INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.323
Annex E
(05/99)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

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

---

Packet-based multimedia communications systems

Annex E: Framework and wire-protocol for multiplexed call signalling transport

ITU-T Recommendation H.323 – Annex E

(Previously CCITT Recommendation)

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

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

ITU-T RECOMMENDATION H.323

PACKET-BASED MULTIMEDIA COMMUNICATIONS SYSTEMS

ANNEX E

Framework and wire-protocol for multiplexed call signalling transport

Summary

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 subclauses detail specific use cases.

Source

Annex E to ITU-T Recommendation H.323, was prepared by ITU-T Study Group 16 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on 27 May 1999.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The 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 Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 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 term *recognized operating agency (ROA)* includes any individual, company, corporation or governmental organization that operates a public correspondence service. The terms *Administration, ROA* and *public correspondence* are defined in the *Constitution of the ITU (Geneva, 1992)*.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The 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, the ITU had not 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

Recommendation H.323

PACKET-BASED MULTIMEDIA COMMUNICATIONS SYSTEMS

ANNEX E

Framework and wire-protocol for multiplexed call signalling transport

(Geneva, 1999)

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 subclauses detail specific use cases. The only profile currently specified in this revision is for transporting H.225.0 Q.931-like messages.

This annex is designed to operate in engineered networks and use the security services provided by H.323 (e.g. H.235, IP-SEC). 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 on-going 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

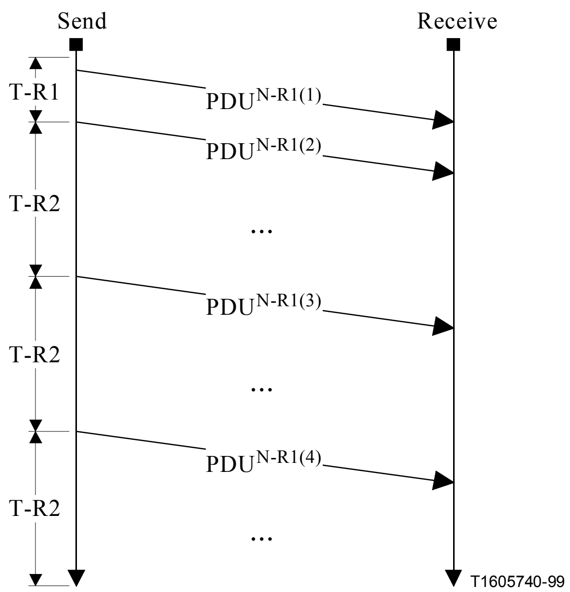
Assigned per host-address + source-port, sending applications shall start with some random value, incrementing by 1 for every PDU sent. If the sequence number reaches 2^{24} (16 777 216) it shall wrap around to 0.

E.1.1.7 Receiver sequence number policy

When receiving a UDP packet, the application shall check the host-address + source-port + sequence number to recognize duplicate messages. The application may re-order 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 re-transmitting quickly, but if that message is lost too, the sender is required to back-off the retransmission delay by a factor of more than two.



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 request/reply interval value from a previous transmission, timer T-R1 should be set to the value +10%.

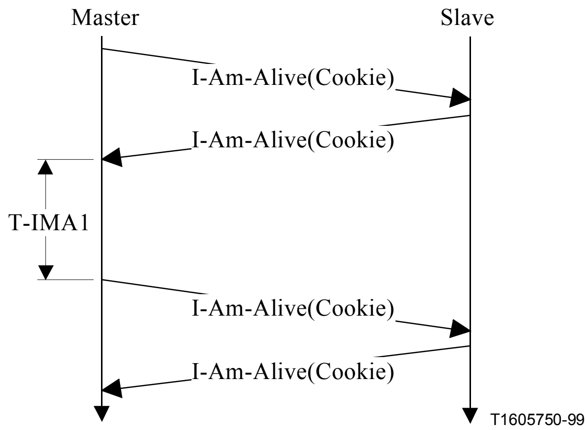
E.1.1.9 Connection keep-alive

When running over TCP, the presence of a persistent TCP connection can insure 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).



I-Am-Alive timers

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

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 application 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 insure 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 application (or rather the Annex E stack) 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. 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-bit long.

NOTE – This field may be used for example to carry the CRV (e.g. Call-Reference-Value as defined in 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-bit long.

NOTE – This field may be used for example in 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.

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="3">VERSION</td><td>6</td><td>M</td><td>H</td><td>L</td><td>A</td><td colspan="24">SEQNUM</td></tr>
<tr><td colspan="32">PAYLOAD(s) 0, ..., N–1</td></tr>
</table>

T1607770-00

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

0					1	2	3
0 1 2 3 4 5 6 7					0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
VERSION	6	M	H	L	A	SEQNUM	
PAYLOAD COUNT					LENGTH		
PAYLOAD(s) 0, ..., N–1							

T1607780-00

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.

0			
0 1	2	3	4 5 6 7
T	S	A	R

T1607790-00

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.

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.

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="2">T</td><td>S</td><td>A</td><td colspan="4">R</td><td colspan="8">TRANSPORT MESSAGE = 0</td><td colspan="16">VALIDITY</td></tr>
<tr><td colspan="15">COOKIE LENGTH</td><td>P</td><td colspan="8">COOKIE OCTET 0</td><td colspan="8">COOKIE OCTET N–1</td></tr>
</table>

T1607800-00

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	$8 \times n$

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.

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="2">T</td><td>S</td><td>A</td><td colspan="4">R</td><td colspan="8">TRANSPORT MESSAGE = 1</td><td colspan="16">ACK COUNT</td></tr>
<tr><td colspan="24">SEQNUM 0</td><td colspan="8">RESERVED</td></tr>
<tr><td colspan="24">SEQNUM N–1</td><td colspan="8">RESERVED</td></tr>
</table>

T1607810-00

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.

<table>
<tr><td colspan="4">0
0 1 2 3 4 5 6 7</td><td>1
0 1 2 3 4 5 6 7</td><td>2
0 1 2 3 4 5 6 7</td><td>3
0 1 2 3 4 5 6 7</td></tr>
<tr><td>T</td><td>S</td><td>A</td><td>R</td><td>TRANSPORT MESSAGE = 2</td><td colspan="2">NACK COUNT</td></tr>
<tr><td colspan="6">SEQNUM 0</td><td>DATA LENGTH</td></tr>
<tr><td colspan="5">REASON 0</td><td>DATA OCTET 0</td><td>DATA OCTET N–1</td></tr>
<tr><td colspan="6">SEQNUM N–1</td><td>DATA LENGTH</td></tr>
<tr><td colspan="5">REASON N–1</td><td>DATA OCTET 0</td><td>DATA OCTET N–1</td></tr>
</table>

T1607820-00

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	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

0 (bits 0–7)	1 (bits 0–7)	2 (bits 0–7)	3 (bits 0–7)
PAYLOAD TYPE	Reserved	ALTERNATE PORT	
ALTERNATE IP ADDRESS			

T1607830-00

Figure E.10/H.323 – Nack reason 1 structure

0 (bits 0–7)	1 (bits 0–7)	2 (bits 0–7)	3 (bits 0–7)
OID LENGTH	OID OCTET 0	OID OCTET 1	OID OCTET N–1
ALTERNATE IP ADDRESS			
ALTERNATE PORT			

T1607840-00

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 it has restarted, and that all active calls have been disconnected. Any message arriving from the previous sequence-number range shall be considered stale and ignored. All outstanding calls that were related to the state of the system before the restart will be dropped.

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

0 (bits 0–1)	0 (bit 2)	0 (bit 3)	0 (bits 4–7)	1 (bits 0–7)	2 (bits 0–7)	3 (bits 0–7)
T = 00	S	A	RESERVED	TRANSPORT MESSAGE = 4	RESERVED	

T1607850-00

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:

Value	Interpretation
0	Octet-stream contains a Q.931 message as defined in Recommendation 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:

<table>
<tr><td colspan="4">0</td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">3</td></tr>
<tr><td>0 1</td><td>2</td><td>3</td><td>4 5 6 7</td><td colspan="2">0 1 2 3 4 5 6 7</td><td colspan="2">0 1 2 3 4 5 6 7</td><td colspan="2">0 1 2 3 4 5 6 7</td></tr>
<tr><td>T</td><td>S</td><td>A</td><td>R</td><td colspan="2">TYPE</td><td colspan="4">PAYLOAD LENGTH</td></tr>
<tr><td colspan="4">PAYLOAD OCTET 0</td><td colspan="2">PAYLOAD OCTET 1</td><td colspan="2">PAYLOAD OCTET 2</td><td colspan="2">PAYLOAD OCTET N–1</td></tr>
</table>

T1607860-00

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	$8 \times n$

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.

<table>
<tr><td colspan="4">0</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>0 1</td><td>2</td><td>3</td><td>4 5 6 7</td><td>0 1 2 3 4 5 6 7</td><td>0 1 2 3 4 5 6 7</td><td>0 1 2 3 4 5 6 7</td></tr>
<tr><td>T</td><td>S</td><td>A</td><td>R</td><td>TYPE</td><td colspan="2">SESSION</td></tr>
<tr><td colspan="5">PAYLOAD LENGTH</td><td>PAYLOAD OCTET 0</td><td>PAYLOAD OCTET 1</td></tr>
<tr><td colspan="4">PAYLOAD OCTET 2</td><td>PAYLOAD OCTET 3</td><td>PAYLOAD OCTET 4</td><td>PAYLOAD OCTET N–1</td></tr>
</table>

T1607870-00

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)	$8 \times n$

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.

0 0 1 2 3 4 5 6 7	1 0 1 2 3 4 5 6 7	2 0 1 2 3 4 5 6 7	3 0 1 2 3 4 5 6 7
T | S | A | R	TYPE	SESSION	
SOURCE/DESTINATION ADDRESS			
PAYLOAD LENGTH		PAYLOAD OCTET 0	PAYLOAD OCTET 1
PAYLOAD OCTET 2	PAYLOAD OCTET 3	PAYLOAD OCTET 4	PAYLOAD OCTET N–1

T1607880-00

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.

0 0 1 2 3 4 5 6 7	1 0 1 2 3 4 5 6 7	2 0 1 2 3 4 5 6 7	3 0 1 2 3 4 5 6 7
T | S | A | R	TYPE	PAYLOAD LENGTH	
SOURCE/DESTINATION ADDRESS			
PAYLOAD OCTET 0	PAYLOAD OCTET 1	PAYLOAD OCTET 2	PAYLOAD OCTET N–1

T1607890-00

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:

0	1	2	3
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
T | S | N | R	OID LENGTH	OID OCTET 0	OID OCTET N–1
PAYLOAD LENGTH		PAYLOAD OCTET 0	PAYLOAD OCTET N–1

T1607900-00

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.

0	1	2	3
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
T | S | N | R	OID LENGTH	OID OCTET 0	OID OCTET N–1
SESSION		PAYLOAD LENGTH	
PAYLOAD OCTET 0	PAYLOAD OCTET 1	PAYLOAD OCTET 2	PAYLOAD OCTET N–1

T1607910-00

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.

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="2">T</td><td>S</td><td>N</td><td colspan="4">R</td><td colspan="8">OID LENGTH</td><td colspan="8">OID OCTET 0</td><td colspan="8">OID OCTET N–1</td></tr>
<tr><td colspan="16">SESSION</td><td colspan="16">PAYLOAD LENGTH</td></tr>
<tr><td colspan="32">SOURCE/DESTINATION ADDRESS</td></tr>
<tr><td colspan="8">PAYLOAD OCTET 0</td><td colspan="8">PAYLOAD OCTET 1</td><td colspan="8">PAYLOAD OCTET 2</td><td colspan="8">PAYLOAD OCTET N–1</td></tr>
</table>

T1607920-00

Field	Content of fields	Bits
OID LENGTH	Unsigned Integer: the length in OCTETs of the Object Identifier following	8
ObjectID	Object Identifier OCTETs	$8 \times n$
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	$8 \times n$

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.

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="2">T</td><td>S</td><td>N</td><td colspan="4">R</td><td colspan="8">OID LENGTH</td><td colspan="8">OID OCTET 0</td><td colspan="8">OID OCTET N–1</td></tr>
<tr><td colspan="32">SOURCE/DESTINATION ADDRESS</td></tr>
<tr><td colspan="16">PAYLOAD LENGTH</td><td colspan="8">PAYLOAD OCTET 0</td><td colspan="8">PAYLOAD OCTET 1</td></tr>
<tr><td colspan="8">PAYLOAD OCTET 2</td><td colspan="8">PAYLOAD OCTET 3</td><td colspan="8">PAYLOAD OCTET 4</td><td colspan="8">PAYLOAD OCTET N–1</td></tr>
</table>

T1607930-00

Figure E.21/H.323 – Extended-3 ObjectID-typed payload format

E.2 H.225.0 Call Signalling over Annex E

This subclause 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.

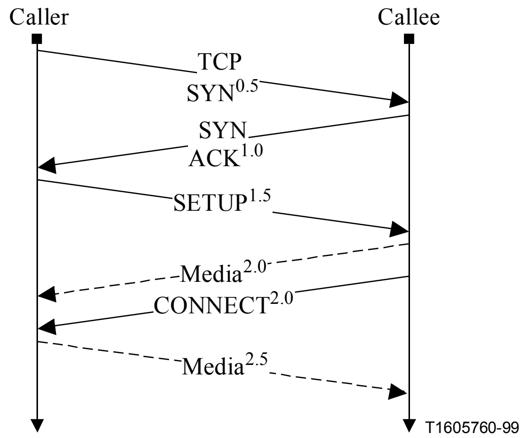
E.2.1 Rational

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

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.



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:

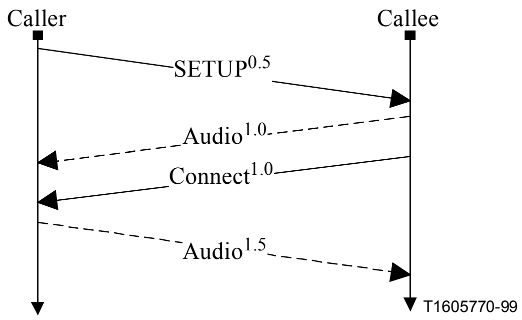


Figure E.23/H.323 – Information flow for UDP-based Call-Setup

Applications 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 subclause 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 re-ordered in delivery. The caller is notified of the selection according to the transport protocol over which the subsequent message (e.g. Connect) has arrived.

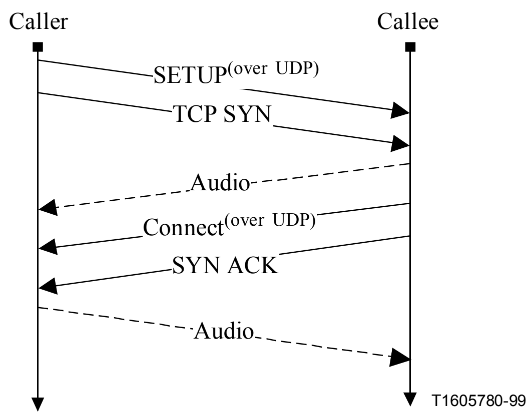


Figure E.24/H.323 – Information flow for mixed TCP and UDP procedure

This insures that if the UDP procedure fails, usual TCP-based procedures can take over immediately:

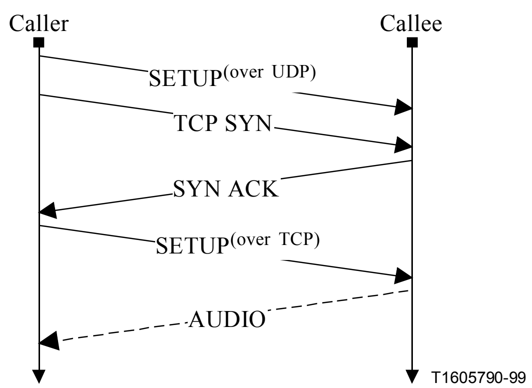


Figure E.25/H.323 – Information flow when UDP is not supported

This means that backwards compatibility when calling H.323 version 1 (1996) or 2 (1998) entities is transparent, as the v1/v2 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.

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 Q.931 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 then path MTU.

E.2.3.8 H.245

H.245 shall be transmitted using the H.323 version 2 (1998) H.245 Tunnelling procedures.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the 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	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