INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.245

(02/2003)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Communication procedures

Control protocol for multimedia communication

ITU-T Recommendation H.245

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

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

ITU-T Recommendation H.245

Control protocol for multimedia communication

Summary

This Recommendation specifies syntax and semantics of terminal information messages as well as procedures to use them for in-band negotiation at the start of or during communication. The messages cover receiving and transmitting capabilities as well as mode preference from the receiving end, logical channel signalling, and Control & Indication. Acknowledged signalling procedures are specified to ensure reliable audiovisual and data communication.

Products claiming compliance with Version 9 of H.245 shall comply with all of the mandatory requirements of this Recommendation. Version 9 products can be identified by H.245 TerminalCapabilitySet messages containing a protocolIdentifier value of {itu-t (0) recommendation (0) h (8) 245 version (0) 9}.

In view of the coming publication of version 10 of this Recommendation, version 9 is published showing only the differences relative to version 8.

Source

ITU-T Recommendation H.245 was prepared by ITU-T Study Group 16 (2001-2004) and approved under the WTSA Resolution 1 procedure on 6 February 2003.

History

Version		Approval
1.0	H.245	1996-03-20
2.0	H.245	1997-07-11
3.0	H.245	1998-02-06
4.0	H.245	1998-09-25
5.0	H.245	1999-05-27
6.0	H.245	2000-02-17
7.0	H.245	2000-11-17
8.0	H.245	2001-07-29
9.0	H.245	2003-02-06

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.

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.

ITU-T Recommendation H.245

Control protocol for multimedia communication

In view of the coming publication of version 10 of this Recommendation, this version 9 is published showing only the differences relative to version 8.

The differences are as enumerated below.

1) *In Annex A*, *the* `TerminalCapabilitySet` *structure is modified as follows:*

```
TerminalCapabilitySet       ::=SEQUENCE
{
     sequenceNumber         SequenceNumber,

     protocolIdentifier     OBJECT IDENTIFIER,
                            -- shall be set to the value
                            -- {itu-t (0) recommendation (0) h (8) 245
                            -- version (0) 9~~8~~}


     multiplexCapability    MultiplexCapability OPTIONAL,

     capabilityTable        SET SIZE (1..256) OF CapabilityTableEntry OPTIONAL,

     capabilityDescriptors  SET SIZE (1..256) OF CapabilityDescriptor OPTIONAL,
     ...
}
```

2) *In Annex A, the* `Capability` *structure is modified as follows:*

```
Capability                                         ::=CHOICE
{
     nonStandard                                   NonStandardParameter,

     receiveVideoCapability                        VideoCapability,
     transmitVideoCapability                       VideoCapability,
     receiveAndTransmitVideoCapability             VideoCapability,

     receiveAudioCapability                        AudioCapability,
     transmitAudioCapability                       AudioCapability,
     receiveAndTransmitAudioCapability             AudioCapability,
     receiveDataApplicationCapability              DataApplicationCapability,
     transmitDataApplicationCapability             DataApplicationCapability,
     receiveAndTransmitDataApplicationCapability   DataApplicationCapability,

     h233EncryptionTransmitCapability              BOOLEAN,
     h233EncryptionReceiveCapability               SEQUENCE
     {
       h233IVResponseTime                          INTEGER (0..255),
                                                   -- units milliseconds
       ...
     },
     ...,
     conferenceCapability                          ConferenceCapability,
     h235SecurityCapability                        H235SecurityCapability,
     maxPendingReplacementFor                      INTEGER (0..255),
     receiveUserInputCapability                    UserInputCapability,
     transmitUserInputCapability                   UserInputCapability,
     receiveAndTransmitUserInputCapability         UserInputCapability,
    genericControlCapability                  GenericCapability,
    receiveMultiplexedStreamCapability        MultiplexedStreamCapability,
    transmitMultiplexedStreamCapability       MultiplexedStreamCapability,
    receiveAndTransmitMultiplexedStreamCapability MultiplexedStreamCapability,
    receiveRTPAudioTelephonyEventCapability   AudioTelephonyEventCapability,
    receiveRTPAudioToneCapability             AudioToneCapability,
    fecCapability                             FECCapability,
    multiplePayloadStreamCapability           MultiplePayloadStreamCapability
}
```

3) *In Annex A, in the* `RefPictureSelection` *structure, a final closing bracket is added as follows:*

```
RefPictureSelection                        ::=SEQUENCE
{
    additionalPictureMemory                SEQUENCE
    {
      sqcifAdditionalPictureMemory         INTEGER (1..256) OPTIONAL,
                                           -- units frame
      qcifAdditionalPictureMemory          INTEGER (1..256) OPTIONAL,
                                           -- units frame
      cifAdditionalPictureMemory           INTEGER (1..256) OPTIONAL,
                                           -- units frame
      cif4AdditionalPictureMemory          INTEGER (1..256) OPTIONAL,
                                           -- units frame
      cif16AdditionalPictureMemory         INTEGER (1..256) OPTIONAL,
                                           -- units frame
      bigCpfAdditionalPictureMemory        INTEGER (1..256) OPTIONAL,
                                           -- units frame
      ...
    } OPTIONAL,
    videoMux                               BOOLEAN,
    videoBackChannelSend                   CHOICE
    {
      none                                 NULL,
      ackMessageOnly                       NULL,
      nackMessageOnly                      NULL,
      ackOrNackMessageOnly                 NULL,
      ackAndNackMessage                    NULL,
      ...
    },
    ...,
    enhancedReferencePicSelect             SEQUENCE
    {
      subPictureRemovalParameters          SEQUENCE
      {
         mpuHorizMBs                       INTEGER (1..128),
         mpuVertMBs                        INTEGER (1..72),
         mpuTotalNumber                    INTEGER (1..65536),
         ...
      } OPTIONAL,
      ...
    }
}
```

4) *In Annex A, the* `AudioCapability` *structure is modified as follows:*

```
AudioCapability                             ::=CHOICE
{
    nonStandard                             NonStandardParameter,
    g711Alaw64k                             INTEGER (1..256),
    g711Alaw56k                             INTEGER (1..256),
    g711Ulaw64k                             INTEGER (1..256),
    g711Ulaw56k                             INTEGER (1..256),

    g722-64k                                INTEGER (1..256),
    g722-56k                                INTEGER (1..256),
    g722-48k                                INTEGER (1..256),

    g7231                                   SEQUENCE
    {
      maxAl-sduAudioFrames                  INTEGER (1..256),
      silenceSuppression                    BOOLEAN
    },

    g728                                    INTEGER (1..256),
    g729                                    INTEGER (1..256),
    g729AnnexA                              INTEGER (1..256),
    is11172AudioCapability                  IS11172AudioCapability,
    is13818AudioCapability                  IS13818AudioCapability,
     ...,
    g729wAnnexB                             INTEGER(1..256),
    g729AnnexAwAnnexB                       INTEGER(1..256),
    g7231AnnexCCapability                   G7231AnnexCCapability,
    gsmFullRate                             GSMAudioCapability,
    gsmHalfRate                             GSMAudioCapability,
    gsmEnhancedFullRate                     GSMAudioCapability,
    genericAudioCapability                  GenericCapability,
    g729Extensions                          G729Extensions,
    vbd                                     VBDCapability,
    audioTelephonyEvent                     NoPTAudioTelephonyEventCapability,
    audioTone                               NoPTAudioToneCapability
}
```

5) *In Annex A, after* `GSMAudioCapability`*, a new structure,* `VDBCapability`*, is introduced as follows:*

```
VBDCapability                               ::=SEQUENCE
{
    type                                    AudioCapability,
                                            -- shall not be "vbd"
    ...
}
```

6) *In Annex A, after* `AudioToneCapability`*, the following structures are introduced:*

```
-- The following definitions are as above but without a Payload Type field.

NoPTAudioTelephonyEventCapability           ::=SEQUENCE
{
    audioTelephoneEvent                     GeneralString,-- As per <list of
                                            -- values> in 3.9/RFC 2833
    ...
}

NoPTAudioToneCapability                     ::=SEQUENCE
{
    ...
}
```

```
-- =============================================================================
-- Capability Exchange Definitions: MultiplePayloadStreamCapability
-- =============================================================================

MultiplePayloadStreamCapability          ::=SEQUENCE
{
    capabilities                         SET SIZE(1..256) OF
                                         AlternativeCapabilitySet,
    ...
}

-- =============================================================================
-- Capability Exchange Definitions: FECCapability
-- =============================================================================

FECCapability                            ::=CHOICE
{
    rfc2733                              SEQUENCE
    {
      redundancyEncoding                 BOOLEAN,
      separateStream                     SEQUENCE
      {
         separatePort                    BOOLEAN,
         samePort                        BOOLEAN,
         ...
      },
      ...
    },
    ...
}
```

7) *In Annex A, the* `DataType` *structure is modified as follows:*

```
DataType                                 ::=CHOICE
{
    nonStandard                          NonStandardParameter,
    nullData                             NULL,
    videoData                            VideoCapability,
    audioData                            AudioCapability,
    data                                 DataApplicationCapability,
    encryptionData                       EncryptionMode,
    ...,
    h235Control                          NonStandardParameter,
    h235Media                            H235Media,
    multiplexedStream                    MultiplexedStreamParameter,
    redundancyEncoding                   RedundancyEncoding,
    multiplePayloadStream                MultiplePayloadStream,
    fec                                  FECData
}
```

8) *In Annex A, the* `H235Media` *structure is modified as follows:*

```
H235Media                                ::=SEQUENCE
{
    encryptionAuthenticationAndIntegrity EncryptionAuthenticationAndIntegrity,

    mediaType                            CHOICE
    {
      nonStandard                        NonStandardParameter,
      videoData                          VideoCapability,
      audioData                          AudioCapability,
      data                               DataApplicationCapability,
      ...,
       redundancyEncoding                  RedundancyEncoding,
       multiplePayloadStream               MultiplePayloadStream,
       fec                                 FECData
     },

     ...
}
```

9) *In Annex A, the* `RedundancyEncoding` *structure is modified and new structures are introduced as follows:*

```
RedundancyEncoding               ::=SEQUENCE
{
     redundancyEncodingMethod    RedundancyEncodingMethod,
     secondaryEncoding           DataType OPTIONAL,-- depends on method
     ...,

     -- The sequence below may be used in place of the above secondaryEncoding
     -- field

     rtpRedundancyEncoding       SEQUENCE
     {
       primary                   RedundancyEncodingElement OPTIONAL,
                                 -- Present when redundancyEncoding
                                 -- is selected as the dataType
                                 -- in an OpenLogicalChannel or
                                 -- as part of a MultiplePayloadSteam

       secondary                 SEQUENCE OF RedundancyEncodingElement OPTIONAL,
       ...
     } OPTIONAL
}

RedundancyEncodingElement        ::=SEQUENCE
{
     dataType                    DataType,
     payloadType                 INTEGER(0..127) OPTIONAL,
     ...
}

MultiplePayloadStream            ::=SEQUENCE
{
     elements                    SEQUENCE OF MultiplePayloadStreamElement,
     ...
}

MultiplePayloadStreamElement     ::=SEQUENCE
{
     dataType                    DataType,
     payloadType                 INTEGER(0..127) OPTIONAL,
     ...
}

FECData          ::=CHOICE
{
     rfc2733     SEQUENCE
     {
       mode   CHOICE
       {
          redundancyEncoding     NULL,
          separateStream         CHOICE
          {
             differentPort      SEQUENCE
              {
                 protectedSessionID        INTEGER(1..255),
                 protectedPayloadType      INTEGER(0..127) OPTIONAL,
             ...
              },
              samePort          SEQUENCE
              {
                 protectedPayloadType      INTEGER(0..127),
                 ...
              },
              ...
          },
          ...
       },
       ...
     }
}
```

10) *In Annex A, the* `ModeElement` *structure is modified and a new structure,* `ModeElementType` *is introduced as follows:*

```
ModeElementType                        ::=CHOICE
{
     nonStandard                       NonStandardParameter,
     videoMode                         VideoMode,
     audioMode                         AudioMode,
     dataMode                          DataMode,
     encryptionMode                    EncryptionMode,
     ...,
     h235Mode                          H235Mode,
     multiplexedStreamMode             MultiplexedStreamParameter,
     redundancyEncodingDTMode          RedundancyEncodingDTMode,
     multiplePayloadStreamMode         MultiplePayloadStreamMode,
     fecMode                           FECMode
}

ModeElement                            ::= SEQUENCE
{
     type                              ModeElementType,~~CHOICE~~
     ~~{~~
          ~~nonStandared                 NonStandardParameter,~~
          ~~videoMode                    VideoMode,~~
          ~~audioMode                    AudioMode,~~
          ~~dataMode                     EncryptionMode,~~
          ~~...,~~
          ~~h223Mode                     H235Mode,~~
          ~~multiplexedStreamMode        MultiplexedStreamParameter~~
     ~~},~~
     h223ModeParameters                H223ModeParameters OPTIONAL,
     ...,
     v76ModeParameters                 V76ModeParameters OPTIONAL,
     h2250ModeParameters               H2250ModeParameters OPTIONAL,
     genericModeParameters             GenericCapability OPTIONAL,
     multiplexedStreamModeParameters   MultiplexedStreamModeParameters OPTIONAL
}
```

11) *In Annex A, new structures after* `MultiplexedStreamModeParameters` *are introduced as follows:*

```
RedundancyEncodingDTMode        ::=SEQUENCE
{
     redundancyEncodingMethod   RedundancyEncodingMethod,
     primary                    RedundancyEncodingDTModeElement,
     secondary                   SEQUENCE OF RedundancyEncodingDTModeElement,
     ...
}

RedundancyEncodingDTModeElement  ::=SEQUENCE
{
     type                        CHOICE
     {
         nonStandard             NonStandardParameter,
         videoMode               VideoMode,
         audioMode               AudioMode,
         dataMode                DataMode,
         encryptionMode          EncryptionMode,
         h235Mode                H235Mode,
         ...
     },
     ...
}

MultiplePayloadStreamMode        ::=SEQUENCE
{
     elements                    SEQUENCE OF MultiplePayloadStreamElementMode,
     ...
}

MultiplePayloadStreamElementMode      ::=SEQUENCE
{
     type                             ModeElementType,
     ...
}

FECMode                               ::=CHOICE
{
     rfc2733Mode                      SEQUENCE
     {
         mode                         CHOICE
         {
             redundancyEncoding       NULL,
             separateStream           CHOICE
             {
                 differentPort        SEQUENCE
                 {
                   protectedSessionID INTEGER(1..255),
                   protectedPayloadType INTEGER(0..127) OPTIONAL,
                   ...
                 },
                 samePort             SEQUENCE
                 {
                   protectedType      ModeElementType,
                   ...
             },
             ...
         },
         ...
       },
       ...
     },
     ...
}
```

12) *In Annex A, the* `H263VideoMode` *structure is modified as follows:*

```
H263VideoMode                       ::=SEQUENCE
{
    resolution                      CHOICE
    {
      sqcif                         NULL,
      qcif                          NULL,
      cif                           NULL,
      cif4                          NULL,
      cif16                         NULL,
      ...,
      custom                        NULL
    },
    bitRate  INTEGER (1..19200),    -- units 100 bit/s
    unrestrictedVector              BOOLEAN,
    arithmeticCoding                BOOLEAN,
    advancedPrediction              BOOLEAN,
    pbFrames BOOLEAN,
    ...,

    errorCompensation               BOOLEAN,
    enhancementLayerInfo            EnhancementLayerInfo OPTIONAL,
    h263Options                     H263Options OPTIONAL
}
```

13) *In Annex A, the* `AudioMode` *structure is modified as follows:*

```
AudioMode                           ::=CHOICE
{
    nonStandard                     NonStandardParameter,
    g711Alaw64k                     NULL,
    g711Alaw56k                     NULL,
    g711Ulaw64k                     NULL,
    g711Ulaw56k                     NULL,

    g722-64k                        NULL,
    g722-56k                        NULL,
    g722-48k                        NULL,

    g728                            NULL,
    g729                            NULL,
    g729AnnexA                      NULL,

    g7231                           CHOICE
    {
      noSilenceSuppressionLowRate   NULL,
      noSilenceSuppressionHighRate  NULL,
      silenceSuppressionLowRate     NULL,
      silenceSuppressionHighRate    NULL
    },

    is11172AudioMode                IS11172AudioMode,
    is13818AudioMode                IS13818AudioMode,

    ...,
    g729wAnnexB                     INTEGER(1..256),
    g729AnnexAwAnnexB               INTEGER(1..256),
    g7231AnnexCMode                 G7231AnnexCMode,
    gsmFullRate                     GSMAudioCapability,
    gsmHalfRate                     GSMAudioCapability,
    gsmEnhancedFullRate             GSMAudioCapability,
    genericAudioMode                GenericCapability,
    g729Extensions                  G729Extensions,
    vbd                             VBDMode
}
```

14) *In Annex A, after the* `G7231AnnexCMode`*, a new structure,* `VBDMode`*, is modified as follows:*

```
VBDMode                             ::=SEQUENCE
{
    type                            AudioMode,-- shall not be "vbd"
    ...
}
```

15) *In Annex B, new clauses B.2.2.14 and B.2.2.15 are added as follows:*

B.2.2.14 Multiple Payload Stream

A multiple payload stream (MPS) contains packets representing a single logical media stream; that is, the packets all represent encodings of that same stream for specified time intervals. To allow identification and correlation of the various encodings used, all packets in a single MPS SHALL carry payload type identifiers in the same location in the packet and SHOULD use timestamps in the same format and derived from a single clock source (e.g., RTP payloads should use the same SSRC). In most cases these packets will represent sequential, non-overlapping time intervals and simply choose distinct encodings for distinct intervals, but there are cases where alternate encodings represent overlapping intervals, such as, when an event occurs in the middle of an encoding interval that must be encoded distinctly in the alternate encoding. This may occur, for example, when a DTMF tone is detected in the middle of a voice-encoding interval and should be sent using RFC 2833 telephone-event. In this case the timestamp in the telephone-event packet will correspond to a time in the middle of the voice-encoding interval. Packets with zero duration may be used where the stream event represented has no measurable duration. It is also permissible to use RFC 2198 to send a packet multiple times, packed into a packet with other payload types and time intervals.

NOTE – Since all packets must represent encodings of a single source (destination) stream it is not appropriate to include distinct media types, such as audio and video, although data-type packets representing data derived from the media stream (such as DTMF digits detected in an audio stream) may be an alternate representation or encoding and are appropriate.

B.2.2.15 Forward Error Correction

An endpoint may advertise the ability to perform Forward Error Correction. When advertising RFC 2733, the endpoint has the ability to signal that FEC data may be sent on a separate stream or the same stream (using redundant encoding), as per RFC 2198. This capability allows the endpoint to indicate (by capability table entry number) which codecs may be used in an FEC stream.

If the endpoint sending **OpenLogicalChannel** wishes to use RFC 2198 (and that capability is supported by the recipient) for carrying the FEC data, it shall use the **DataType redundancyEncoding**, including the VBD encoding, for example, as the **primary** encoding and the **DataType fec** as a **secondary** encoding. The payload type for the RFC 2198 packets shall be specified in the **dynamicPayloadType** field of the **OpenLogicalChannel**. The payload type for the **primary** encoding and the FEC data may be signaled in the **payloadType** field of the **primary** and **secondary RedundancyEncodingElement** fields.

If an endpoint wishes to transmit FEC data on a separate stream, it has two choices: to transmit to the same port as the FEC protected data or to a different port. When transmitting on a different port, it shall use a separate **OpenLogicalChannel** explicitly for the FEC stream. The **dataType** selected shall be **fec** and shall not be contained within a **redundancyEncoding** field. It shall select **mode.separateStream.differentPort** and include the session ID of the protected stream and, optionally, the payload type of the protected media, in the case that the subject channel carries multiple payload types, such as an MPS stream. When transmitting on a separate stream, but to the same port as the protected media, the FEC data shall be signalled as part of an MPS stream. In that case, one element of the MPS stream would be the protected audio and one element would be **fec**. In this case, it would select **mode.separateStream.samePort** and would advertise the payload type of the protected stream.

16) *In Annex B, clause B.3.1 is modified and new paragraphs between existing paragraphs are inserted as follows:*

B.3.1 Open Logical Channel

...

If it is nullData, the logical channel will not be used for the transport of elementary stream data, but only for adaptation layer information – if video is to be transmitted in one direction only, but a retransmission protocol is to be used, such as AL3 defined in ITU-T Rec. H.223, a return channel is needed to transport the retransmission requests – it may also be used to describe a logical channel that only contains PCR values in the case of H.222.1 Transport Streams [9].

A dataType of h235Media is used to specify encryption of the logical channel; the actual data type is indicated within H235Media, along with the encryption specification.

Terminals capable only of unidirectional (transmit or receive) operation on media types which make use of bidirectional channels shall send capabilities only for the supported direction of operation. The reverse direction shall use the nullData type, for which no capability is necessary. Transmit-only terminals should send transmit capabilities, but terminals should not assume that the absence of transmit capabilities implies that transmit-only operation is not possible.

...

redundancyEncoding indicates that the redundant encoding method indicated in this parameter is to be used for the logical channel to be opened. The primary encoding is defined by the *dataType* of the *forwardLogicalChannelParameters* or the *reverseLogicalChannelParameters*, respectively. The type of redundancy encoding to be applied for this logical channel is identified by the *redundancyEncodingMethod* parameter, the secondary encoding is specified in the *secondaryEncoding* parameter. The *DataType* (audio, video, etc.) selected for both primary and secondary encoding shall match and shall be in accordance with the *redundancyEncodingMethod* selected. The source parameter is used to identify the terminal number of the sender of the OpenLogicalChannel message.

The opening of a channel protected by redundancy, as specified in RFC 2198, is achieved using **dataType.redundancyEncoding**. This field allows signalling a primary data type and a number of **secondary** data types. It also makes it possible to use RFC 2198 with "multiple payload stream" and with Forward Error Correction.

When opening a logical channel, the RTP payload type for the RFC 2198 packet is specified by the **dynamicPayloadType** field in the **OpenLogicalChannel** or by the **payloadType** field inside the **multiplePayloadStreamElement** structure. The payload types for the primary and secondary payload types are specified in the **RedundancyEncodingElement** structure, along with the **DataType** of the primary or secondary data.

When RFC 2198 redundancy encoding is used, the **redundancyEncodingMethod** shall be set to **rtpRedundancyEncoding**. Also, when using RFC 2198 and populating the **RedundancyEncoding** SEQUENCE, only the **rtpRedundancyEncoding** SEQUENCE shall be used. The fields **RedundancyEncoding.secondaryEncoding** and **RedundancyEncoding.rtpRedundancyEncoding** shall not be used at the same time.

When encryption is specified for a channel carrying multiple payloads, redundancy encoding using RFC 2198 is used to preserve the actual payload types transmitted. The Encapsulating payload type is set to the value specified in the syncFlag field of the encryptionSync element.

h235 Key: is used to include, and specify the method by which media specific session keys are protected as they are passed between two endpoints. The encoding of this field is a nested ASN.1 value as described in ITU-T Rec. H.235.

...

17) *In Annex B, the fourth paragraph of clause B.6.1.1 is modified as follows:*

H263VideoMode: Indicates the requested picture resolution (SQCIF, QCIF, CIF, 4CIF and 16CIF or some custom picture format) and bit rate, in units of 100 bit/s. When communicating with an endpoint supporting H.245 version 8 or earlier, it is not possible to request only a custom picture format. Therefore, when receiving RequestMode from an endpoint supporting H.245 version 8 or earlier, if the RequestMode contains a custom picture format, this should be considered the requested resolution rather than the resolution indicated in the resolution field of H263VideoMode.

18) *In Annex D, Table D.1 is modified as follows:*

Table D.1/H.245

Object Identifier Value	Description
{itu-t (0) recommendation (0) h (8) 245 version (0) 1}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. This indicates the first version of this Recommendation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 2}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. At this time there are nine~~eight~~ standardized versions defined. This indicates the second version of this Recommendation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 3}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. At this time there are nine~~eight~~ standardized versions defined. This indicates the third version of this Recommendation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 4}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. At this time there are nine~~seven~~ standardized versions defined. This indicates the fourth version of this Recommendation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 5}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. At this time there are nine~~eight~~ standardized versions defined. This indicates the fifth version of this Recommendation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 6}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. At this time there are nine~~eight~~ standardized versions defined. This indicates the sixth version of this Recommendation.

Table D.1/H.245

Object Identifier Value	Description
{itu-t (0) recommendation (0) h (8) 245 version (0) 7}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. At this time there are nine~~eight~~ standardized versions defined. This indicates the seventh version of this Recommendation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 8}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. At this time there are nine~~eight~~ standardized versions defined. This indicates the eighth version of this Recommendation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 9}	This Object Identifier is used to indicate the version of this Recommendation in use as a multimedia system control protocol. At this time there are nine standardized versions defined. This indicates the ninth version of this Recommendation.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) video (0) ISO/IEC 14496-2 (0)}	This Object Identifier is used to indicate the generic capability for ISO/IEC 14496-2. This capability is defined in Annex E.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 14496-3 (0)}	This Object Identifier is used to indicate the generic capability for ISO/IEC 14496-3. This capability is defined in Annex H.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) amr (1)}	This Object Identifier is used to indicate the generic capability for the GSM Adaptive Multi rate speech codec. This capability is defined in Annex I.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) acelp (2)}	This Object Identifier is used to indicate the generic capability for the TIA/EIA/ANSI IS-136 ACELP voice codec. This capability is defined in Annex J.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) us1 (3)}	This Object Identifier is used to indicate the generic capability for the TIA/EIA/ANSI IS-136 US1 voice codec. This capability is defined in Annex K.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) is127evrc (4)}	This Object Identifier is used to indicate the generic capability for the TIA/EIA IS-127 Enhanced Variable Rate Codec. This capability is defined in Annex L.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) ISO/IEC 14496-1 (0)}	This Object Identifier is used to indicate the generic capability for ISO/IEC 14496-1. This capability is defined in Annex G.

Table D.1/H.245

Object Identifier Value	Description
{itu-t (0) recommendaton (0) h (8) 245 generic-capabilities (1) control (3) logical-channel-bit-rate-management (0)}	This Object Identifier is used to indicate the generic capability for logical channel bit rate management. This capability is defined in Annex F.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 13818-7 (5)}	This Object Identifier is used to indicate the generic capability for the ISO/IEC 13818-7. This capability is defined in Annex M.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ITU-R BS.1196 (6)}	This Object Identifier is used to indicate the generic capability for the ITU-R BS.1196. This capability is defined in Annex M.

19) *In Appendix VI, immediately after Figure VI.3, the following sentence is introduced:*

An example of setting H263Capability parameters in each OpenLogicalChannel messages is summarized in Table VI.2.

20) *In Appendix VIII, a new row is added at the end of Table VIII.1 as follows:*

Table VIII.1/H.245 – List of generic capabilies defined in other standardization organizations

G.722.2	Audio protocol	{itu-t (0) recommendation (0) g (7) 7222 generic-capabilities (1) 0}	ITU-T Rec. G.722.2

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, 2003