable H.323 entity and to locate a Gatekeeper with which an endpoint may register.

Additionally, ENUM [O-9] defines a system for storage of and access to mappings between E.164 numbers [O-4] and the services associated with them. The ENUM system is implemented using the Domain Name System (DNS) where the available services are represented by standard URIs [O-2].

Other uses for the H.323 URL are for further study.

O.8.1 Locating H.323 destination

When an H.323 URL is embedded in a web page or other hyperlink it means that a particular user or a service can be reached using the H.323 protocol.

Any H.323 entity may resolve the H.323 URL by utilizing DNS, including endpoints, gatekeepers, or border elements as a part of a call setup procedure defined in 8.1.

If an originating endpoint chooses to resolve the destination URL, it shall encode both the URL and the successfully resolved destination IP address (according to O.9) in the **destinationInfo** of ARQ RAS message or in the **destinationAddress** of Setup and continue the normal H.323 call setup. Otherwise, i.e., if the originating endpoint chooses not to resolve the destination URL or the DNS lookup fails, the endpoint shall encode the H.323 URL according to O.5 in the **destinationInfo** of ARQ RAS message or in the **destinationAddress** of Setup message and continue the normal H.323 call setup.

If the destination URL contains a *user* part only, a resolving H.323 entity shall logically act as if the *hostport* contained its own domain name.

Only a resolving entity that belongs to the URL domain (as specified by the *hostport*) shall interpret and process the *user* part of the H.323 URL based on its local policy. Such local policy may be based on (but is not limited to) procedures defined by H.225.0 RAS, Annex G/H.225.0, LDAP or local configuration.

If the *hostport* of the H.323 URL, is different from the DNS domain of the resolving entity it shall first perform the DNS procedure as specified in O.9. Only if the DNS procedure fails, the resolving entity may retreat to performing a different address resolution procedure based on its local policy.

O.8.2 Locating a Gatekeeper

This Recommendation defines a means of discovering a Gatekeeper via the GRQ RAS message. Generally, this entails sending out GRQ messages without any prior configuration required.

However, static provisioning of a Gatekeeper location within an endpoint is very common. It allows for better management and flexible security schemes to be implemented in the network.

Provisioning of a Gatekeeper location in terms of an H.323 URL and supporting DNS procedures for Gatekeeper discovery by the endpoints provide additional benefits. If SRV Resource Records are implemented, Gatekeeper redundancy and load-balancing schemes can be deployed transparently to the endpoints.

If an endpoint is provisioned for its Gatekeeper location with an H.323 URL in a form of "h323:*@hostport*" with no parameters it should use the *hostport* value for its Gatekeeper discovery. If an endpoint is provisioned for its Gatekeeper location with just a valid DNS domain name it is assumed that this DNS domain name is the value of the *hostport* of the H.323 URL above.

If an endpoint isn't provisioned with the H.323 URL for its Gatekeeper location but is provisioned with its own H.323 URL, it may use the *hostport* value of the endpoint's URL for the Gatekeeper discovery.

In order to discover its Gatekeeper the endpoint should use the provisioned *hostport* value and the implied **service** equals h323*rs* and **proto** equals *udp* as the inputs to the address resolution procedure defined in O.9.

If the procedure fails, the endpoint shall follow the normal Gatekeeper discovery procedures outlined in the main text of this Recommendation.

O.9 Resolving an H.323 URL to IP Address using DNS

The *host* part of the H.323 URL can specify one of the following:

– The IP numeric address of an H.323 entity.

– The DNS name of a host which is an H.323 entity.

– The remote DNS domain containing H.323 SRV RRs.

This clause defines the address resolution procedure covering these three cases.

When the *host* contains an IP numeric address, nothing needs to be resolved using DNS. The H.323 messages shall be sent directly to the specified IP address.

When the *hostport* part of the URL is present and contains a port number, it means that the *host* points to a specific H.323 entity (rather than specifying a DNS domain containing H.323 SRV RRs). This *port* value shall be assumed to be the port to which H.323 messages to be directed. Note that if the default port is to be used, the default port number shall be inserted in order to represent this case. The resolving entity shall attempt to retrieve Address Resource Record(s) ("A" RR or "AAAA" RR) for the domain name specified by the *host*. If more than a single record is retrieved, the resolving entity should select a single record based on its local policy (see also O.10.1). The H.323 messages shall be sent to the retrieved (and potentially selected) IP address and the port specified by the URL.

When the *hostport* part of the URL is present but doesn't contain a port number, it hints that the *host* most probably specifies a DNS domain containing H.323 SRV RRs. The resolving entity should attempt to locate the entity by performing a sequential SRV records retrieval within a subset of the possible H.323 services (i.e., *h323ls*, *h323rs*, *h323be*, and *h323cs*) and their corresponding possible transport protocols (i.e., *udp*, *tcp*, and *h323mux*) according to the procedure specified in O.10.4. The subset shall match the resolving entity capabilities and the purpose of the procedure (i.e., locating a Gatekeeper vs. locating a foreign border element vs. locating a destination). If the H.323 URL *service-parameter* is present or the SRV *service* (e.g., *h323rs*) is specified, the order of the SRV lookups should start based on its value. In the case that the *service-parameter* is not specified, the resolving entity may search for any or all SRV record types in any order.

For each successful retrieval, an additional DNS lookup needs to be performed to retrieve the Address Resource Records. If successful, the H.323 messages shall be sent to the retrieved and selected IP address and a default port number (corresponding to the transport protocol).

If the SRV RR retrieval procedure is not implemented or fails, the resolving entity may attempt to retrieve the Address Resource Record(s) for the domain name specified by the *hostport* even if the *port* hasn't been specified. If more than a single record is retrieved, the resolving entity should select a single record based on its local policy (see also O.10.1). If successful, the H.323 messages shall be sent to the retrieved (and potentially selected) IP address and a corresponding default port number.

O.10 Using DNS SRV Resource Records

O.10.1 Applicability

By using DNS SRV RRs (RFC 2782 [O-3]) it is possible to publish an address (i.e., an URI) for a specific service (*Service*) that can be reached over a specific protocol (*Proto*). "The SRV RR allows administrators to use several servers for a single [DNS] domain, to move services from host to host with little fuss and to designate some hosts as primary servers for a service and others as backups."

In the following clauses, this annex defines symbolic names for H.323 services and H.323 transport protocols to be registered with IANA and required for using the DNS SRV RRs. This annex also defines normative procedures for the SRV RRs usage in H.323 systems.

O.10.2 IANA registration

This specification defines the following symbolic names to be used in the *Service* field of the SRV record according to RFC 2782 [O-3].

Service	Name	Meaning
h323ls	Location Service	H.323 entity supporting H.225.0 LRQ procedure
h323rs	Registration Service	H.323 entity supporting H.225.0 RRQ procedure (i.e., a Gatekeeper that accepts registration of endpoints)
h323cs	Call Signalling	H.323 entity that performs H.225.0 call signalling
h323be	Border Element	H.323 supporting communication as defined in Annex G/H.225.0

This specification defines the following symbolic names to be used in the *Proto* field of the SRV record according to RFC 2782 [O-3].

Symbolic name	Meaning
udp	UDP as defined by RFC 768 "User datagram protocol" [O-5]
Tcp	TPKT [O-7] over TCP [O-6] in accordance with Appendix IV/H.225.0
sctp	SCTP as defined by RFC 2960 [O-10]
h323mux	As defined by Annex E, "Framework and wire-protocol for multiplexed call signalling transport."

O.10.3 SRV RR population

As defined in RFC 2782 [O-3], the DNS type code of SRV RR is 33 and its format is as follows:

_Service._Proto.Name TTL Class SRV Priority Weight Port Target

All the fields shall be populated in accordance with RFC 2782.

Service and *Proto* shall have one of the symbolic names defined above. *Port* shall have a value of a listening port on the H.323 host, which is defined by a *Target*.

If different forms of H.323 access (i.e., combinations of *Service* and *Proto*) are available for the DNS domain, all of them shall be published each using a separate SRV record.

Priority and *Weight* fields shall be used to express services local preferences' policy.

O.10.4 SRV RR retrieval and processing

This procedure does not define processing priority among H.323 *Service*s or among H.323 *Proto*s.

This procedure takes as an input a specific H.323 *Service* value and a specific *Proto* value only. Lookup in a form of _service.* is not allowed.

If no SRV records are retrieved, the procedure fails.

When the retrieved SRV records are locally processed, the selection algorithm based on *Priority* and described in RFC 2782 shall be followed. The selection algorithm based on *Weight* and described in RFC 2782 should be followed. The Priority and Weight values shall not be compared across different H.323 *Services* or different H.323 *Protos*.

The output of the process is an ordered list of SRV RRs (with or without corresponding address RR potentially provided in the Additional Data section of SRV RR).

O.10.5 Example 1

This example shows a fragment of a DNS zone or DNS domain file for **example.com**. All H.323 servers are listening on well-known TSAPs. There are two Gatekeepers installed in the domain. The **local-gatekeeper** provides registration services and can be "discovered" by its local endpoints. From the outside, the H.323 services can be reached through the external-gatekeeper by looking up the call signalling services of the domain. In addition, the external-gatekeeper would resolve its endpoint addresses by answering to the LRQ requests from the outside of its domain.

The functional separation between the two gatekeepers could be logical only and be useful in "NATted" environments where the two gatekeepers would represent local and external IP addressing.

```
$ORIGIN example.com.
_h323rs._udp            SRV 0 1 2517 local-gatekeeper.example.com.
_h323ls._udp            SRV 0 1 2517 external-gatekeeper.example.com.
_h323cs._tcp            SRV 0 1 1720 external-gatekeeper.example.com.
local-gatekeeper        A   172.30.79.11
external-gatekeeper     A   172.30.79.12
; NO H.323 access over H.323 Annex E is supported
*._h323mux              SRV  0 0 0 .
; NO other services are supported (including H.323 Border Element)
*._tcp                  SRV  0 0 0 .
*._udp                  SRV  0 0 0 .
```

O.10.6 Example 2

This example shows a fragment of a DNS zone or DNS domain file for **example.com**. All H.323 servers are listening on well-known TSAPs. The H.323 service is provided through both a Border Element and Gatekeepers. There is no priority defined or assumed between the Border Element and the Gatekeepers. It is a matter of application. For example, voice-only high quality service is provided through the Border Element while H.323 videoconferencing is provided through the Gatekeepers.

An H.323 voice phone residing in a domain would have the following URL: **h323:my-alias@example.com;service=be**. In this case a lookup for **_h323be._udp** will be performed first and succeed. Note that a lookup for **_h323cs._tcp** is allowed as well.

A videoconferencing service, provided by an H.323 MCU located in a zone of either a **main-gatekeeper** or **secondary-gatekeeper** would be published as **h323:conference-alias@example.com;service=cs**. This is due to the fact that the SRV records matching **_h323cs._tcp** will be retrieved, based on the *service-parameter*. Further, using the **Weight** field, actual access to the **main-gatekeeper** is three quarters that of the **secondary-gatekeeper** if either of the Gatekeepers is up and running.

```
$ORIGIN example.com.
_h323be._udp            SRV 0 1 2099 border-element.example.com.
_h323cs._tcp            SRV 0 1 1720 secondary-gatekeeper.example.com.
_h323cs._tcp            SRV 0 3 1720 main-gatekeeper.example.com.
border-element          A   172.30.79.10
main-gatekeeper         A   172.30.79.11
secondary-gatekeeper    A   172.30.79.12
; NO H.323 access over H.323 Annex E is supported
*._h323mux              SRV  0 0 0 .
; NO other services are supported (including H.323 Location Service)
*._tcp                  SRV  0 0 0 .
*._udp                  SRV  0 0 0 .
```

Annex P

Transfer of modem signals over H.323

P.1 Scope

The purpose of this annex is to describe the procedures for transferring modem signalling over an H.323-based network. The signalling procedures describe the use of H.245 (including Fast Connect and Extended Fast Connect), State Signalling Events (SSEs) to signal endpoint capabilities, to open and close logical channels, and to signal state changes. H.323 entities that support the carriage of modem signals of IP networks shall provide that functionality in accordance with this annex.

P.2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[P-1] ITU-T Recommendation V.150.1 (2003), *Modem-over IP networks: Procedures for the end-to-end connection of V-series DCEs*.

[P-2] ITU-T Recommendation H.460.6 (2002), *Extended Fast Connect feature*.

[P-3] IETF RFC 2198 (1997), *RTP Payload for Redundant Audio Data*.

P.3 Definitions

This annex defines the following terms:

P.3.1 modem over IP: The transport of modem signals over an IP network as described in ITU-T Rec. V.150.1.

P.3.2 modem relay: The transportation of modem data across a packet network using modem termination at the network access points.

P.3.3 state signalling event: RTP-encoded event messages that coordinate switching between different media states as defined in Annex C/V.150.1.

P.3.4 voice band data: The transport of modem signals over an audio channel of a packet network with the encoding appropriate for modem signals.

P.4 Abbreviations

This annex uses the following abbreviations:

FEC Forward Error Correction

MoIP Modem over IP

MPS Multiple Payload Stream

OLC Open Logical Channel

RTP Real Time Protocol

SPRT Simple Packet Relay Transport

SSE State Signalling Event

VBD Voice Band Data

P.5 Introduction

H.323 systems have been widely deployed throughout the world for the carriage of audio, video and data traffic over packet-based networks, including IP networks. One of the applications of H.323 has been for transiting audio calls between two disjoint circuit-switched networks or two points on the same circuit-switched network. In such an application, the call is originated in a circuit-switched network and delivered to an H.323 Gateway. This Gateway then establishes communication with a remote Gateway that, in turn, delivers the call to a circuit-switched network.

In such applications, it is desirable to also allow calls between Gateways to be data calls, rather than audio or video calls only. Annex D introduced the signalling procedures necessary to facilitate the transport of facsimile data over an IP-based network between Gateways and other devices. The focus of this annex is to specify the procedures for transporting modem data over an IP-based network between two Gateways.

Figure P.1 graphically depicts H.323 Gateways that carry modem signals between modems over an IP network.

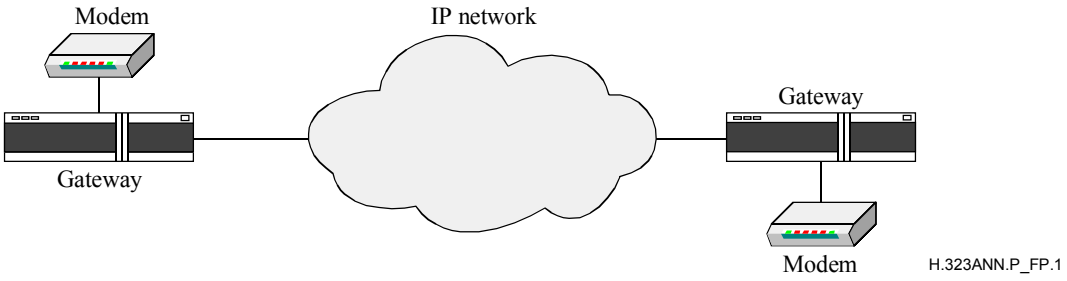


Figure P.1/H.323 – Typical modem over IP application

ITU-T Rec. V.150.1 defines the general procedures for carrying modem signals over IP-based networks between two Gateways and should be read in conjunction with this annex. Whereas ITU-T Rec. V.150.1 does not define the carriage of modem signals within the context of any particular call control protocol, this annex defines the procedures that are necessary and particular to this Recommendation.

Unless explicitly stated otherwise, references to H.323 endpoints throughout the remainder of this annex are to endpoints that are capable of carrying modem signals over an IP network.

P.6 Capability advertisement

As usual, endpoints advertise their capabilities using the **terminalCapabilitySet** message in H.245. The capabilities that are of particular importance and required for the application of modem over IP are the MoIP and SSE data application capabilities (defined in Annex F/V.150.1), RTP audio telephony events (see B.2.2.13/H.245), and the **vbd** audio capability. The **fecCapability** and/or **redundancyEncodingCapability** capabilities may be supported in order to improve the reliability of Voice Band Data (VBD) channel.

Endpoints shall also advertise support for the **multiplePayloadStream** (MPS) in the capability set transmitted to the other endpoint.

The MoIP and SSE capability definitions are in Annex F/V.150.1.

In accordance with ITU-T Rec. V.150.1, the list of codecs supported as VBD codecs shall include G.711 μ-law and A-law. Further, H.323 endpoints shall support G.711 for VBD at 64 kbit/s and, optionally, at 56 kbit/s.

P.7 Call establishment

Because of the time-critical nature of modem signalling, the calling endpoint should use the Fast Connect procedure to offer one or more channels suitable for MoIP operation. The calling endpoint should also include its terminal capabilities in the **parallelH245Control** field in order to facilitate the rapid negotiation of MoIP-related channels.

Likewise, the called endpoint should return a Fast Connect reply as quickly as possible. This reply may be an acceptance or a refusal of the offered channels. Additionally, if the **parallelH245Control** field is present in the Setup message, the called endpoint should acknowledge the receipt of that information as specified in 8.2.4.

In the case that media cannot be negotiated through Fast Connect for any reason, the endpoints shall begin logical channel signalling via the H.245 Control Channel as quickly as possible. Again, the implementor is cautioned about the time critical nature of MoIP and is encouraged to initiate this signalling well before the transmission of the Connect message.

P.8 Logical channel signalling

There are five types of streams of particular importance to an endpoint that supports MoIP. Those streams are: an audio stream, a VBD stream, RTP audio telephony events, State Signalling Events (SSEs), and an SPRT stream. An endpoint shall logically group streams necessary for MoIP together via an MPS channel. One exception to this requirement is that the SPRT stream may be signalled as a separate channel and associated to the audio/VBD channel using the **associatedSessionID** field.

Within the context of a MoIP session, the MPS channel that contains the audio and/or VBD streams and other streams for MoIP should be considered the primary audio session. As such, the H.245 **sessionID** should be set to 1. However, endpoints are at liberty to use dynamic session ID values, as prescribed by ITU-T Rec. H.245.

While there are no strict limitations on the number of streams that may be contained within any MPS channel, the MPS channel used for MoIP shall contain no more than one audio stream, no more than one VBD stream, no more than one SSE stream, and no more than one SPRT stream. If the SPRT stream is opened as a separate channel, the MPS channel shall not also include an SPRT stream. In addition, there may be one payload type for normal audio, one for the VBD stream, one for the SSE stream, and one for the SPRT stream. It is possible that more than four payload types may be utilized for those four streams. For example, if the VBD stream is protected with Forward Error Correction (FEC), and if those FEC packets are contained within a Redundancy Encoding packet, there may be not just one payload type value for the VBD stream, but three: one used in the RTP header to signify that the packet contains a redundantly encoded payload, one for the primary payload (the VBD data), and one for the FEC data carried as the secondary encoding.

To optionally protect the VBD stream, an endpoint may utilize forward error correction and/or redundancy encoding. A stream that utilizes forward error correction shall be signalled via the **fec** field of the **DataType** structure within the **MultiplePayloadStreamElement** structure. A stream that utilizes redundancy encoding shall be signalled via the **redundancyEncoding** field in the **DataType** structure within the **MultiplePayloadStreamElement** structure.

To illustrate the usage of the MPS for MoIP, consider an OLC that has a G.729 audio stream, a G.711 A-law VBD stream that is protected with redundancy encoding, an SSE stream, and an SPRT stream. The **OpenLogicalChannel** would essentially have a composition similar to that shown in this abbreviated example:

```
{
     forwardLogicalChannelNumber 1,
     forwardLogicalChannelParameters {
          dataType : multiplePayloadStream {
               element {
                    dataType : audioData : g729 2
               },
               element {
                    dataType : redundancyEncoding {
                         primary {
                              dataType : audioData : vbd : g711Alaw64k 160
                         },
                         secondary {
                              dataType : audioData : vbd : g711Alaw64k 160
                              payloadType 97 -- The PT for the redundant encoding
                         }
                    },
                    payloadType 101 -- The PT for the RFC 2198 packet
               },
               element {
                    dataType : data {
                         application : genericDataCapability {
                              -- SSE capability
                              capabilityIdentifier : standard {
                                   itu-t(0) recommendation(0) v(22) 150 sse(1)
                              },
                              nonCollapsing {
                                   {
                                        parameterIdentifier : standard 0,
                                        parameterValue : octetString "3,5"
                                             -- A comma-separated string
                                             -- of supported events (this string
                                             -- illustration of syntax and is not
                                             -- necessarily an appropriate list)
                                   },
                                   {
                                        parameterIdentifier : standard 1,
                                        parameterValue : logical
                                   }
                              }
                         }
                    },
                    payloadType 102     -- The PT for the SSE packets
               },
               element {
                    dataType : data {
                         application : genericDataCapability {
                              -- MoIP capability
                              capabilityIdentifier : standard {
                                   itu-t(0) recommendation(0) v(22) 150 moip(0)
                                   major-version-one(1) minor-version-one(1)
                              },
                              nonCollapsingRaw '0000'H
                                   -- This value shown is only presented
                                   -- for illustration and is not
                                   -- a valid value
                         }
                    },
```

```
                    payloadType 103     -- The PT for the MoIP packets
                }
            }
        },
        multiplexParameters : h2250LogicalChannelParameters {
            sessionID 1
        }
}
```

P.8.1 Extended fast connect

Extended Fast Connect [P-2] should be used in order to reconfigure logical channels, as it is much faster than exchanging a series of H.245 messages. If an endpoint needs to transition from audio operation to MoIP operation and does not presently have an open channel suitable for usage with MoIP, it should first attempt to reconfigure channels using Extended Fast Connect.

Extended Fast Connect should also be the first preference for logical channel signalling even when existing channels do support MoIP. For example, if the endpoint wishes to exchange the G.729 audio codec within an MPS with the G.723.1 audio codec, it should attempt to reconfigure the logical channels via Extended Fast Connect, as opposed to using H.245 signalling.

P.8.2 H.245 signalling

H.245 logical channel signalling via the H.245 Control Channel may be employed to configure or reconfigure media streams as necessary. MoIP-capable endpoints shall support H.245 Tunnelling when there is a need to utilize an H.245 Control Channel. However, it should be understood that support for H.245 Tunnelling does not guarantee that it will be utilized and that a separate connection may be necessary, though discouraged.

While signalling the opening of new channels is typically not an issue for H.323 endpoints, the possibility does exist that two endpoints may attempt to open channels independently which results in an incompatible configuration. To resolve such issues, the master shall reject the OLC proposals from the slave device with the reason **masterSlaveConflict**. The master should then send a **RequestMode** message to the slave device to propose a compatible mode of operation.

If an endpoint determines that it is necessary to switch modes of operation, for example, to switch from audio-only mode to a mode that supports MoIP, the endpoint shall send a **RequestMode** message to the other endpoint. For example, assume that two endpoints open G.729 audio in each direction and, then one endpoint determines a need to change the mode of operation from audio to MoIP. The endpoint shall send a **RequestMode** message over the H.245 Control Channel indicating the desired mode of operation. The receiving endpoint shall respond with an acknowledgement or a rejection message, as appropriate, but it should make every effort to accept the requested mode of operation. The endpoints should exchange messages in a manner similar to that shown in Figure P.2. As much as possible, messages should be exchanged in parallel to reduce mode transition delays.

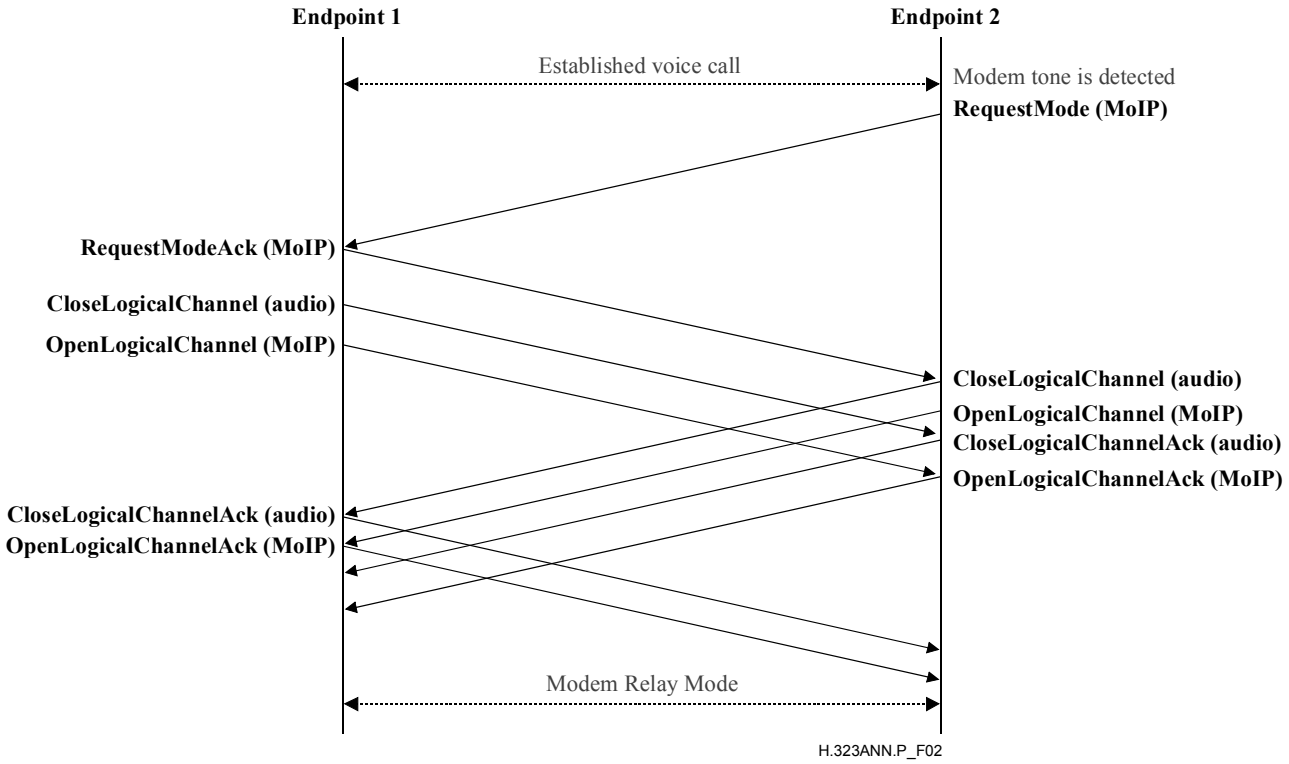


Figure P.2/H.323 – Successful switch between audio and MoIP modes

Annex Q

Far-end camera control and H.281/H.224

Q.1 Scope

The purpose of this annex is to provide far-end camera control protocol based on H.281/H.224. It also permits an H.323 endpoint to run any H.224 application using the IP/UDP/RTP/H.224 protocol defined in this annex.

Q.2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[Q-1] ITU-T Recommendation H.224 (2000), *A real time control protocol for simplex applications using the H.221 LSD/HSD/MLP channels*.

[Q-2] ITU-T Recommendation H.281 (1994), *A far end camera control protocol for videoconferences using H.224*.

[Q-3] ITU-T Recommendation T.140 (1998), *Protocol for multimedia application text conversation*.

Q.3 Introduction

The protocol described in this annex may be used to support far-end camera control (FECC) in this Recommendation using the stack IP/UDP/RTP/H.224/H.281. This protocol supports both point-to-point and multipoint scenarios.

This method may be used as a "simple" FECC scheme when the more sophisticated features of H.282/H.283 are not needed.

This method shall be used for FECC thru H.320-H.323 and H.324-H.323 gateways when the H.320 or H.324 endpoints do not support ITU-T Rec. H.282.

The requirements given below apply only in the case that the protocol described in this annex has been selected, using the normal procedures of ITU-T Rec. H.245.

It is allowed to run any H.224 application using the IP/UDP/RTP/H.224 protocol defined in this annex. The only other currently standardized H.224 application is ITU-T Rec. T.140.

Q.4 Far-end camera control protocol

Q.4.1 General

This protocol is based on ITU-T Rec. H.281 running over ITU-T Rec. H.224 in an RTP/UDP channel.

On IP transport networks, the H.224 protocol octet structure shall be the same as Figure 2/H.224 except that the HDLC bit stuffing, HDLC flags and HDLC Frame Check Sequence shall be omitted. The entire remaining content of each frame shall be placed in a single RTP packet.

References in ITU-T Rec. H.224 to the LSD channel of ITU-T Rec. H.221 shall be interpreted as referring to the H.224 logical channel as described in this annex. The maximum transmission time requirements of ITU-T Rec. H.224 shall be met, with the H.224 logical channel considered as operating at 4800 bit/s, regardless of the actual bit rate of the channel.

This protocol shall run over RTP in a unidirectional unreliable H.245 logical channel. The RTP payload value shall be dynamic. The payload descriptor field of H.245 **RTPPayloadType** shall use the H.224 Object ID.

Terminal numbering according to the procedures in ITU-T Rec. H.243 shall be used in order to support the data link layer in multipoint. The MCU/Terminal address pair <M><T> shall be used to uniquely identify each terminal in a conference. The special destination address of <0><0> shall be used as the broadcast address. The special source address <0><0> shall indicate that the sender does not know its address. An address with the terminal number set to 0 indicates the MC. For example, <n><0> indicates MC number n.

In a point-to-point call, when only two terminals are involved, then the terminals do not have an <M><T> address. In this case, the <M><T> source and destination addresses shall be always <0><0>.

In a centralized conference, an H.224 channel shall be opened between each terminal and the MC. When a terminal sends an H.224 packet, the MC shall forward the packet to the destination terminal by either retransmitting each packet to all other connected terminals or, by selectively retransmitting each packet only toward the destination terminal. The decision which method to use is up to the MCU manufacturer.

In a decentralized multicast conference, each terminal shall multicast the FECC packet to all other terminals. The MC is not involved in forwarding the packets. Terminal numbers per ITU-T Rec. H.243 shall be used to identify the source and destination terminals.

In decentralized multi-unicast conferences, each terminal shall use a separate logical channel to each far-end terminal to which it wants to send H.224 packets.

Q.4.2 H.320 to H.323 gateways

H.320-H.323 gateways shall insert and remove HDLC flags, HDLC bitstuffing, and HDLC Frame Check Sequence(s) as appropriate in each direction, so that the bitstream on the H.320 side shall conform with ITU-T Rec. H.224, and the bitstream on the H.323 side shall conform with the paragraphs above.

Q.4.3 H.324 to H.323 gateways

H.324-H.323 gateways shall insert and remove HDLC flags, HDLC octet-stuffing, and HDLC Frame Check Sequence(s) as appropriate in each direction, so that the bitstream on the H.324 side shall conform with the use of ITU-T Rec. H.224 as described in ITU-T Rec. H.324, and the bitstream on the H.323 side shall conform with the clauses above.

Q.4.4 H.245 signalling

The use of this protocol shall be signalled by the **GenericCapability** part of the **DataApplicationCapability** sequence in H.245. The Generic Capability for H.224, described in ITU-T Rec. H.224, shall be used. This shall be placed in the **receiveAndTransmitDataApplicationCapability** part of the **Capability** choice.

This protocol shall not be signalled in the **receiveDataApplicationCapability** or **transmitDataApplicationCapability** parts of the **Capability** choice.

Q.5 RTP header information

The following fields shall be filled in the RTP header:

V:	2
M:	0 NA
PT:	The same number sent in the OLC dynamicRTPPayloadType field
Sequence number:	Filled, incremented by one for each RTP packet sent
Timestamp:	Filled with 8 kHz clock rate
SSRC:	Filled with the synchronization source

Annex R

Robustness methods for H.323 entities

R.1 Introduction and scope

This annex specifies methods that can be used by H.323 entities to implement robustness or tolerance to a specified set of faults. Methods for recovery of call signalling (ITU-T Rec. H.225.0) and call control signalling (ITU-T Rec. H.245) channels are specified. RAS (ITU-T Rec. H.225.0) does not involve a connection and recovery, involving registering with an alternate gatekeeper, is covered elsewhere and so not specified in this annex. Recovery of Annex G service relationships is for further study.

H.323 calls require the cooperation of two or more H.323 entities. Call state information is distributed among the various entities involved in the call. Call signalling may depend on persistent connections between some of the entities involved. If any entity fails and does not have a backup peer, it may not be possible to establish new calls. If any entity involved in an active call fails and the entity does not have a backup peer or that peer does not have a method of retrieving sufficient call state information, it may not be possible to continue with the call. This Recommendation

provides some support for building robust systems but the mechanisms are distributed throughout this annex and few, if any, procedures for using them are given.

This annex describes two alternative methods consisting of sets of mechanisms along with procedures for using them to build systems that can recover from a significant set of specified failures. One method is more appropriate for small-scale systems, uses simpler entities and does not recover as much call state information. The other method is appropriate for large-scale systems and can recover as much state information as desired but requires more complex entities. The two methods share several mechanisms and can be used concurrently in different parts of a system.

R.2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[R-1] ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems.*

[R-2] ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*.

[R-3] ITU-T Recommendation X.680 (2002), *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

R.3 Definitions

In addition to terms defined in ITU-T Rec. H.323, the following terms are used:

R.3.1 backup entity or backup peer: A peer of an entity that can take over the entity's functions if the entity fails.

R.3.2 peer entities: Two entities of the same type in an H.323 system, e.g., two gatekeepers. The two entities may cooperate in a call (e.g., originating and terminating gatekeepers in gatekeeper-routed call signalling) or one may provide backup for the other.

R.3.3 robustness methods: Procedures and mechanisms that allow recovery from the failure of one or more H.323 entities. The extent of recovery varies between different robustness methods and may include preservation of active calls in a stable state or only the ability to place new calls. Methods described in this annex are usually able to preserve active calls.

R.3.4 signalling neighbour: Other entities to which a specific entity has direct call signalling or call control signalling connections for a specific call. For example, a gatekeeper using the gatekeeper-routing model may have a direct call signalling connection for a specific call to a gateway and to another gatekeeper. These two other entities would be the gatekeeper's signalling neighbours for that call.

R.3.5 stable calls: A call is considered stable or in a stable state after Connect has been sent or received and media channels in both directions are established (though H.245 or fast connect procedures). A call becomes unstable when Release Complete has been sent or received. Certain Facility commands used to change call signalling connections may also cause a call to be considered unstable. This version of this Recommendation offers methods to preserve only stable calls during recovery.

R.3.6 tandem entities: Two (or more) peer entities, where all but one act as backup entities for one active entity.

R.3.7 virtual entity: Two (or more) closely-coupled peer entities that collectively appear as a single entity to the rest of an H.323 system while providing fault recovery.

R.4 Abbreviations

This annex uses the following abbreviations:

CRV Call Reference Value

GK Gatekeeper

GW Gateway

RAS Registration, Admission and Status

SCTP Stream Control Transmission Protocol (IETF RFC 2960) (Informative use only)

SDL Specification and Description Language

TCP Transmission Control Protocol

UDP User Datagram Protocol

R.5 Overview of the two methods

Two robustness methods are offered by this version of this annex.

The problem we are trying to solve is to recover from a failed H.323 entity. The goal is to preserve as many active calls as possible. As a minimum we wish to preserve all calls in a "stable" state. Calls not yet fully connected or in the process of tear-down may be lost. It is also a goal to preserve most relevant billing information, such as call start time, stop time, etc., even if maintained in the failed entity (e.g., routing gatekeeper).

It is assumed that the failed entity has one or more designated backup entities, although the small-scale solution may allow recovery when the failed entity returns to service quickly. Two basic problems must be solved to recover signalling for active calls:

1) Redirecting/re-establishing signalling to the backup entity.
2) The backup entity must recover sufficient call state information that had resided in the failed entity.

The two methods are primarily distinguished by the method of recovering state information about the active calls and the amount of information recovered.

R.5.1 Method A: State recovery from neighbours

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

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

R.5.1.1 Partial method A

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

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

R.5.2 Method B: State recovery from a shared repository

The second architecture depends on a fault-tolerant pseudo-entity. This may be implemented by:

1) Using a fault-tolerant platform/OS.
2) A pool of non-fault-tolerant entities that share call state information through shared-memory, shared-disks, or through messages. The mechanism for sharing is not specified in this Recommendation.

The real entities in this fault-tolerant pseudo-entity must share sufficient state information with its peer entities to allow recovery of the desired call-state without any assistance from its signalling neighbours. This Recommendation will define the minimum information entities that must be shared. Any additional information that is desired to be recoverable can be shared. We note that method B will require that all entities in the pool constituting the pseudo-entity be from the same vendor since the sharing mechanism is not standard. The group would suggest one or two possible solutions and will consider recommending a standard sharing mechanism in H.323 versions beyond version 4.

More details of this architecture will be given below.

R.5.3 Comparison

Each of these two architectures has advantages, which makes the choice less than obvious. Some of the issues are listed below.

The Recovery from Neighbour approach:

1) allows simpler entities;
2) adds less overhead before a failure (still need keepAlive messages in some cases),

but:

1) requires more changes to H.323 messages;
2) makes recovery somewhat slower (due to the Status and StatusInquiry messages);
3) is non-scalable; only suitable for small-scale systems.

The Shared Repository approach:

1) hides most of the recovery process from H.323 and so requires fewer changes to existing messages;
2) makes recovery faster;
3) allows future use of state-maintenance protocols that might be implemented below the H.323 application layer. (See Informative Note 2 in R.13.);
4) can support recovery of billing information and other desired state information,

but:

1) adds significant overhead to all signalling (before failure);
2) requires more complex entities or pseudo-entities.

R.6 Common mechanisms

The two methods share several common mechanisms.

R.6.1 Detection of TCP based connection lost

In case of Network failure, the first "automatic" attempt would be on the IP routing protocol level. If it does not succeed, the TCP failure will be reported to both sides (entity and its signalling neighbour, e.g., gatekeeper and endpoint). Either a network failure or failure of the signalling neighbour will appear as a TCP failure.

When the call was set up, it was determined whether the entity's neighbour supported robustness procedures.

In the case where one of the sides does not support the defined Robustness Procedure, it is suggested to release the call because of TCP connection failure.

On the Endpoint side, in case both sides support the Robustness Procedure, it is suggested to maintain a reasonable timeout for the robustness procedure to be initiated by the other side. This timeout is necessary in order to address a potential Network connectivity problem. After the timeout is expired, internal resources (consumed by the call) should be released.

R.6.2 Protocol failure handling

For entities that use this annex, if a protocol failure occurs in an H.245 Control Channel and both signalling neighbours support robustness, the channel and all associated logical channels are **not** closed (contrary to 8.6). Recovery procedures of this annex are instead attempted.

R.6.3 Detecting failure – KeepAlive

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

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

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

KeepAlive messages, especially at the H.323-level, can add significant signalling overhead. But note that only Method A with TCP connections uses these KeepAlives and Method A is for the small-scale case where the number of connections per entity is low. To minimize the overhead, the

use of TCP should be avoided. StatusInquiry/Status keepAlives are **not** needed in our large-scale solution.

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

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

R.6.4 Transport address and re-established connections

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

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

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

R.6.4.1 Establishment of a new TCP connection

An entity that detects loss of a call signalling channel with a signalling neighbour shall try to re-establish the channel using the backup transport address. Alternatively, the entity detecting failure may attempt to probe its original signalling neighbour using methods outside the scope of this Recommendation (e.g., ping) and, if it believes the original signalling neighbour maybe usable, it may try to re-establish the channel to the original signalling neighbour before trying the backup transport address. Implementers choosing this option should be aware that attempting to establish a TCP connection to a non-responding entity may cause significant delays.

The re-established call signalling channel will assume the state of the previous – not behave as a new channel (it will **not** begin with Setup). See further detail below for ensuring synchronization of state between signalling neighbours.

INFORMATIVE – An alternative is to use SCTP for transport rather than TCP. SCTP channels are associated with a list of alternate transport addresses that can be used as needed to maintain the channel with no intervention by the application layer. Note that more information about using SCTP is given in Informative Note 2 in R.13.

R.6.4.2 Association between the call and the new TCP connection

The association between the Call and the new TCP connection (in the endpoint side) shall be done by retrieving the callIdentifier value from the messages received on the new TCP connection.

R.6.4.3 An old TCP connection closure

After the new connection is opened, there might be two TCP connections opened, belonging to the same call on the side that did not fail. There are two options in this case:

1) The TCP connection was lost after the SETUP message was sent (and received). In this case the side that did not fail shall identify the situation and close the connection. This should be accomplished by detecting an identical callIdentifier for both connections.

2) The TCP connection was lost before the first message was transferred.

In that case, the side that did not fail has no way to find the relation between the first (old) and the second (new) TCP connections. This may be solved by a procedure that will enable the receiving side to close a connection if it is opened for a while and no message was received on it within a pre-defined timeout. (This procedure is not described in this annex.)

R.6.5 Support for extended status

To enhance interoperability between the two methods, all entities supporting robustness shall support the extended Status message including the fastStart field. This will allow an entity with a shared-repository to cooperate with a neighbour requiring Status for state recovery.

R.7 Method A: State recovery from neighbours

R.7.1 Introduction

Currently ITU-T Recs H.323 and H.225.0 do not explicitly define procedures for connection failure detection and recovery. The purpose of this method is to introduce a procedure for:

- detection of TCP based connection failure;
- synchronization between the two sides of the connection in terms of Call State;
- definition of recommended behaviour on each side in order to renew the Call Signalling connection and proceed with the call as normal in each State of the Call.

The main motivation to sustain a call (when a connection is lost) is in situations where a gatekeeper, that handles a big number of calls, fails due to hardware or software problem. In this case a control can be transferred to a standby gatekeeper (this gatekeeper may hold all the information about the calls by means of some common database). The procedure defined and presented in this annex addresses this case of Gatekeeper failure and enables the managed calls to proceed without any interruption.

This procedure does not address all the aspects of TCP based connection failure and recovery in other possible cases and topologies. Nevertheless, it is possible to address additional cases in a similar manner in the future.

R.7.2 Scope

This proposal addresses TCP based connections only (H.225.0 call signalling and H.245 call control channels). UDP (RAS) channels will not be discussed since their failure situations are already covered using the retries mechanism defined for UDP channels.

R.7.3 The robustness procedure

After a failure the H.323 Entity shall re-establish the Call Signalling connection and shall send both STATUS INQUIRY and STATUS messages to the other H.323 Entity. The other H.323 Entity shall respond with a STATUS messages, thus reaching a state where both sides are aware of the Call

State of the other side. If the receiving entity is unaware of the call, it shall respond with a STATUS message with CallState IE set to NULL. The Call Signalling connection should be established to one of the entries in **backupCallSignalAddresses** in the order of preference defined by the order of elements in **backupCallSignalAddresses** structure.

In the event that both entities simultaneously initiate Call Signalling connection, the entity with the numerically smaller value of TransportAddress used from **backupCallSignalAddresses** shall close the TCP connection it opened and use the connection opened by the other endpoint. For purposes of comparing the numeric values of TransportAddress from **backupCallSignalAddresses**, each octet of the address shall be individually compared beginning with the first octet of the OCTET STRING and continuing through the OCTET STRING left to right until unequal numeric octet values are found. Comparison shall first be performed on the network-layer address element of the TransportAddress from **backupCallSignalAddresses**, and, if found to be equal, then on the transport (port) address element. See Figure R.1.

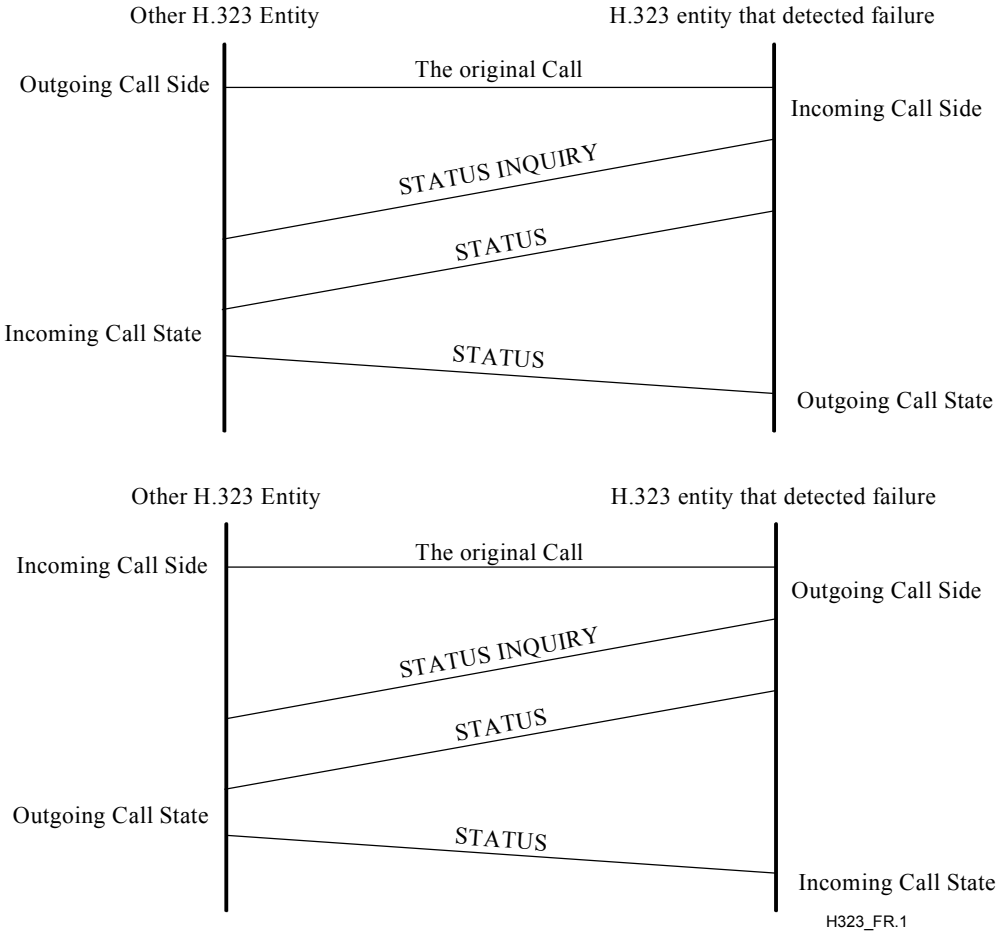


Figure R.1/H.323 – Robustness procedure

Any previous connections that might be still open for the call shall be closed, this applies both to the Call Signalling connection and the Call Control connection.

To facilitate synchronization of the logical channels state, the new fields **IncludeFastStart** in STATUS INQUIRY and **RobustnessFastStart** in STATUS message may be used. Sender of the STATUS message should include the **RobustnessFastStart** field containing the currently active receive and transmit channels with the receive addresses for media and media control streams. Sender of the STATUS INQUIRY message may request inclusion of the **RobustnessFastStart** field in the STATUS message by setting **IncludeFastStart** to TRUE.

If an intermediate entity needs to synchronize the logical channel state it should send the STATUS INQUIRY message to one of the call legs, should wait to STATUS message with fast start field, should issue the STATUS and STATUS INQUIRY message to the other leg of the call, should wait for STATUS message with the second leg logical channel information and the should send the STATUS message to the first leg of the call.

This procedure is used to synchronize the states of the logical channels that were opened through both fast start procedure and H.245 logical channel establishment procedure.

In situations where the call before failure has not reached the active state, the call should be dropped.

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

R.7.4 SDL for Method A state machine

See Figure R.2.

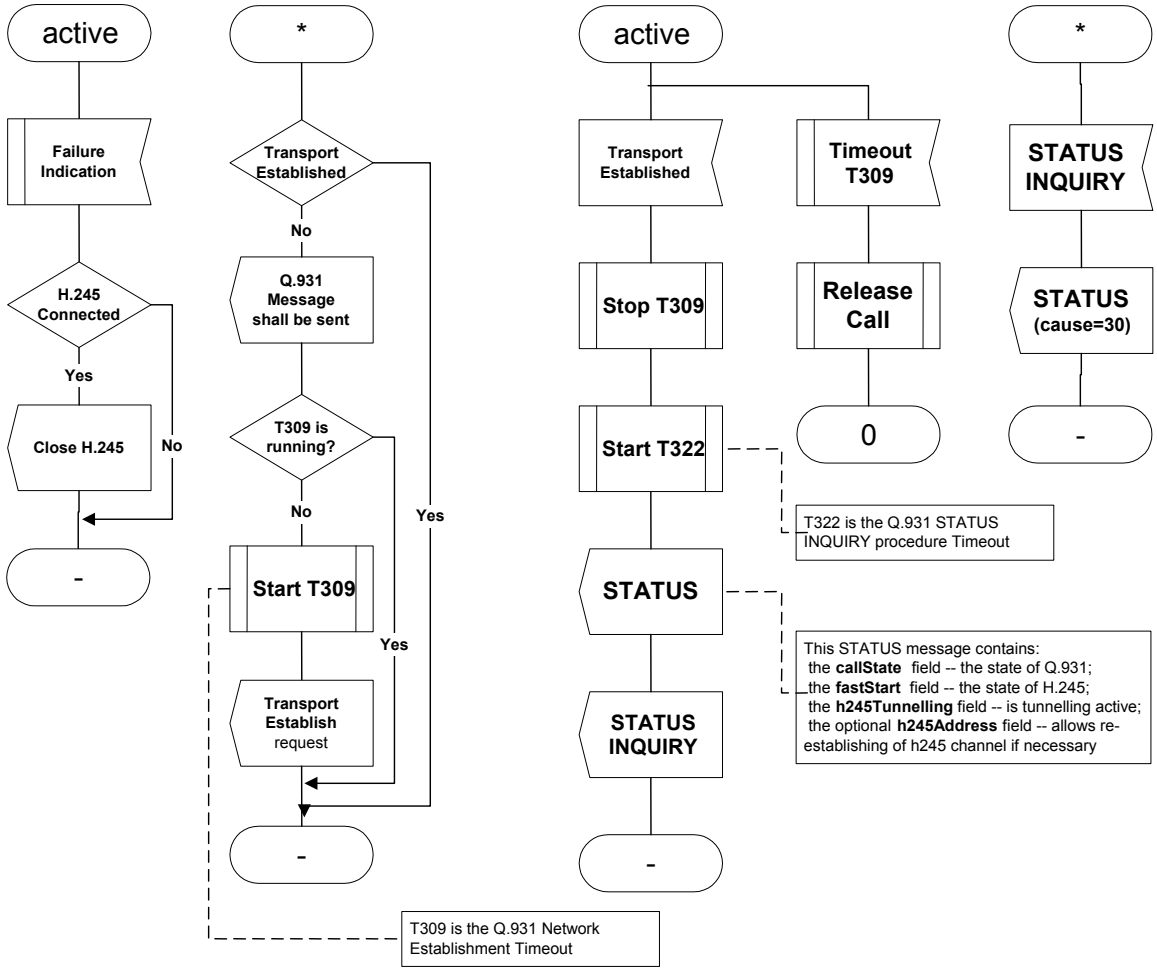


Figure R.2/H.323 – Method A state machine ***(sheet 1 of 2)***

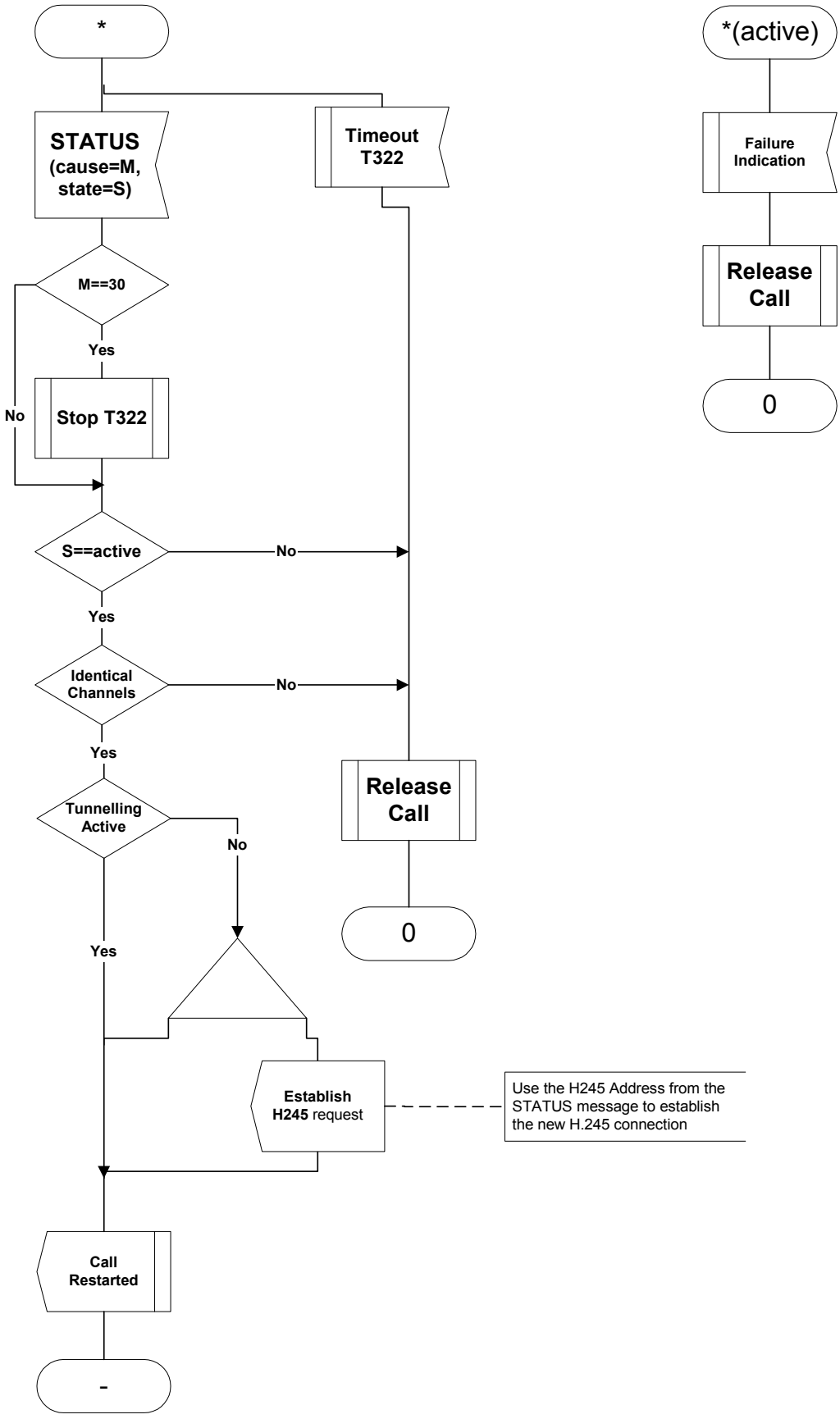


Figure R.2/H.323 – Method A state machine ***(sheet 2 of 2)***

R.8 Method B: State recovery from a shared repository

This method depends on a fault-tolerant entity or pseudo-entity and (if the backup entity requires a different signalling address) a mechanism to re-establish call signalling to the backup. There are several ways this can be done. The fault-tolerant mechanism will not be standardized in this version of this Recommendation but we will suggest some solutions. We may recommend standardizing the solution in a future version of this Recommendation. There are some emerging IETF protocols that may help solve this problem but these are not yet in a state that could be referenced by H.323v4 (November, 2000).

R.8.1 Fault-tolerant platform

One solution is to implement the robust entity on a fault-tolerant platform that uses hardware and OS support. Such a solution would make state recovery completely transparent to H.323. If the platform also maintains a constant transport address then it is a fault-tolerant virtual entity, the signalling channel will not fail and no application level procedures are needed. If the transport address changes, then the mechanism of this clause will be needed.

R.8.2 Fault-tolerant cluster

Another solution is to establish a cluster (two or more) of non-fault-tolerant tandem entities, that collectively behave as a fault-tolerant pseudo-entity. The entities of the cluster would arrange to share specified call state information sufficient to allow a peer to take over in the event of the failure of the active entity. Solutions can include:

1) active/spare ("1+1");
2) single spare shared by several active entities (spare sharing state information with each active entity that it might substitute for) ("N+1");
3) and others configurations.

Although state information is shared, allowing the cluster to appear like a fault-tolerant virtual entity, it will not be able to maintain a constant call signalling transport address and so must use one of the mechanisms of R.8.3 to re-establish the call signalling channel.

One key problem for the cluster model is how to share state. State information must be synchronized at key times in the call, to which the system can safely fall back. We will call these times *checkpoints*. This Recommendation specifies the checkpoints and the minimal data items that must be shared. We do not suggest a standard solution for sharing in this version of this Recommendation but in Informative Note 2 in R.13, we will discuss several solutions to illustrate the practicality of this model.

R.8.3 Call signalling connection re-establishment

Sharing of backup signalling addresses is the same as with Method A. Re-establishment of call signalling connections is similar but has differences since the backup entity has sufficient information to re-establish the connection on the second side rather than wait for the other neighbour to also detect failure.

When a backup entity takes over for a failed peer and receives a message over a new connection, it would retrieve the call state (using the callIdentifier as key). This will allow it to continue supporting the call, including routing signalling, maintaining billing information, etc. An entity detecting a failure shall not re-establish the connection until it has a message to send over the connection. The backup entity will have new channels for each call that used the failed peer, unless multiplexed channels are used. The policy to re-establish only when needed will spread the re-establishments over time.

Delaying re-establishment until the channel is needed for a message and the fact that the backup entity has sufficient information to establish the new channel on the other side means that a keepAlive mechanism is not needed for Method B.

Since both the recovered entity and its signalling neighbour may re-establish the connection, there is a potential race condition but we avoid the need for keepAlive messages with TCP connections. Since traffic is greater in one direction than the other and re-establishment occurs only when there is message traffic, the race condition will be rare. We can resolve the race condition by the same methods used for H.245 channel establishment. The entity with the numerically smaller h245Address shall close the TCP connection it opened and use the connection opened by the other endpoint.

For multiplexed signalling channels, detecting a failure on any call shall imply failure of the channel. When a new channel is established it shall be used for the same set of calls as the failed channel. Note that this implies that the list of calls sharing a channel must be part of the data shared between an entity and its backup entity or entities through the shared repository. After a failure, the multiplexed channel is re-established when there is a message to send for any of the calls sharing the channel. A similar race condition exists to that in non-multiplexed channels. If two signalling channels are found handling the same set of calls or any calls from the same set, it shall drop one connection.

If an entity receives a new signalling connection with a callIdentifier matching that of an existing connection, it shall verify that the connection is from either the same entity as the earlier connection or the backup call signalling address for that same entity. If either is true, the entity receiving the new connection shall consider the earlier connection as failed and close it.

R.8.4 H.245 connection re-establishment

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

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

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

R.8.5 Data items shared through shared repository

As a minimum, the following data shall be shared through a shared repository:

1) backupCallSignallingAddresses;
2) hasSharedRepository;
3) callIdentifier;
4) openLogicalChannel structures, from H.245 or fastStart.

Additional data may be shared to support recovery of unstable calls or to allow recovery of additional data that changes during stable calls (e.g., call detail records, call timing data, billing data, authorization tokens).

R.8.6 Checkpoints

In this version of this Recommendation we only maintain calls that are in the stable state. Thus the only checkpoint needed is when entering the stable state. This occurs when Connect has been sent or received and media channels in both directions are established (through H.245 or fast connect procedures).

Entities may use additional checkpoints to support recovery of unstable calls or to allow recovery of additional data that changes during stable calls.

R.9 Interworking between robustness methods

Signalling neighbours must agree on the robustness method to be used between them. It is **not** necessary that the same method be used end-to-end.

Support for robustness (any of the methods) is indicated by the originating side entity by including RobustnessGenericData field in Setup. In addition support for Method B (Shared Repository) is indicated in hasSharedRepository field of Setup. The terminating side entity indicates its support for robustness and Method B by the same fields in Connect. The choice of Method A versus Method B is then made as indicated in R.10, Procedures for recovery.

If an entity routing call signalling supports Method B (has a shared repository), it may be required to use Method B on one connection and Method A on the other connection it has for the same call. In this case, it follows the rules in R.10 independently on the two connections. If a backup entity with a shared repository receives StatusInquiry, it may reply with Status using information in the shared repository.

R.10 Procedures for recovery

1) If neighbour does not support Method B (Shared Repository) and TCP signalling is used, then StatusInquiry keepAlives shall be used. If entity has Shared Repository (even though neighbour does not), then it shall send StatusInquiry periodically. If entity does not have a Shared Repository, then only the entity nearer the called party shall send StatusInquiry periodically.
2) If an entity has a message to send on a call signalling channel (including a keepAlive StatusInquiry) and it detects failure, then it shall attempt to establish a channel to the first address in backupCallSignalAddresses (backup entity).
3) After a call signalling channel is re-established, if the neighbour does not have Shared Repository, Method A shall be used and the establishing entity shall send a Status (with the fastStart field) before the message waiting to be sent.
4) The establishing entity may also send a StatusInquiry before the message, if it wishes to audit state consistency.
5) If an entity with Shared Repository receives StatusInquiry, it shall send StatusInquiry to its neighbour on the other side to retrieve necessary state information (including fastStart data) unless it keeps all such data in its repository.
6) If an entity not having Shared Repository receives a StatusInquiry wait until it receives a Status from its neighbour on the other side (sending StatusInquiry, if necessary, to the other neighbour if signalling channel on other side is available).

R.10.1 Recovery procedures with conflicting CRV values

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

If the failed entity implements Method A, the signalling neighbour re-establishes a call signalling connection with the backup entity of the failed entity. Then the signalling neighbour shall send StatusInquiry and Status messages to the backup entity. But before sending the StatusInquiry and Status messages, the entity shall check if it is the one that originated (caller side of) the call and if it already has prior calls to the backup entity. If the signalling neighbour is on the caller side of the call and it has prior calls to the recovered entity as shown in Figure R.3, then the signalling neighbour shall assign a new unique CRV value for this call to the recovered entity and use it (in CRV IE) in all subsequent H.225.0 call signalling and RAS messages. The recovered entity shall assign a unique CRV value for this call and use it in its communication with the gatekeeper. If the signalling neighbour is on the called side of the call and it has prior calls from the recovered entity as shown in Figure R.4, then the entity shall assign a new unique CRV value in StatusInquiry message with CRV flag = 1 because it is the destination side of the call. The recovered entity shall adopt this new CRV for this call. All the subsequent H.225.0 call signalling messages for this call shall use this new CRV value. The recovered entity, if required, shall assign a unique CRV value for this call to be used in RAS messages.

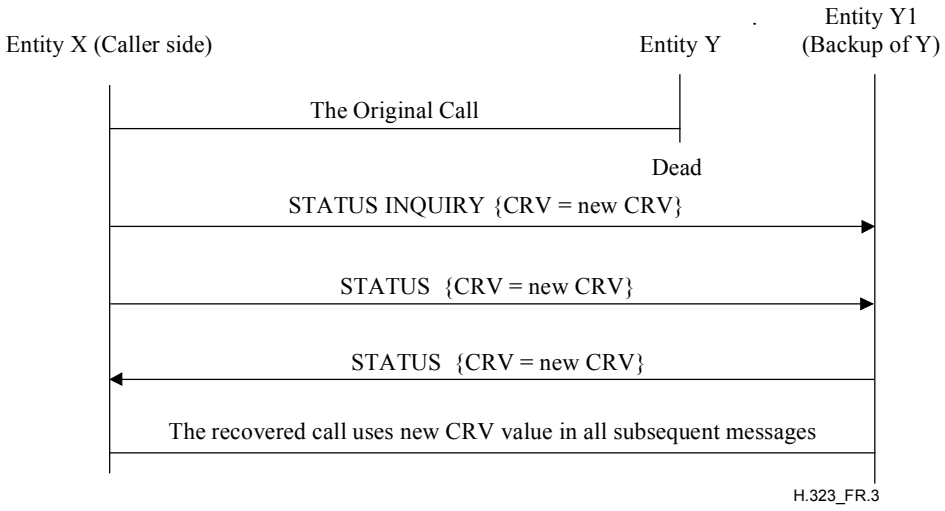


Figure R.3/H.323 – Failed entity is of Method A and called side

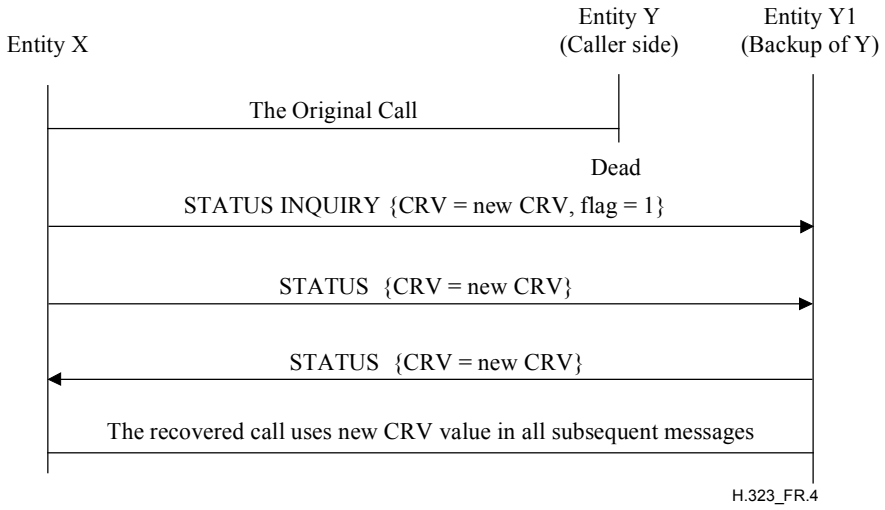


Figure R.4/H.323 – Failed entity is of Method A and caller side

If the failed entity implements Method B, the signalling neighbour or the backup entity of the failed entity can re-establish the call signalling connection. Whoever re-establishes the call signalling connection, before sending any H.225.0 call signalling messages, the entity shall check if it is the one that originated (caller side of) the call and if it already has prior calls to the recovered entity. If the entity that is re-establishing the connection is on caller side of the call and it has prior calls to the signalling neighbour as shown in Figure R.5, then the entity shall assign a new CRV value for this call and use it in all subsequent H.225.0 call signalling and RAS messages. The signalling neighbour shall assign a unique CRV value for this call and use it in RAS messages in its communication with the gatekeeper. If the entity that is re-establishing the connection is on called side of the call and it has prior calls from the signalling neighbour entity as shown in Figure R.6, then the entity shall assign a new unique CRV value and use it in a H.225.0 call signalling message with CRV flag = 1 because it is the destination side of the call. The signalling neighbour entity shall adopt this new CRV for this call. All the subsequent H.225.0 call signalling messages for this call shall use this new CRV value. The signalling neighbour entity, if required, shall assign a unique CRV value for this call to be used in RAS messages.

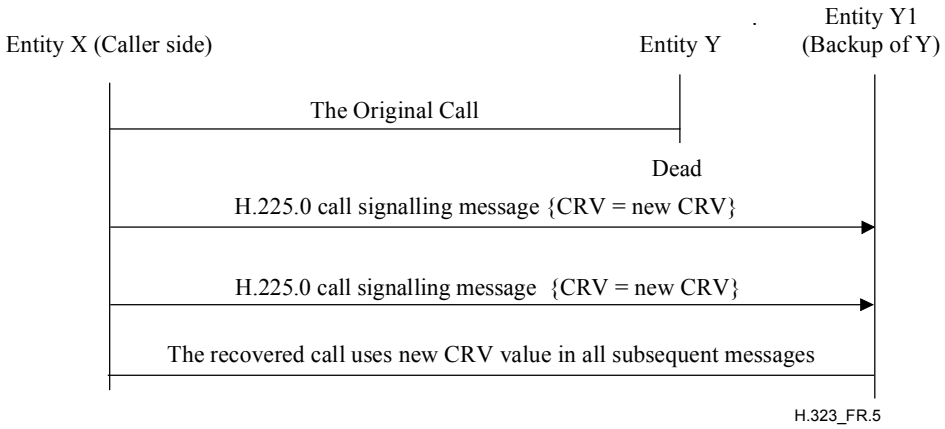


Figure R.5/H.323 – Failed entity is of Method B and called side and Survived entity initiates re-establishment

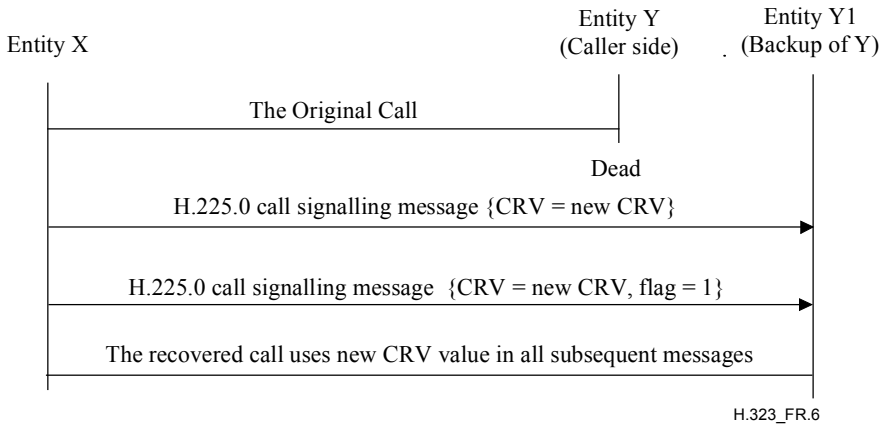


Figure R.6/H.323 – Failed entity is of Method B and caller side and survived entity initiates re-establishment

R.11 GenericData usage

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

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

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

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

Rcf-RD ::= SEQUENCE
{
     hasSharedRepository     NULL OPTIONAL,
     ...,
     irrFrequency            INTEGER (1..65535) OPTIONAL     -- in seconds;
                                                             -- not present
                                                             -- if GK does not
                                                             -- want IRRs for
                                                             -- recovered calls
}
```

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

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

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

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

The GenericIdentifier shall be 1:

```
robustnessId GenericIdentifier ::= standard:1
```

In addition a featureDescriptor carrying the robustnessId shall be included in desiredFeatures of messages specified below.

R.11.1 GenericData usage in H.225.0 messages

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

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

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

R.12 Informative Note 1: Background on robustness methods

This clause describes types of system failures and types of robustness from a general viewpoint. Not all types of system failures described are addressed by robustness methods in the current version of this annex. This more general view is provided to give context to the methods currently defined and help the reader understand which types of system failure are addressed. It also serves as a list of failures that might be addressed in future versions of this annex.

R.12.1 Types of robustness methods

System robustness can be provided in several ways:

1) hardware/operating system redundancy methods (possibly including several NIC cards);
2) tandem entities;
3) virtual entities.

R.12.2 Robust entities

Entities to be considered for robustness include essentially all H.323 entities:

1) Gatekeepers;
2) Border Elements;
3) Multipoint Controllers;
4) Possibly Multipoint Processors (for media stream failure);
5) Gateways (including IP-to-IP Gateways);
6) Firewall proxies; and
7) certain types of endpoints.

Not all robustness models may be suitable for all system components.

R.12.3 Robust system scope

The scope of robustness or the part of a system implementing robustness can include one or more of:

1) H.323 Zones (intra-zone, with one or more Gatekeepers).
2) H.323 Intra-Domain (intra-domain, inter-zone with several Gatekeepers).
3) H.323 Inter-Domains (inter-domain, with several Gatekeepers and Border Elements).

R.12.4 System termination and failures

Orderly system termination (such as an MC leaving a conference) should be catered for as well as system failure. Terminating orderly in principle allows the terminating endpoint to notify its peers thereby potentially simplifying detection but also requiring additional/slightly different mechanisms. It should be noted that the notification may not succeed due to repeated packet loss, so that the border to system failures is almost seamless.

System failure aspects are addressed in the following clauses:

R.12.4.1 Types of failures

The methods of this annex only address failures that can be detected from a protocol "on the wire" point of view. Failure of a processor on a multi-processor system with otherwise shared memory is not visible to the outside and hence is not a failure addressed by these methods. Failure of a NIC card on the other hand requires the use of a different transport address and hence is visible and is to be dealt with. The following types of failures will be visible to signalling neighbours and are targets for this work:

1) Full system component failure (power failure, software crash);
2) Partial system component failure (failure of one out of many communication interfaces);
3) Full network link failure (a system component is no longer reachable); and
4) Partial link network failure (not all system components can reach each other, but some can still communicate; this particularly includes partial connectivity and half-link failure).

It should be noted that various of these failure modes may be not only hard to detect (symmetrically) and may be indistinguishable from one another (see below).

5) Malicious attacks on the system – should be looked at in the context of the H.323 security work.

R.12.4.2 Failure detection

1) Time to detect a failure.
2) Ways of detecting a failure (explicit permanent surveillance vs detection upon invoking a function).
3) Entities responsible for/involved in detecting a failure.
4) Appearance of a failure to a system component/a set of system components.
5) Possibility to determine the type of failure.
6) Consistency/timing of failure detection among various system components.
7) Failure detection may not be transitive, i.e., from "A can/cannot talk to B" and "B can/cannot talk to C" cannot necessarily be concluded that also "A can/cannot talk to C".
8) How much overhead is acceptable?

R.12.4.3 Failure handling

1) Time to repair.
2) Entity initiating the repair process.
3) Possibility to repair the failure.
4) Consequences if the failure cannot be repaired.
5) How to ensure consistent handling of a failure by all involved entities?
6) How to deal with inconsistent views/detection of failures by various components (failed vs not)?
7) How to deal with different timing of failure detection?
8) How to deal with inconsistent state when handling a failure?
9) How to deal with gaps in state information when handling a failure?
10) Consequences on the overall system operation (e.g., an ongoing call).
11) How much overhead is acceptable?
12) How to deal with multiple simultaneous failures?

R.12.4.4 Failure scenarios

This clause lists many identified failure scenarios of H.323 systems. The robustness methods of this annex do not allow recovery from all of these failures but they are listed here for completeness and to give context to the list of failures that are covered by the robustness methods.

1) (Gatekeeper – endpoint): No relationship yet/anymore.
2) (Gatekeeper – endpoint): discovered but not registered.
3) (Gatekeeper – endpoint): discovered and registered.
4) In the process of call establishment:
   a) direct;
   b) gatekeeper-routed.

5) During a call/conference: ITU-T Rec. D.160: "stable state" – discuss what this means for the various protocols:
   a) direct;
   b) gatekeeper-routed.
6) In the process of call teardown:
   a) direct;
   b) gatekeeper-routed.

Consider the implications that arise from the various new protocols under development (H.450.x family, Annex K, Annex L of this Recommendation etc.)

Consider media streams as well as RAS/call signalling/conference control communication relationships.

R.13 Informative Note 2: Call state sharing between an entity and its backup peer

This Note suggests ways to implement call state sharing between an entity and another that serves as its backup peer. The selection of a method is not part of this Recommendation. Since the method is not standardized, peers from different vendors may not be able to serve as robust backup peers.

R.13.1 Shared-memory

If members of the cluster are physically located in the same cabinet, they may be able to use a shared (or reflective) memory device. This is similar to many fault-tolerant platforms, but might simply write to shared memory at each checkpoint rather than running a fault-tolerant OS.

R.13.2 Shared-disk

If the members of the cluster are physically located near each other, they can use a shared-disk and write state information at each checkpoint.

R.13.3 Message passing

The active entity can send a message updating the shared state to each of the other members of the cluster at each checkpoint. This implements a distributed shared memory sometimes referred to as a *bulletin board*. The messages can be sent using distinct UDP messages, multicast messages, persistent TCP links or fault-tolerant message passing protocol such as ASAP (which supports a send-to-group multicast mechanism not requiring multicast IP). This is discussed in more detail and with some suggested checkpoints in APC-1772.

R.13.3.1 SCTP/ASAP

This clause will illustrate, with an H.323 call example, the use of ASAP and SCTP for robustness purposes in an H.323 system. It will give in brief:

1) an architectural overview of an H.323 system using ASAP/SCTP;
2) a view of the necessary protocol stacks in the respective H.323 nodes; and
3) fail-over scenarios of an example H.323 call with two gatekeepers and two endpoints.

R.13.3.1.1 References

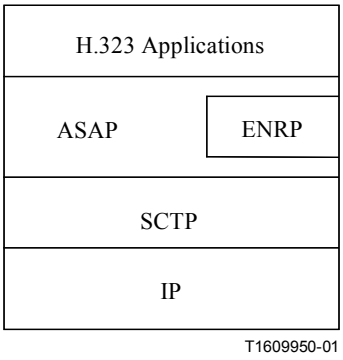
[R.13-1] IETF RFC 2960 (2000), *Stream Control Transmission Protocol*.

[R.13-2] STEWART (R.R.) et al.: *Aggregate Server Access Protocol (ASAP)*, <draft-ietf-serpool-asap-07.txt>, IETF, May 2003.

[R.13-3] XIE (Q.) et al.: *Endpoint Name Resolution Protocol (ENRP)*, <draft-ietf-rserpool-enrp-06.txt>, IETF, May 2003.

R.13.3.1.2 Protocol stacks

In general, an H.323 application using ASAP/SCTP [R.13-1] to [R.13-3] for fault tolerance will have the following protocol stack:



This can provide fast fail-over, transparent to the upper layer application, at both link and session levels:

1) link level (SCTP) – multi-homing support, surviving network failures;
2) session level (ASAP) – server pool support (2N, N+K, etc.), surviving process/node failures.

In addition, ASAP provides:

- location transparency;
- load sharing;
- Plug-n-Play, i.e., hot scalability;
- avoiding single point of failure.

R.13.3.1.3 An architectural view of an H.323 system

Figure R.7 shows an H.323 system built upon ASAP/SCTP model.

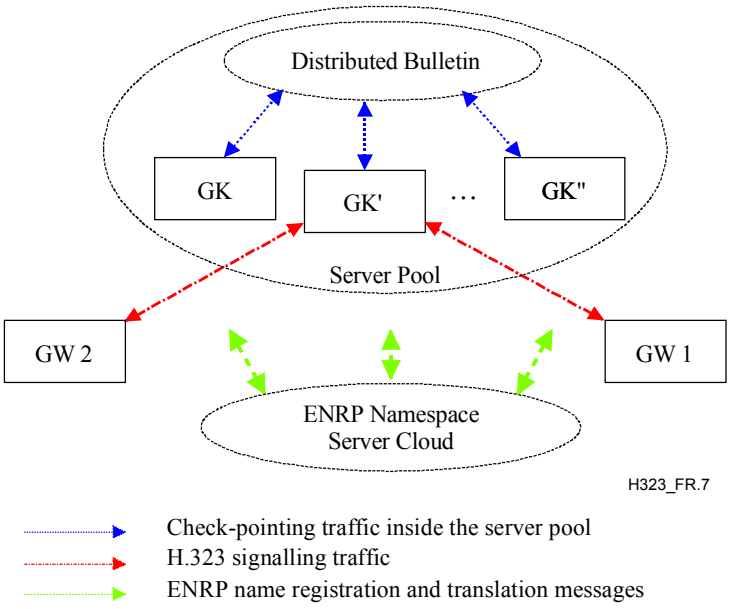


Figure R.7/H.323 – H.323 system built upon ASAP/SCTP model

In the system, all the H.323 components, including the GW 1, GW 2, and GKs employ the ASAP/SCTP stacks as shown in the previous clause. In this example, we assume that the H.323 gatekeeper is implemented as a server pool (the figure depicts the internals of the server pool), while the gateways may or may not be implemented as server pools.

As shown in the figure, inside the gatekeeper server pool we have multiple instances of functionally identical H.323 gatekeepers, GK, GK', … GK". The GK instances share call state and other call recovery critical information among themselves using an internal distributed bulletin board. The mechanism and implementation of the distributed bulletin board is vendor specific and thus out of the scope of either ASAP or SCTP (the bulletin board, however, can use ASAP/SCTP to gain fault tolerance and scalability for itself).

All the ASAP/SCTP nodes, including GWs and GKs, rely on either a single ENRP namespace server cloud or a group of bridged ENRP clouds for name registration and name translation services [R.13-2]. To form the gatekeeper server pool, all GK instances register to the ENRP namespace under the same name. However, each individual GK instance may choose to register with a different load handling capability.

Each H.323 call message will be delivered by ASAP to one of the GK instance in the server pool. The selection of the receiver GK instance is based on both the load sharing policy in effect and the current status of each GK instances in the server pool. It is sometimes very desirable to have all the H.323 signalling messages related to a call be handled by the same GK instance for the entire life cycle of the call, and only let another GK instance take over the call in case the original handler dies. We call this relationship between the call and the server instance "loose binding". ASAP is designed to support this type of "loose binding" relationship very easily [R.13-2] and [R.13-3].

Moreover, when a GK instance is handling a call, it should publish to the distributed bulletin board (i.e., "checkpoint") all critical call state information every time the call reaches a certain stage in its life cycle. This information will help the alternate GK instance to recover the call in case the original call handler crashes.

R.13.3.1.4 An example H.323 call

For the purposes of describing a call, signalling flows are used in Figure R.8.

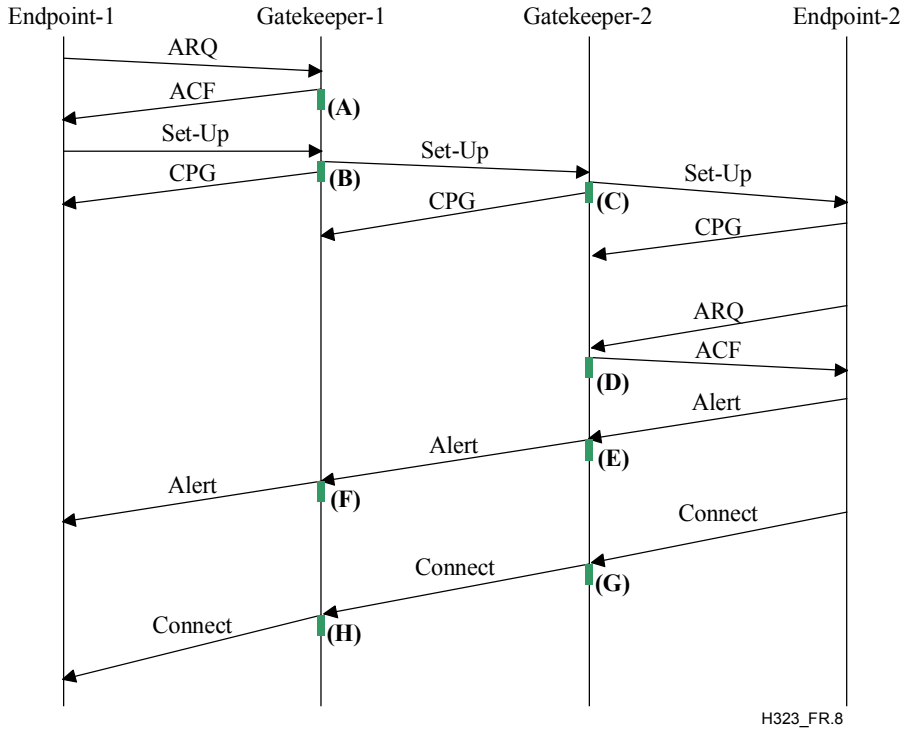


Figure R.8/H.323 – Example H.323 call

Please note that the references in this call flow are quite old and the second gatekeeper is extrapolated. There may be some differences in the way the current H.323 specification would have a call flow, but the point here is to emphasize how ASAP/SCTP would be used. Even if minor items are incorrect in the above figure, this does not invalidate the example.

R.13.3.1.4.1 General description

The call starts with Endpoint-1 requesting bandwidth. The endpoint would in this case use ASAP to query a gatekeeper, known by a name or possibly by a well-known IP address and port. In either case, an ENRP name translation query (not shown) would propagate to the endpoint the set of all gatekeepers (primary and any redundant) in the server pool. This information would be populated into a ASAP layer local cache in Endpoint-1 for future reference in case of a failure. This same caching would occur in all ASAP endpoints in the chain transparent to the call itself. Note that caching is an optional feature. Being that it is an option, endpoints not implementing it can still obtain an alternate gatekeeper, an additional query would be needed to the ENRP server at the time of failure detection.

Note we now hit point (**A**), at this step the gatekeeper allocates bandwidth and checkpoints this bandwidth utilization information message into a "bulletin board" area. This "bulletin board" area could be any of the following:

- a piece of distributed shared memory being maintained by a separate subsystem;
- a piece of reflective memory specifically built for this purpose;
- a distributed commercial database;
- some other creative invention.

Please note that the point here is that the redundant/peer gatekeepers have to share call states in some way. Any existing or future mechanism conceived to share call state can be used.

Gatekeeper-1 populates its ARQ related state and pushes this information to the "bulletin board" and responds to the request in the normal manner, i.e., with an ACF.

Endpoint-1 now reacts and sends the set-up message to Gatekeeper-1. Upon reception of the set-up message Gatekeeper-1 selects the next gatekeeper, Gatekeeper-2, and forwards its set-up, pushing the state information about the call (point (**B**)), possibly tied in some way to the previous information (perhaps with some form of cross-reference, i.e., Call-X is using Y bandwidth represented by the ARQ information). After pushing the information at point (**B**), Gatekeeper-1 sends out the call proceeding message to Endpoint-1.

Gatekeeper-2 receives the set-up message from its peer gatekeeper, selects the destination endpoint forwards the set-up and pushes state information at point (**C**) for the call. After pushing its state to the bulletin board, it sends Gatekeeper-1 a call proceeding message.

Endpoint-2, upon reception of the set-up, sends back a call proceeding message and asks its gatekeeper for bandwidth with its own ARQ message.

This causes Gatekeeper-2 to allocate bandwidth, push state at point (**D**) and send back the ACF message. Upon reception of which, Endpoint-2 sends an Alerting message to Gatekeeper-2.

Upon reception of the Alerting message, Gatekeeper-2 would push a small update to its bulletin board (point (**E**)), i.e., that the call is in Alerting, and forward an Alerting message to Gatekeeper-1.

Gatekeeper-1 will repeat the same procedure, updating its state at point (**F**) and forwarding the Alerting message.

Endpoint-2 at some point answers the call, sending a Connect message to Gatekeeper-2. Gatekeeper-2, upon reception of the Connect message, will push another small update to the state at point (**G**) indicating that the call is now in an answered state and forward the connect message to Gatekeeper-1.

Upon reception of the Connect message, Gatekeeper-1 will perform the same operation, saving its state at point (**H**) and sending the connect message on to Endpoint-1.

R.13.3.1.4.2 Failure scenarios

The above descriptions assume the maximum level of redundancy and state/call preservation. In this scenario any failure of either Gatekeeper becomes transparent to either endpoint. If a failure occurs, the message would be re-routed by ASAP to an alternate. The alternate would need to take the following actions on any message it received that it did not have a call object/block for:

- look up the call in the "bulletin board";
- pull the state information and construct a call control block or object to the call;
- continue processing the message on behalf of the dead peer.

Endpoints become completely transparent to failure scenarios. No knowledge is placed in the endpoint itself (other than ASAP) to recover from a Gatekeeper failure.

R.13.3.1.4.3 State saving issues

As stated above, the example assumes a maximum state saving model. In this mode updates to state would need to be minimized to the smallest amount of information possible. In particular, state should be limited to the smallest set of information necessary to re-construct the call AND updates should be as small as possible. In some cases an operator may not wish to have this level of redundancy. To achieve a robust system with less state, the following state sharing points could be eliminated:

- At points (**A**) and (**D**) – If the gatekeeper uses some other methodology to calculate bandwidth utilization (besides tracking the number of calls by count) these steps could be completely skipped with no harm. It may be that the operator has NO concern for admission control and its gatekeepers do not perform this, in these cases this step is not necessary.
- At points (**F**) and (**E**) – These points are optional in that they may not provide any information worth saving, i.e., the call is ringing versus still setting up.
- At points (**B**) and (**C**) – If the operator is NOT interested in saving anything but stable calls, these points can be eliminated. In this case, any calls that were being set up would be lost if a failure did occur.

Trade-offs, such as the above, are outside the scope of using ASAP/SCTP and are strictly an operator/manufacture decision as to how much state may be saved by a given implementation and what controls/options the operator may have.

Appendix I

Sample MC to terminal communication mode command

I.1 Sample conference Scenario A

Endpoints A, B and C are in an audio and video distributed conference using multicast. The MC (which could be anyone of the nodes) has decided to place the media and media control channels on the following multicast addresses:

Stream	Multicast address
Audio for all endpoints	MCA1
Audio Control for all endpoints	MCA2
Video from endpoint A	MCA3
Video Control data about endpoint A	MCA4
Video from endpoint B	MCA5
Video Control data about endpoint B	MCA6
Video from endpoint C	MCA7
Video Control data about endpoint C	MCA8

I.2 CommunicationModeTable sent to all endpoints

All entries are commands for endpoints to open a logical channel for transmission. **terminalLabel** is only present when the entry is specific to a single endpoint in the conference.

```
ENTRY 1 – AUDIO & AUDIO CONTROL FOR CONFERENCE
sessionID                       1
sessionDescription              Audio
dataType                        Audio Capability
mediaChannel                    MCA1
mediaControlChannel             MCA2

ENTRY 2 – VIDEO & VIDEO CONTROL FOR NODE A
sessionID                       2
associatedSessionID             1
terminalLabel                   M/T for A
sessionDescription              Video for Node A
dataType                        Video Capability
mediaChannel                    MCA3
mediaControlChannel             MCA4

ENTRY 3 – VIDEO & VIDEO CONTROL FOR NODE B
sessionID                       3
associatedSessionID             1
terminalLabel                   M/T for B
sessionDescription              Video for Node B
dataType                        Video Capability
mediaChannel                    MCA5
mediaControlChannel             MCA6

ENTRY 4 – VIDEO & VIDEO CONTROL FOR NODE C
sessionID                       4
associatedSessionID             1
terminalLabel                   M/T for C
sessionDescription              Video for Node C
dataType                        Video Capability
mediaChannel                    MCA7
mediaControlChannel             MCA8
```

I.3 Sample conference Scenario B

Endpoints A, B and C are in a multipoint conference where audio is unicast from each endpoint and centrally mixed, but video is multicast from the endpoints. The MC may send a unique CommunicationModeCommand to each endpoint, or it may send the same message to all endpoints if the table entries are identified by the destination endpoint's label. For this example, assume that the same message is sent to all endpoints.

Stream	Multicast Address
Audio from endpoint A	UCA1
Audio Control data about endpoint A	UCA2
Audio from endpoint B	UCA3
Audio Control data about endpoint B	UCA4
Audio from endpoint C	UCA5
Audio Control data about endpoint C	UCA6
Video from endpoint A	MCA1
Video Control data about endpoint A	MCA2
Video from endpoint B	MCA3
Video Control data about endpoint B	MCA4
Video from endpoint C	MCA5
Video Control data about endpoint C	MCA6

I.4 CommunicationModeTable sent to all endpoints

All entries are commands for endpoints to open a logical channel for transmission. **terminalLabel** is only present when the entry is specific to a single endpoint in the conference.

```
ENTRY 1 – AUDIO & AUDIO CONTROL FOR NODE A
sessionID                       1
sessionDescription              Audio
terminalLabel                   M/T for A
dataType                        Audio Capability
mediaChannel                    UCA1
mediaControlChannel             UCA2

ENTRY 2 – AUDIO & AUDIO CONTROL FOR NODE B
sessionID                       2
sessionDescription              Audio
terminalLabel                   M/T for B
dataType                        Audio Capability
mediaChannel                    UCA3
mediaControlChannel             UCA4

ENTRY 3 – AUDIO & AUDIO CONTROL FOR NODE C
sessionID                       3
sessionDescription              Audio
terminalLabel                   M/T for C
dataType                        Audio Capability
mediaChannel                    UCA5
mediaControlChannel             UCA6

ENTRY 4 – VIDEO & VIDEO CONTROL FOR NODE A
sessionID                       4
associatedSessionID             1
terminalLabel                   M/T for A
sessionDescription              Video for Node A
dataType                        Video Capability
mediaChannel                    MCA1
mediaControlChannel             MCA2
```

```
ENTRY 5 – VIDEO & VIDEO CONTROL FOR NODE B
sessionID                       5
associatedSessionID             2
terminalLabel                   M/T for B
sessionDescription              Video for Node B
dataType                        Video Capability
mediaChannel                    MCA3
mediaControlChannel             MCA4

ENTRY 6 – VIDEO & VIDEO CONTROL FOR NODE C
sessionID                       6
associatedSessionID             3
terminalLabel                   M/T for C
sessionDescription              Video for Node C
dataType                        Video Capability
mediaChannel                    MCA5
mediaControlChannel             MCA6
```

Appendix II

Transport level resource reservation procedures

II.1 Introduction

H.323 recommends the use of transport level resource reservation mechanisms to fulfil the QOS requirements of real-time video and audio streams. Although the transport level resource reservation mechanisms themselves are beyond the scope of this Recommendation, the general method and coordination of these transport level mechanisms between H.323 entities is described in this appendix to prevent conflicting interoperability issues.

This appendix describes the use of RSVP (Resource reSerVation Protocol) as a possible mechanism for providing transport level QOS over IP-based networks. Other protocols may be used; however, the basic procedures defined in this appendix should still apply. Participants in a conference should be able to signal their intentions, capabilities, and requirements in a standard, protocol-specific manner. In addition, the signalling sequence of the resource reservation mechanisms must be specified such that the call establishment interval is minimal.

RSVP is the transport level signalling protocol for reserving resources in unreliable IP-based networks. Using RSVP, H.323 endpoints can reserve resources for a given real-time traffic stream based on its QOS requirements. Only best-effort delivery of the packets is possible if the network fails to reserve the required resources or if RSVP is absent.

II.2 QOS support for H.323

When an endpoint requests admission with a Gatekeeper, it should indicate in the ARQ message whether or not it is capable of reserving resources. The Gatekeeper should then decide, based on the information it receives from the endpoint and on information it has about the state of the network, either:

- to permit the endpoint to apply its own reservation mechanism for its H.323 session; or
- to perform resource reservation on behalf of the endpoint; or
- that no resource reservation is needed at all. Best-effort is sufficient.

This decision is conveyed to the endpoint in the ACF message. The endpoint shall accept the Gatekeeper's decision in order to place a call.

The Gatekeeper should reject an endpoint's ARQ, if the endpoint does not indicate that it is capable of resource reservation, and the Gatekeeper decides that resource reservation must be controlled by the endpoint. In this case, the Gatekeeper should send an ARJ back to the endpoint.

The specific field in H.225.0 RAS signalling to permit this functionality is the **transportQOS** field.

In addition to **transportQOS**, an endpoint should also calculate and report the bandwidth it currently intends to use in all channels of the call. This bandwidth should be reported in the **bandWidth** field of the ARQ message independent of the decision by the endpoint to use RSVP signalling or not. In addition, if bandwidth requirements change during the course of the call, an endpoint should report changes in bandwidth requirements to the Gatekeeper using BRQ independent of the decision to use RSVP.

RSVP reservations can only be made by network entities which are in the path of media flow between endpoints. It is possible through Gatekeeper routed call signalling to route media streams through a Gatekeeper. However, most of the time media channels will be routed between endpoints without passing through the Gatekeeper. If a Gatekeeper decides to route media streams, then the procedures followed should be identical to those for RSVP signalling directly from the endpoints. It is best if RSVP reservations are made directly by the endpoints since this will reserve resources along the entire routed path of the call. The remainder of this appendix discusses the use of RSVP by the H.323 endpoints.

Some of the salient points of RSVP are as follows:

- RSVP supports both unicast and multicast environments;
- RSVP is tied to specific streams (i.e., specific Transport Address pairs);
- RSVP is soft-state based, and therefore adapts dynamically to changing group membership and routes;
- RSVP is unidirectional;
- RSVP is receiver-oriented – the recipient of the media stream makes the reservation (scalable).

II.3 RSVP background

In the following description, the high-level usage of RSVP in a simple H.323 conference will be outlined.

In Figure II.1, Endpoint A wishes to send a media stream to Endpoint B. Therefore, it has to open a logical channel to B. RSVP signalling for resource reservation should be a part of the opening logical channel procedure. Endpoint A would cause RSVP *Path* messages to be sent out to B. These *Path* messages go through routers and leave "state" on their way tracing towards B. *Path* messages contain the complete source and destination addresses of the stream and a characterization of the traffic that the source will send. Endpoint B would use the information from the *Path* to make the RSVP *Resv* request for the full length of the path. *Resv* messages contain the actual reservation and will generally be the same as the traffic specification in the *Path* message.

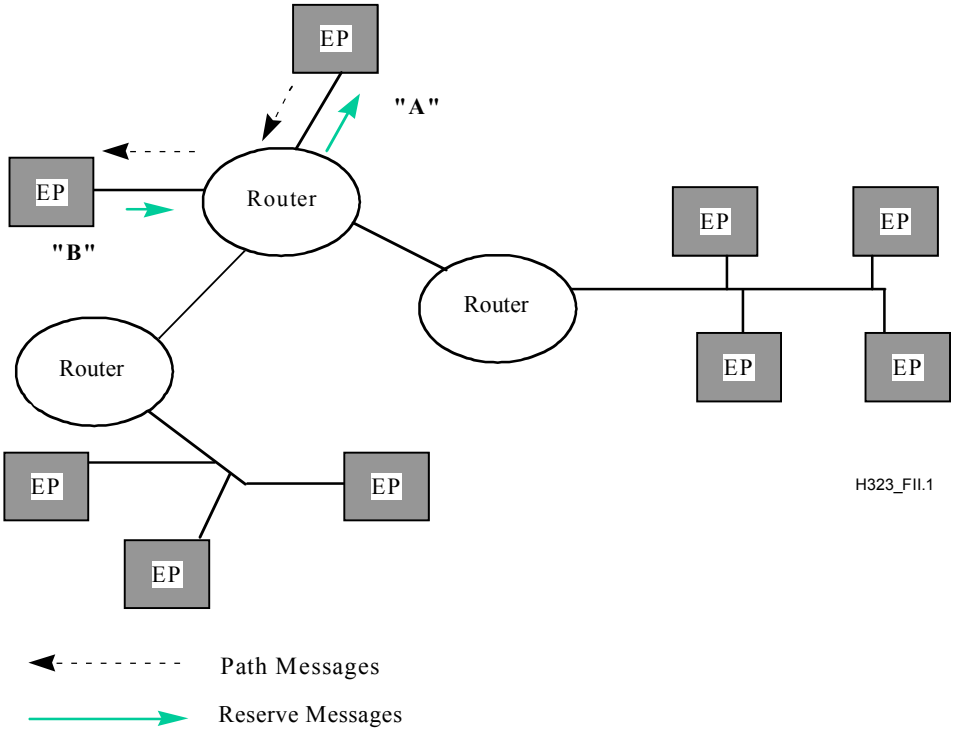


Figure II.1/H.323 – Resource reservation for a point-to-point connection

In Figure II.2, a multipoint conference is shown. The *Path* messages are utilized in the same manner as the simpler point-to-point case. It should be noted that the *Resv* requests are aggregated by the routers to keep redundant reservation requests from occurring upstream.

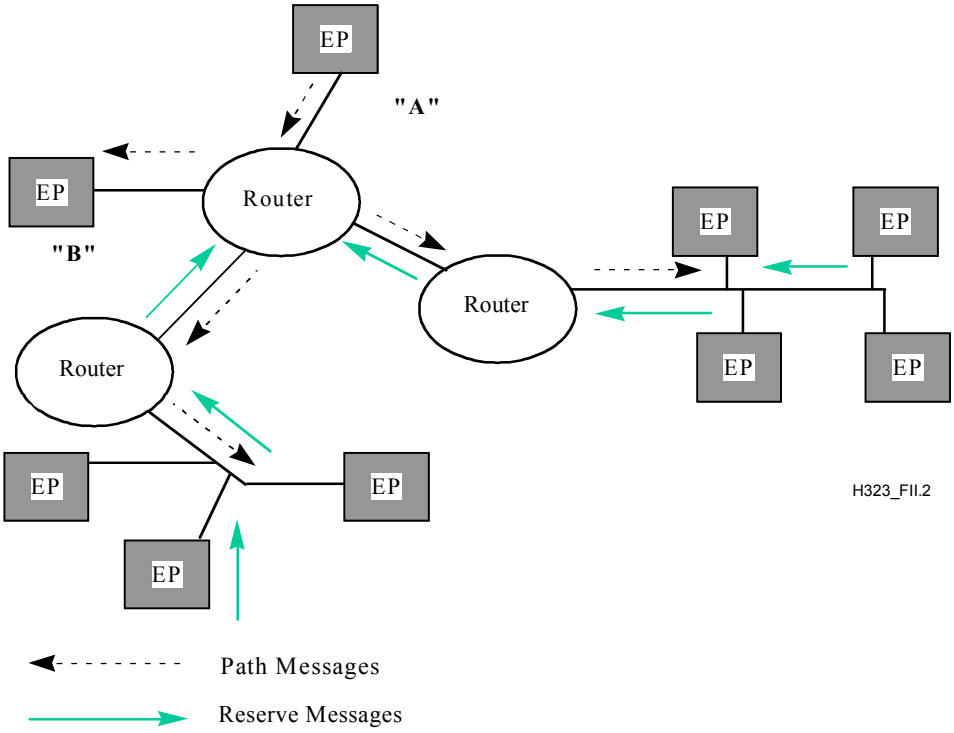


Figure II.2/H.323 – Resource reservation for a point-to-multipoint connection

Path messages must contain the complete destination/source addresses and a traffic specification. *Resv* messages contain the reservation parameters and the required service. *Path* and *Resv* messages for a given traffic stream should be sent as part of the **openLogicalChannel** procedure for that particular stream. The reservation should be released during the **closeLogicalChannel** procedure using the RSVP *PathTear* and *ResvTear* messages.

Note that RSVP Path and Resv messages use the same IP address/port pair as the media to be delivered between endpoints. This means that these messages must be filtered out of the media stream by the endpoints. This is not an issue for endpoints which do UDP filtering since RSVP messages themselves are not UDP messages. Even so, the sender of a media stream should not use RSVP when the receiver is not capable of it. RSVP capabilities are exchanged as part of the capability exchange and open logical channel procedures.

RSVP is only a signalling protocol. Together with the appropriate QOS services (e.g., guaranteed QOS or controlled-load service), scheduling mechanisms (e.g., weighted fair queuing), and policy-based admission control module (e.g., local policy manager), RSVP is capable of satisfying the QOS requirements of H.323 conference participants. In addition, RSVP is designed for point-to-point links. If a path traverses a shared link, RSVP invokes the appropriate resource reservation mechanism for the specific shared medium, e.g., SBM (Subnet Bandwidth Management) in case of Ethernet. All the mechanisms mentioned in this paragraph are controlled completely from within RSVP. Therefore, all that an H.323 endpoint needs is RSVP signalling.

II.4 The H.245 capability exchange phase

During the H.245 capability exchange phase, each endpoint indicates its transmit and receive capabilities to the other endpoint. The **qOSCapability** is part of the capability exchange. However, it is not stream-specific. Therefore, the RSVP parameters if specified in the **qOSCapability** would represent an aggregate for all streams (either those to be transmitted or those to be received). Such parameters will not be of any use to the other endpoint. Therefore, the only RSVP-related information an endpoint should convey to the other endpoint in the capability set is whether or not it is RSVP-capable.

To signal RSVP capability, an endpoint shall set the appropriate available **qOSMode** fields within the capability PDU during capability exchange. Endpoints which do not receive RSVP capabilities from the receiving endpoint shall not use RSVP when opening logical channels.

II.5 Open logical channel and setting up reservations

In this clause, we describe the steps that should be followed for opening an H.245 logical channel and reserving resources for a given traffic stream. Reservations are established only if both endpoints indicate that they are RSVP enabled during capability exchange. We consider only the point-to-point case. The case of point-to-multipoint (multicast) connections will be discussed in II.7.

The sender shall specify the RSVP parameters of the stream to be transmitted and the integrated services the sender supports in the **qOSCapability** field of the **openLogicalChannel** message. In case of a point-to-point stream, the sender does not specify a receiver port ID in the **openLogicalChannel** message. This ID is selected by the receiver after receiving the **openLogicalChannel** and is returned to the sender in the **openLogicalChannelAck** message. Only then can the sender create an RSVP session for that stream (to create an RSVP session for a given stream means that the endpoint registers with RSVP to get notified when messages arrive that may affect the state of the RSVP reservation for that stream) and start emitting RSVP *Path* messages. The receiver has sufficient information to create an RSVP session for the same stream before sending the **openLogicalChannelAck** message. The information needed to create an RSVP session and initiate RSVP processing are: the receiver IP address in case of point-to-point or the group

multicast IP address in case of point-to-multipoint, the receiver port ID, and the protocol (always UDP in case of H.323 audio and video streams on IP networks).

A receiver may not want to start receiving stream packets until the RSVP reservations are in place. To achieve this, the receiver may set the Boolean **flowcontrolToZero** field of the **openLogicalChannelAck** message to TRUE to indicate that it does not wish to receive any traffic on that channel before the resource reservations are complete. When a sender receives an **openLogicalChannelAck** message with **flowControlToZero** set to TRUE, the sender shall not transmit any traffic on that channel.

When the receiver starts receiving the sender's *Path* messages, it should start sending RSVP *Resv* messages. When the receiver receives an RSVP *ResvConf* message confirming that reservations have been established, it may send a **flowControlCommand** to the sender unrestricting the bit rate of the traffic stream, i.e., cancelling the effect of the previous **flowcontrolToZero** field in the **openLogicalChannelAck** message. When the sender receives the **flowControlCommand** it starts transmitting packets.

Note that the *ResvConf* message and similarly all other RSVP messages are transmitted unreliably. As a result, they may get delayed or even lost. An endpoint should be aware of that fact and set timers with appropriate value while waiting for a *ResvConf*. The action taken if the endpoint times out without receiving a *ResvConf* is up to the individual endpoint vendors.

The behaviour of an endpoint if RSVP reservations fail at any point during an H.323 call is not specified in this appendix and is left to the individual vendors. However, if an RSVP reservation fails and the receiving endpoint decides that best-effort level of service is not acceptable, it may request to close its logical channel using the **requestChannelClose** message. The **closeReason** field is available in the **requestChannelClose** message to allow the receiver to signal to the sender that the RSVP reservation has failed. Along with the failure indication, **requestChannelClose** includes **qOSCapability** which can be used by the receiver to tell the sender the resources which are actually currently available on the path from the sender to the receiver. At this point, the sender can decide to try to reopen the channel with a lower bandwidth codec and/or data format and go through the Open Logical Channel procedure again.

All RSVP *Resv* requests shall use the same reservation style, the **Fixed Filter** style, for the following reasons:

- Shared filter styles reduce to fixed filters in case of point-to-point calls.
- Different reservation styles for the same session cannot be merged in the network. For example, if in a multipoint call some of the receivers request fixed filter reservations while the rest request shared explicit reservations, then either the fixed filter reservations or the shared explicit reservations will fail.
- Shared reservations, created by wildcard filter and shared explicit filter styles, are appropriate for those multicast applications in which multiple data sources are unlikely to transmit simultaneously. In distributed multipoint H.323 calls, there is no mechanism to permit only one source to transmit at a specific time. On the other hand, in centralized multipoint H.323 calls, the MCU is the only multicast source. Shared reservation styles are not suited for either case.

It is up to the endpoint vendors to choose which intserv QOS service (guaranteed QOS or controlled-load) to use. However, any RSVP-enabled H.323 endpoint shall support the controlled-load service as a least common service. This requirement is necessary to avoid interoperability problems that may arise from RSVP-enabled H.323 endpoints which do not support a common intserv QOS service.

Figure II.3 shows the sequence of messages in case of successful RSVP reservation.

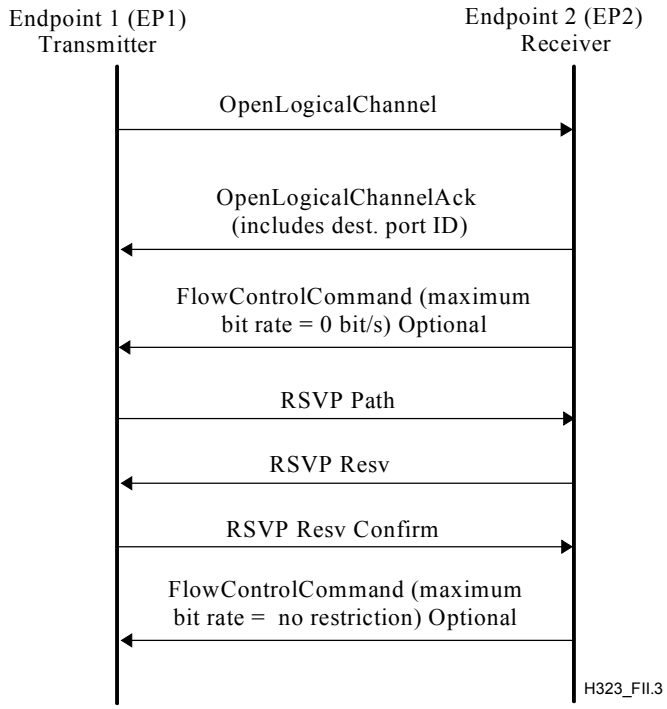


Figure II.3/H.323 – Message sequence for opening a unicast logical channel with RSVP

II.6 Close logical channel and tearing down reservations

Before sending out a **closeLogicalChannel** message for a given traffic stream, a sending endpoint should send a *PathTear* message if an RSVP session has been previously created for that stream. When a receiving endpoint receives a **closeLogicalChannel** for a given traffic stream, it should send a *ResvTear* message if an RSVP session has been previously created for that stream.

II.7 Resource reservation for multicast H.323 logical channels

The H.245 **openLogicalChannel** procedure is point-to-point even if the traffic stream involved is a multicast stream. However for the receiving endpoint to start receiving packets of a multicast stream, it has to join the multicast group and get connected to the source's multicast tree. When a receiver receives an **openLogicalChannel** message, it joins the multicast group and the source's multicast tree using standard IGMP procedures. The IGMP join (using IGMP *Report* message) takes place before the receiver sends an **openLogicalChannelAck** back to the sender.

In case of a multicast stream, the sender specifies the receiver port ID in the **openLogicalChannel** message instead of receiving the receiver port ID in the **openLogicalChannelAck** message.

The receiver may set the **flowControlToZero** field of the **openLogicalChannelAck** message to TRUE, similar to the unicast case. However, the sender (an endpoint in a distributed conference or an MCU in centralized conference) should decide not to interrupt the data stream on the opened channel, if it determines this interruption may affect other receivers of the same multicast group which are already receiving that stream. As a result, in the multicast case, the receiver may initially receive the data at best-effort until the RSVP reservations are established.

Figure II.4 shows the sequence of messages required to open a logical channel and to join the multicast tree and to reserve resources for a multicast stream.

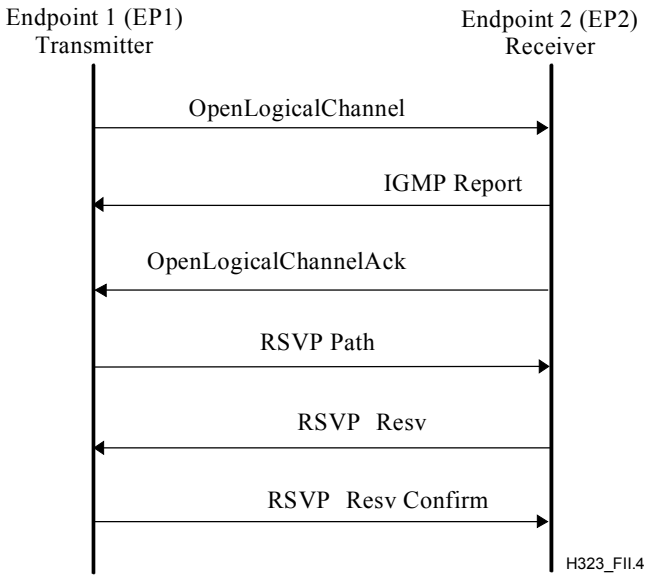


Figure II.4/H.323 – Message sequence for opening a multicast logical channel with RSVP

Before sending out a **closeLogicalChannel** message for a given multicast stream, a sending endpoint should send an RSVP *PathTear* message if the logical channel being closed is the last channel carrying that multicast stream and if an RSVP session has been previously created for that stream. When a receiving endpoint receives a **closeLogicalChannel** for a given multicast stream, it should send an RSVP *ResvTear* message and an IGMP *Leave* message, if an RSVP session has been previously created for that stream.

II.8 Synchronized RSVP

Synchronized RSVP is defined as the process of reserving resources with RSVP prior to transitioning to the Alerting phase of the call. Details of synchronizing RSVP without Fast Connect and with Fast Connect, respectively, are discussed in the following two subclauses. This clause introduces the general concept of a prioritized list of QOS levels, expressed by each endpoint from which a new set of QOS levels 'D' is derived. This derived set 'D' comprises the intersection of the two preferred **QOSMode** sets. The two endpoints can attempt to establish RSVP reservations based on a QOS level in the derived set starting with the most preferred QOS level.

Upon deriving the QOS set, the called endpoint suppresses the Alerting phase of the call until reservations are established in both directions. On successful reservation establishment, the Alerting can proceed, and call setup is resumed. In the event of failures, the lowest QOS level in the derived set is examined. If this is indicated to be "best effort", the call setup procedures are resumed; otherwise, the call is released. Sending a **QoSCapability** structure with an empty **QOSMode** element in the **rsvpParameters** block shall indicate a "best effort" level of QOS. The **QOSMode** sequence is prioritized by the **QOSMode** element of the **rsvpParameters** block with the priority decreasing from the first element to the last. **GuaranteedQoS** is the highest level of QOS that an endpoint can receive, and "best effort" is the lowest. If the preferred QOS that the calling endpoint wishes to receive is higher than "best effort", the endpoint should start RSVP procedures by listening for PATH messages from the called endpoint.

The called endpoint shall examine the sequence of **QoSCapability** structures, if present, and compare it to its own preferred set of QOS levels based on **QOSMode**. It then derives a new set of QOS levels 'D' based on **QOSMode** that represents the intersection of QOS levels from the preferred sets of the two endpoints. This new set denotes the different QOS levels in a prioritized order based on **QOSMode** that are supported by both endpoints. For example, if the calling endpoint's preferred set of QOS levels is {**GuaranteedQOS**, **ControlledLoad**} and that of the

called endpoint is {**ControlledLoad**, "best effort"}, the derived set representing the intersection is {**ControlledLoad**}. Based on the preferred QOS levels of the two endpoints, different general cases are possible. The different cases and the corresponding call handling are shown in the Table II.1.

Table II.1/H.323 – Handling calls for various QOS classes

QOS Scenario	Example	Call handling
1) The Derived QOS Set 'D' is empty	Preferred set of Calling Endpoint : {GQ} Preferred set of Called Endpoint : {CL,BE} Derived QOS Set 'D' : {}	The called endpoint shall release the call.
2) The Derived QOS Set 'D' has just one QOS level: "best effort"	Preferred set of Calling Endpoint : {BE} Preferred set of Called Endpoint : {CL,BE} Derived QOS Set 'D' : {BE}	The called endpoint shall not attempt RSVP procedures. It shall, however, continue with call setup procedures.
3) The Derived QOS Set 'D' has at least one QOS level higher than "best effort"	Preferred set of Calling Endpoint : {GQ,CL,BE} Preferred set of Called Endpoint : {CL,BE} Derived QOS Set 'D' : {CL,BE}	The called endpoint shall suppress Alerting and attempt Synchronized RSVP. The detailed procedures are described in the individual subclauses below.
BE "Best Effort" CL ControlledLoad GQ GuaranteedQoS		

In the event of failure in RSVP procedures, the called endpoint shall examine the next most preferred QOS, if present, in the derived set 'D'. If a QOS level other than "best effort" exists, the called endpoint should reinitiate RSVP reservations with that QOS level. In the event of successive failures, it is possible to reattempt RSVP reservation procedures for all QOS levels (other than "best effort") in the derived set. On expiry of the reservation timer on the called endpoint or, if the called endpoint fails to establish RSVP reservations with the lowest non-"best effort" level of QOS in the derived set, the called endpoint shall examine the lowest level of QOS in the derived set. If this QOS level is not "best effort", the called endpoint shall release the call; otherwise, the call setup is resumed with a QOS level of "best effort". Reservation failures and expiry of the reservation timer are handled similarly on the calling endpoint.

The following two subclauses discuss Synchronized RSVP and Synchronized RSVP with Fast Connect, respectively, using the concept of the prioritized **QOSMode** derived list.

II.8.1 Synchronizing RSVP when not using Fast Connect

A calling endpoint that wishes to reserve resources via synchronized RSVP when not placing a Fast Connect call shall, as a prerequisite, include an H.245 address in the Setup message. Likewise, a called endpoint that wishes to reserve RSVP resources prior to call setup completion shall retrieve the calling endpoint's H.245 address, if present, from the incoming Setup message. Subsequently, the called endpoint shall establish the H.245 Control Channel and commence H.245 procedures. Until H.245 and RSVP procedures have completed, the called endpoint shall not continue with the H.225.0 call setup phase. It is recommended, however, that the called endpoint return a Call Proceeding message to the calling endpoint to prevent any H.225.0 timer on the originating side from expiring.

If the called endpoint desires to attempt synchronized RSVP, yet the calling endpoint does not include its H.245 address in the incoming Setup message, then the calling endpoint shall assume that the originating endpoint will not accept or initiate synchronized RSVP procedures. It is then the responsibility of the called endpoint to decide on the appropriate action to take, based on the derived QOS mode as discussed in II.8. Similarly, if the calling endpoint desires to attempt synchronized RSVP and has included its H.245 address in the Setup message, yet the called endpoint has failed to establish the H.245 Control Channel and has resumed with H.225.0 procedures, then it is up to the calling endpoint to determine which action to take, based on the derived QOS mode as shown in Table II.1.

Otherwise, if the calling endpoint has offered its H.245 address in the Setup message and the called endpoint has established the H.245 Control Channel, H.245 procedures will progress as usual through master-slave determination and capability exchange.

During the H.245 capability exchange, endpoints wishing to attempt RSVP are required to include a sequence of **qOSCapabilities** (as part of the **transportCapability** element of the **H2250Capability** structure), prioritized by the **qosMode** (e.g., **guaranteedQOS**, **controlledLoad**) element of the **rsvpParameters**.

Likewise, when opening logical channels using H.245, each endpoint shall specify the RSVP parameters of the stream to be transmitted in the **qOSCapability** field of the **openLogicalChannel** message.

Upon receiving an OLC message from its peer and under the condition that the peer has indicated during capability exchange that it is RSVP enabled, the endpoint shall start listening for incoming Path messages. When it receives a Path message, the endpoint shall respond by sending a Resv message along the receive stream.

Upon receiving an OLC ACK message from its peer, the endpoint shall start sending Path messages to its peer along its transmit stream. RSVP procedures have successfully completed when the endpoint has received a Resv Confirm in response to its Resv message transmission and a Resv message in response to its Path message. If multiple streams are involved (e.g., voice, video, and data), then the endpoint must wait for reservation confirmation for all streams requiring RSVP-based QOS.

It is recommended that the endpoint start a timer for a short amount of time (e.g., five or six seconds), once it has attempted RSVP. If the timer expires before the RSVP reservations have completed, then the endpoint can determine appropriate action to take.

In the case that RSVP procedures (and therefore H.245 procedures) have successfully completed before the timer expires, the called endpoint may then resume normal call setup procedures by returning an Alerting message to the calling endpoint. If, however, the attempt to reserve RSVP resources fails, then it is the individual endpoint's responsibility to decide on appropriate action to take, based on the derived **QOSMode** set, as described in II.8. In any case, it is recommended that if the call has reached the Alerting phase of the call and RSVP reservations have failed, then the call is allowed to proceed.

Figure II.5 illustrates the modified call flow for a successful synchronized RSVP when not using Fast Connect.

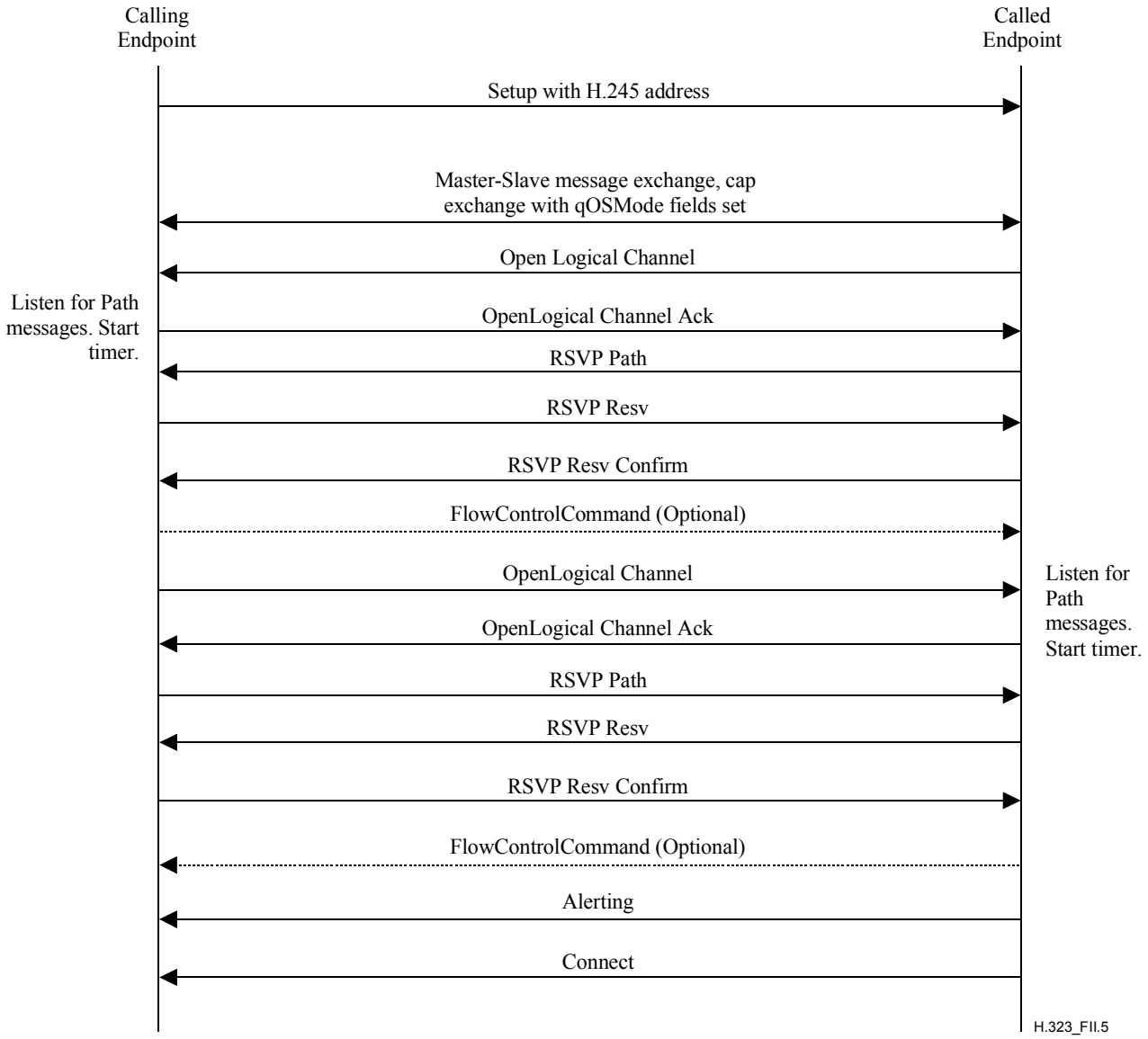


Figure II.5/H.323 – Synchronizing RSVP when not using Fast Connect

II.8.2 Synchronizing RSVP with fast connect

This clause describes synchronizing Fast Connect call setup procedures with RSVP reservation procedures in order to eliminate transporting in-band ringing before the reservations have been established.

A calling endpoint that wishes to use RSVP in a Fast Connect procedure shall send a sequence of prioritized **QoSCapability** structures in the **OpenLogicalChannel** structures contained in the **fastStart** element of the Setup message.

Upon receiving the Fast Connect Setup message, the called endpoint shall derive the **QOSMode** set using the mechanism described in Table II.1. Assuming that the derived set contains a valid (i.e., non-best effort intersection), the called endpoint shall respond to the Setup message from the calling endpoint by sending a **fastStart** element including only the **QoSCapabilities** indicated in the derived QoS set. The **fastStart** element shall be sent as soon as possible (e.g., in a Call Proceeding message) to expedite the resource reservation. The calling endpoint's set will be a subset of the list sent by the calling endpoint in the **OpenLogicalChannel** structures and will, similarly, be a sequence in decreasing order of priority by **QOSMode**. Each **QoSCapability** included in the **OpenLogicalChannel** in the response message indicates an acceptance of the corresponding

QOS level by the called endpoint. The **OpenLogicalChannel** structures in the **fastStart** element also contain information about the media ports used on the called endpoint.

The called endpoint shall initiate RSVP procedures by sending a PATH message to its peer along the transmit stream. In addition, the endpoint may use a reservation timer which would represent the total time available to establish synchronized RSVP reservations for any QOS level (other than "best effort") in the derived set. Furthermore, the called endpoint shall respond to an incoming PATH message with a RESV message along the receive stream. Note that the called end should suppress the Alerting phase of the call and not send an Alerting message to the calling endpoint until reservations are established in both directions. After RSVP procedures are established, the called endpoint shall continue with the H.225 call setup procedures.

When the calling endpoint receives the **fastStart** element, it shall extract the media port information in the **OpenLogicalChannel** and also record the prioritized list of **QoSCapabilities** returned by the called endpoint. The endpoint shall start sending PATH messages to its peer along the transmit stream. Also, when it receives a PATH message from the called endpoint, it shall respond with a RESV message along the receive stream. The calling endpoint may start a reservation timer that would represent the total time available to establish synchronized RSVP reservations.

The establishment of RSVP reservations is said to have successfully completed when the called endpoint receives a RESV message in response to its PATH message and a RESV CONFIRM message in response to its RESV message. As soon as the RSVP procedures are completed successfully, the called endpoint shall stop the reservation timer and resume with the call setup procedures. It subsequently sends Alerting/Connect messages to the calling endpoint. Figure II.6 illustrates the call flow for a successful synchronized Fast Connect call.

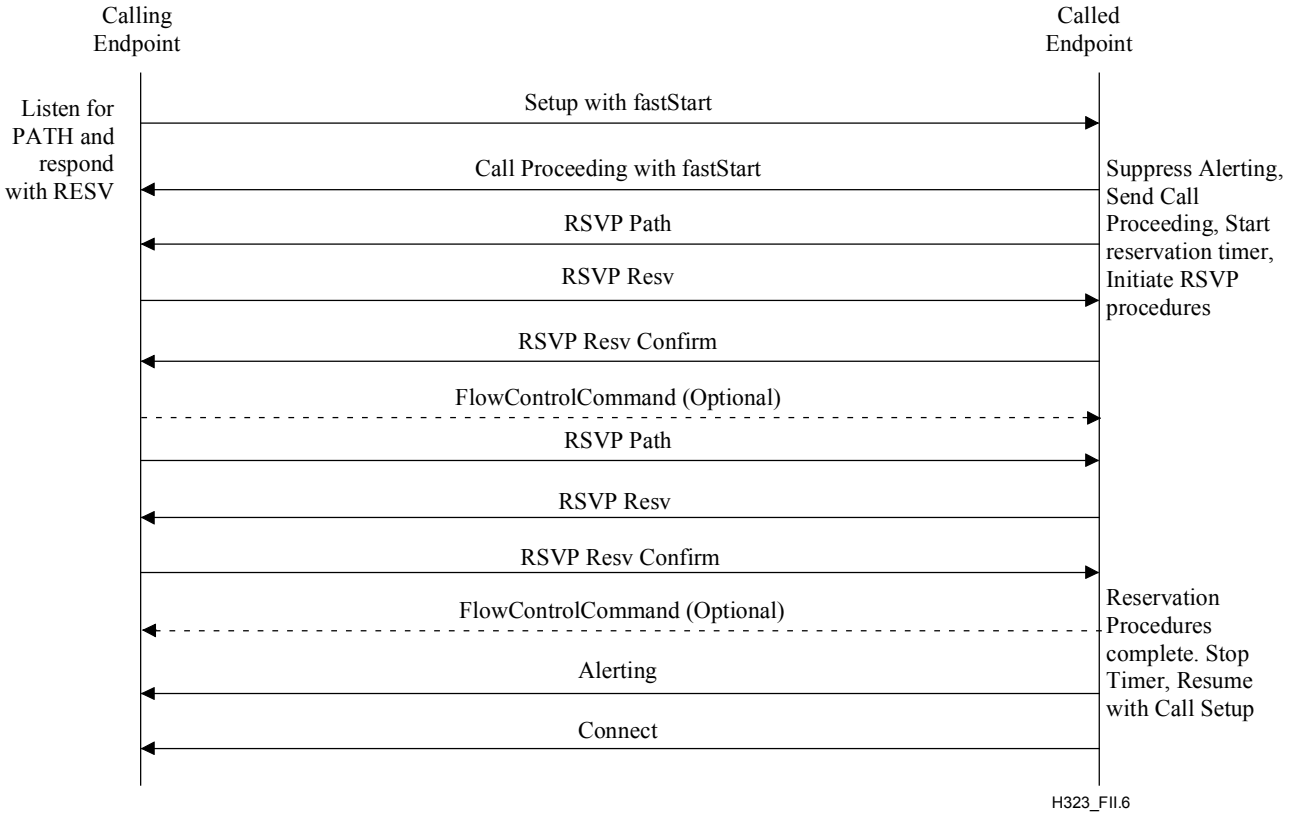


Figure II.6/H.323 – Synchronizing RSVP when using Fast Connect

In the event of RSVP failure, the called endpoint will take action according to the derived **QOSMode** set, as described in II.8.

Appendix III

Gatekeeper-based user location

III.1 Introduction

This appendix gives examples of how a Gatekeeper/proxy can implement user location services. These services depend on the Gatekeeper using the Gatekeeper routed call signalling model.

III.2 Signalling

In the scenario shown in Figure III.1, the Gatekeeper implements a "divert on no reply" service. Endpoint 1 calls Endpoint 2 with the Call Signalling Channel routed through the Gatekeeper. If there is no answer after some timeout, the Gatekeeper diverts the call to an alternate endpoint. Messages (1) to (5) show the Gatekeeper attempting to establish a call between Endpoint 1 and Endpoint 2. In this example, Endpoint 2 does not answer and so the Gatekeeper clears the call to Endpoint 2 by sending Release Complete (6). The Gatekeeper then tries Endpoint 3 by sending Setup (7). When Endpoint 3 answers the call using Connect (9), the Gatekeeper forwards the Connect (10) back to Endpoint 1.

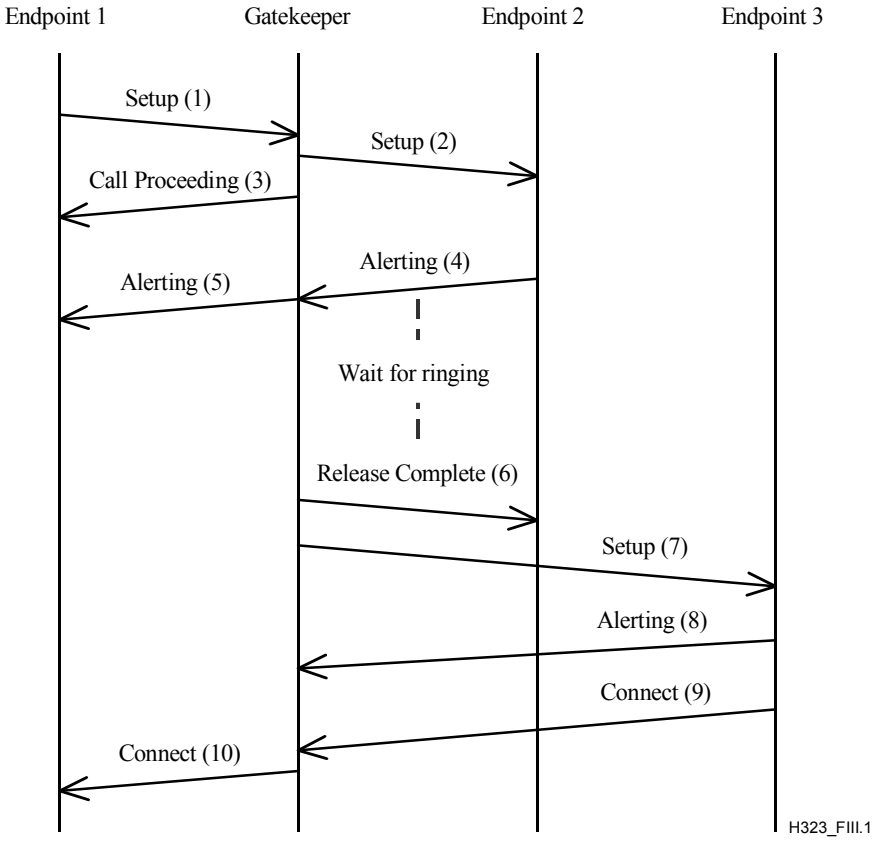


Figure III.1/H.323 – Example of user location using H.225.0 call signalling (RAS signalling not shown for clarity)

A similar approach can be used to provide "divert on busy" service. In this case, Endpoint 2 would return a Release Complete indicating that is busy. The Gatekeeper would then attempt to establish a call to Endpoint 3.

In the scenario shown in Figure III.2, the Gatekeeper attempts to establish contact with Endpoints 2 and 3 simultaneously by sending Setups (2) and (3). In this example the user at Endpoint 3 answers by sending Connect (7). The Gatekeeper forwards the Connect (8) back to Endpoint 1 and clears the call attempt to Endpoint 2 using Release Complete (9). The Gatekeeper should ignore any Connect message received from Endpoint 2 which arrives after the Connect (8) message from Endpoint 3 so that only one call is completed.

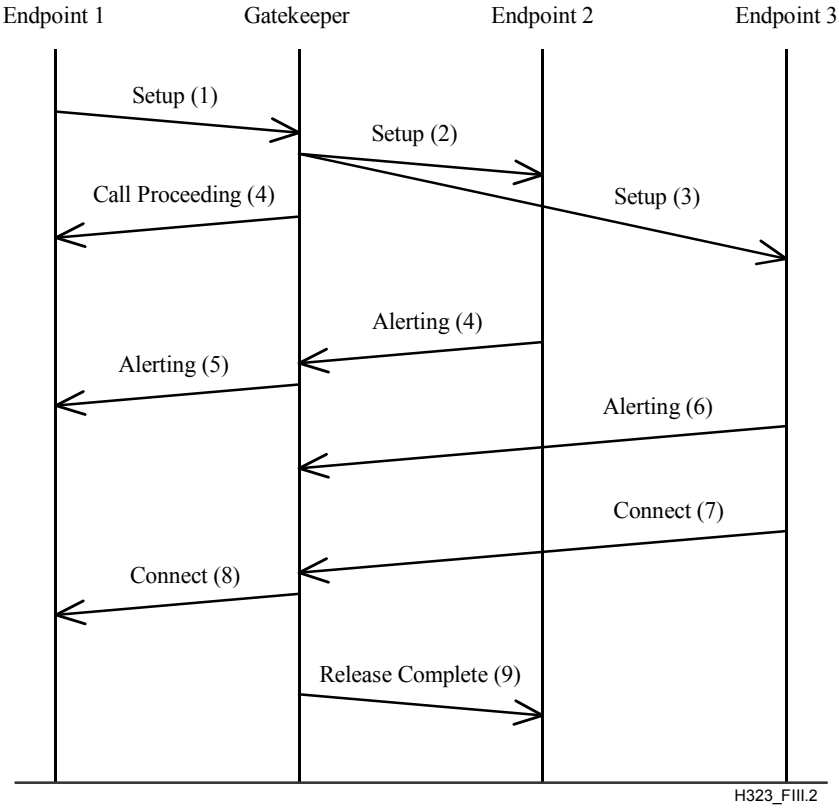


Figure III.2/H.323 – Example of user location using H.225.0 call signalling (RAS signalling not shown for clarity)

Note that if the Gatekeeper is performing this type of user location algorithm, it should not pass the **h245Address** field in any of the Setup Acknowledge, Call Proceeding, and Alerting messages from Endpoint 2 or Endpoint 3 to Endpoint 1 as this may give the wrong result.

Appendix IV

Signalling prioritized alternative logical channels in H.245

IV.1 Introduction

This appendix describes a simple method by which alternative logical channels may be signalled. No coding or semantic changes are required.

This method depends upon the guaranteed ordered delivery that is provided by TCP and is consequently equally applicable to both tunnelled and non-tunnelled H.245 signalling. Tunnelled signalling further depends upon guaranteed processing order where multiple H.245 messages are tunnelled in a single H.225.0 call signalling message.

IV.2 Signalling

All alternative logical channels are identified by the use of a common **forwardLogicalChannelNumber** in **openLogicalChannel** messages, one alternative per message. Messages may be sent either via the H.245 tunnel (one or more OLC messages per call signalling message) or via separate H.245 connection. Alternative logical channels are signalled in order of decreasing desirability, i.e., the first OLC message specifies the **dataType** that the sender of the OLC would prefer to use on the logical channel.

The receiver of these OLC messages is not required to be aware that this method of alternative propositions is being used. Prior to reception of an acceptable OLC request, it will reject unacceptable OLC requests, typically with a cause code of **dataTypeNotSupported**, **dataTypeNotAvailable**, or **unknownDataType**. When an acceptable OLC request is received, the endpoint will respond with an **openLogicalChannelAck** message. Any subsequently received alternative OLCs are rejected by the receiver with a cause code of **unspecified**, as the requested logical channel number will map to a currently open channel.

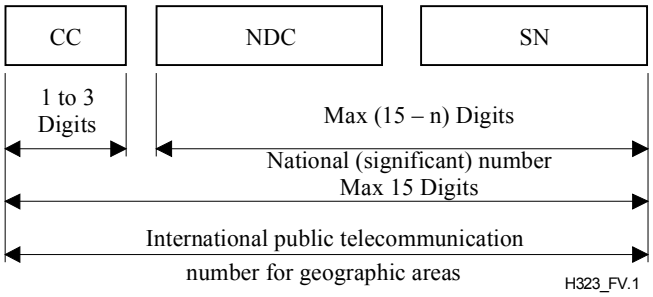
The sender of such a prioritized sequence of **openLogicalChannel** messages must keep track of the number of OLC reject messages received prior to reception of an **openLogicalChannelAck** message in order to determine which proposed alternative was accepted by the peer.

Appendix V

Use of E.164 and ISO/IEC 11571 numbering plans

V.1 E.164 numbering plan

ITU-T defines E.164 numbers the following way for geographic areas (see Figure V.1):



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

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

Figure V.1/H.323 – International public telecommunication number structure for geographic areas

Similar descriptions are also defined for non-geographic areas. ITU-T Rec. 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 dialling 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 "dialled 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 dialling plan takes over. Each telephone company or private network is free to choose its own dialling 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 dialling 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 dialling automatically, the user is usually required to customize the dialling software for every region or network.

Because of these issues with varying dialling plans and automated dialling, 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 dialled digits: for example, a PBX may gather the dialled 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 (ITU-T Rec. E.164)" indicates an E.164 number. Specifying the type of number as "unknown" and specifying the numbering plan as "unknown" indicates dialled digits.

The following are a set of definitions from ITU-T Rec. E.164:

V.1.1 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.

V.1.2 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.

V.1.3 dialling plan: A string or combination of decimal digits, symbols, and additional information that define the method by which the numbering plan is used. A dialling plan includes the use of prefixes, suffixes, and additional information, supplemental to the numbering plan, required to complete the call.

V.1.4 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).

V.1.5 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.

V.1.6 international prefix: A digit or combination of digits used to indicate that the number following is an International Public Telecommunication Number.

V.1.7 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.

V.1.8 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.

V.1.9 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.

V.1.10 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.

V.1.11 subscriber number (SN): The number identifying a subscriber in a network or numbering area.

V.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 Numbering Plan (PNP) number 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 Rec. E.164 and Figure V.2.

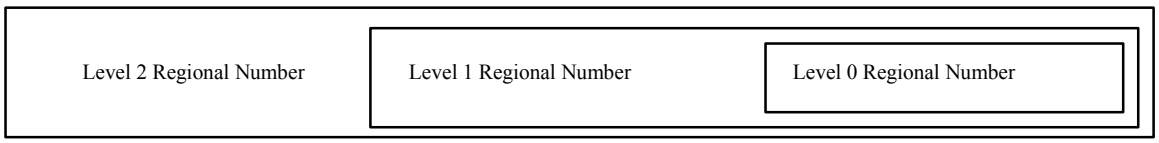


Figure V.2/H.323 – Structure of a PNP Number with three levels of regions

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 dialling 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:

V.2.1 private numbering plan (PNP): The numbering plan explicitly relating to a particular private numbering domain, defined by the PISN Administrator of that domain.

V.2.2 PNP number: A number belonging to a PNP.

V.2.3 region: The entire domain or a sub-domain of a PNP. A region does not necessarily correspond to a geographical area of a PISN.

V.2.4 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.

V.2.5 regional number (RN): A particular form of a PNP Number which is unambiguous in the region concerned.

V.2.6 complete number: A number which is unambiguous in the entire PNP, i.e., which corresponds to the highest regional level employed in that PISN.

V.3 H.323 versions 1, 2 and 3 usage

H.323 versions 1, 2 and 3 systems had a terminology problem with respect to dialled digits and real E.164 numbers. References to E.164 addresses in those versions actually referred to dialled digits and not E.164 digits, as the names of the fields implied. In H.323 versions 2 and 3 systems, a real E.164 number was placed in the **publicNumber** field and not in the **e164** field. The **e164** field thus corresponded to a dialled digits sequence.

Beginning with H.323 Version 4 systems, the field **e164** was renamed to **dialledDigits** and the field **publicNumber** was renamed to **e164Number**. The name change was intended to more explicitly convey that dialled digits shall be stored in the **dialledDigits** field and that E.164 numbers shall be stored in the **e164Number** field.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2004