Telecommunications Standardisation sector
Study Group 15
Experts Group for Video Coding and Systems
in ATM and Other Network environments

Document AVC-745
Version 1
15-18 May, Haninge

Source: BT
Title: Draft Recommendation H.245
Purpose: Draft Recommendation

H.223
H.245
H.324
} for alignment

The meeting of SG15-LBC in March 1995 agreed to harmonise draft recommendations H.245 and H.246, and agreed technical changes related to PSTN operation. These changes were made to the draft that had been submitted to SG15 WP1, and the resulting document was distributed by e-mail on 7 April.

Since then further editorial improvements have been made as a result of e-mail discussion, and as a result of a LBC editor's meeting held, in Boston 8-10 May, to ensure editorial consistency between the PSTN multimedia draft recommendations.

Change bars have been used to indicate changes relative to the version that was distributed on 7 April 1995.

INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

DRAFT H.245

(11 May~~7 April~~, 1995)

LINE TRANSMISSION OF NON-TELEPHONE SIGNALS

MULTIMEDIA SYSTEM CONTROL

DRAFT ITU-T Recommendation H.245

FOREWORD

The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the International Telecommunication Union. 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, established the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

ITU-T Recommendation H.245 was prepared by the ITU-T Study Group 15 (199x-199x) and was approved by the WTSC (Place, Month xx-xx, 199x).

NOTES

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

CONTENTS

MULTIMEDIA SYSTEM CONTROL

(Place, 199x)

{Ed. Text within curly brackets beginning "Ed" is not part of the Recommendation, but serves to hold notes, questions etc by the editor.}

1 Scope

This Recommendation specifies syntax and semantics of multimedia terminal system control ~~information~~ messages as well as procedures to use them for the inband negotiation at the start of or during communication. The messages cover receiving and transmitting capabilities as well as mode preference from the receiving end. Acknowledgement is applied to the procedures for reliable~~secure~~ exchange of messages. There is independence of the two directions of transmission.

This recommendation covers a wide range of applications, including storage/retrieval, messaging and "distribution" services as well as conversational. It covers ~~both ATM-based and PSTN-based~~ multimedia systems that use the multiplexes defined in H.222.0 and H.223. These are subsequently referred to as H.222.0-based systems and H.223-based systems respectively~~:~~. Tthese different systems share the same syntax and semantics, and are therefore bit-wise compatible. The procedures are also the same, except for some minor differences. However, they do make use of different transport protocols, as specified in the Annex A and Annex B.

Note: there should be no confusion with the T.120 management system, which is carried within the data stream, and covers different functionalities from those described here - the multimedia system control stream and the T.120-data stream are complementary.

2 References

The following ITU-T Recommendations, and other references, contain provisions which, through reference in this text, constitute the provisions of the Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all 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.

[1] ITU-T Recommendation G.711 (1988) - Pulse code modulation (PCM) of voice frequencies

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

[3] ITU-T Recommendation G.723 (199x) - Speech coding for visual telephony

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

[5] ITU-T Recommendation H.221 (1993) - Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices

[6] ITU-T Recommendation H.222.0 (199x) - Coding of moving Pictures and Associated Audio - ISO/IEC 13818-1

[7] ITU-T Recommendation H.222.1 (199x) - Multimedia multiplex and synchronisation for audiovisual communication in ATM environments

[8] ITU-T Recommendation H.223 (199x) - Multiplexing protocol for low-bit-rate audiovisual services

[9] ITU-T Recommendation H.230 (1993) - Frame-synchronous control and indication signals for audiovisual systems

[10] ITU-T Recommendation H.233 (1993) - Confidentiality system for audiovisual services

[11] ITU-T Recommendation H.234 (1993) - Authentication and key management

[12] ITU-T Recommendation H.261 (1993) - Video Codec for audiovisual services at px64 kbit/s

[13] ITU-T Recommendation H.262 (199x) - Generic cCoding of moving Pictures and Associated Audio - ISO/IEC 13818-2

[14] ITU-T Recommendation H.263 (199x) - Video coding for narrow telecommunications channels at < 64 kbit/s

[15] ITU-T Recommendation Q.922 (1993) - ISDN data link layer specification for frame mode bearer services

[16] ITU-T Recommendation T.35 (1991) - Procedure for the allocation of CCITT defined codes for non-standard facilities

[17] ITU-T Recommendation T.50 (1993) - International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) - Information technology - 7-bit coded character set for information interchange

[18] ITU-T Recommendation T.120 (199x) - Data protocols for multimedia conferencing - under development.

[19] ITU-T Recommendation V.8bis (1995) - Procedures for the identification and selection of common modes of working between data circuit terminating equipment (DCE) and between data terminal equipment (DTE) over the general switched telephone network and leased point-to-point telephone-type circuits.

[20] ITU-T Recommendation V.14 (1993) - Transmission of start-stop characters over synchronous bearer channels.

[21] ITU-T Recommendation V.42 (1993) - Error-correcting procedures for DECs using asynchronous-to-synchronous conversion

[22] ITU-T Recommendation V.42bis (1990) - Data compression procedures for data circuit terminating equipment (DCE) using error correcting procedures

[23] ITU-T Recommendation X.214 (1993) - Open systems interconnection - transport service definition

[24] ITU-T Recommendation X.224 (1993) - Transport protocol specification for open systems interconnection

[25] ITU-T Recommendation X.680 (1994). Information Technology - Abstract Syntax Notation One (ASN.1) - Specification of basic notation

[26] ITU-T Recommendation X.691 (199x). Information Technology - ASN.1 Encoding Rules - Specification of Packed Encoding Rules (PER) - under development

[27] ISO/IEC 11172-3 (1993) - Information Technology - Coding of moving Pictures and Associated Audio for digital storage media at up to about 1.5 Mbit/s

[28] ISO/IEC 13818-6 (1995) - Information Technology - Generic Coding of moving Pictures and Associated Audio

{Ed. Definitions section deleted as nothing to put in it.}

DSM-CC

3 General

The procedures in this Recommendation are intended to ensure that the only multimedia ~~only those~~ signals ~~are~~to be transmitted are those that can be received and ~~appropriately~~ treated appropriately by a receive ~~the remote~~ terminal ~~without ambiguity~~. This requires that the capabilities of each terminal to receive and decode be known to the other terminal. It is not necessary that a terminal understand or store all incoming capabilities: those that are not understood, or can not be used ~~(because the terminal has no means to transmit corresponding information)~~ shall be ignored, and no fault shall be considered to have occurred.

The total capability of a terminal to receive and decode various signals is made known to the other terminal by transmission of its capability set.

There are separate capability messages correspondingfor ability to the receive and transmit capabilities of a terminal: receiveCapSet and transmitCapSet.

These capability sets provide for more than one stream of a given medium type to be sent "simultaneously" ~~of course not at exactly the same time, but multiplexed within logical channels~~. For example, a terminal may declare its ability to receive (or send) two independent H.262 video streams and two independent G.722 audio streams at the same time. The capability sets also provide for terminals that are software-oriented, and have capabilities that are interdependent; for example, they might be able to decode pictures faster (lower MPI) when the audio algorithm is less computation-intensive (G.722) and vice versa.

Capabilities are indicated by a sequence of TermCapSet records. Each record is indicated as being an independent record or a dependent record. Independent capability records simply indicate the mediumdata types that can be processed (G.711 audio, G.728 audio, QCIF H.261 video etc.), without any implication of simultaneous capability,

except that one of each mediumdata type (video, audio, etc.) can be processed at the same time. Dependent capability records explicitly state the simultaneous capabilities.

Means are provided to control the mode transmitted by the far-end terminal whenever this has not been explicitly prevented by that terminal. Messages are defined for encryption and key management, for maintenance, for audio C&I (such as AIM etc. contained in H.230), for video C&I (such as freeze-picture, fast-update, etc.), for non-standard messages and for changing or ending a session. Provision is made for keyboard, keypad and other user-input control messages.

This Recommendation defines messages and procedures to acknowledge the receipt of messages for robust communications.

4 Messages: syntax

This section specifies the syntax of messages using the notation defined in ASN.1 [25]. Messages shall be encoded for transmission by applying the packed encoding rules specified in [26] using the basic aligned variant.

4.1 Multimedia System Control Messages

```
MultimediaSystemControl         ::=CHOICE
{
    declaredTermCapSet          [APPLICATION 0]IMPLICIT DeclaredTermCapSet,
    requestCapability           [APPLICATION 1]IMPLICIT RequestCapability,
    requestMode                 [APPLICATION 2]IMPLICIT RequestMode,
    acknowledgement             [APPLICATION 3]IMPLICIT Acknowledgement,
    downloadableSoftware        [APPLICATION 4]IMPLICIT DownloadableSoftware,
    encryption                  [APPLICATION 5]IMPLICIT Encryption,
    changeOrEndSession          [APPLICATION 6]IMPLICIT ChangeOrEndSession,
    c&i                         [APPLICATION 7]IMPLICIT C&I,
    logicalChannelSignalling    [APPLICATION 8]IMPLICIT LogicalChannelSignalling,
    userInputAndOther           [APPLICATION 9]IMPLICIT UserInputAndOther,
    multiplexTableDownload      [APPLICATION 10]IMPLICIT MultiplexTableDownload,
    flowControl                 [APPLICATION 11]IMPLICIT FlowControl,
    ...
}
```

T.120

4.2 Transmit and Receive capabilities

```
DeclaredTermCapSet              ::=SEQUENCE
{
    connectionless              BOOLEAN,
    remoteControllability       BOOLEAN,
    symmetricalCapSets          BOOLEAN,
    yearOfRecommendation        NumericString (SIZE(4..4)) (FROM ("0123456789")),
    independentTermCapSet       [0]IMPLICIT SEQUENCE OF TermCapSet OPTIONAL,
    dependentTermCapSet         [1]IMPLICIT SEQUENCE OF TermCapSet OPTIONAL,
    ...
}
```

{Ed. I have moved yearOfReccommendation to here as it seems more appropriate. I have added booleans for connectionless and remote-controllability indication: this has little cost in efficiency (two bits with PER) and seems a better solution to this problem than indicating these by the absence of any receive or transmit capabilities.}

```
TermCapSet                      ::=SEQUENCE
{
    receiveCapSet               [0]IMPLICIT CapSet OPTIONAL,
    transmitCapSet              [1]IMPLICIT CapSet OPTIONAL,
    ...
}

CapSet                          ::=SEQUENCE
```

```
{
    nonStandard             [0]IMPLICIT SEQUENCE OF OCTET STRING OPTIONAL,
    videoCap                [1]IMPLICIT SEQUENCE OF VideoCap OPTIONAL,
    audioCap                [2]IMPLICIT SEQUENCE OF AudioCap OPTIONAL,
    dataCap                 [3]IMPLICIT SEQUENCE OF DataCap OPTIONAL,
    atmNetworkAdaptCap      [4]IMPLICIT ATMNetworkAdaptCap OPTIONAL,
    h223Capability          [5]IMPLICIT H223Capability OPTIONAL,
    ...
}

VideoCap                    ::=CHOICE
{
    nonStandard             [0]IMPLICIT OCTET STRING,
    h261VideoCap            [1]IMPLICIT H261VideoCap,
    h262VideoCap            [2]IMPLICIT H262VideoCap,
    h263VideoCap            [3]IMPLICIT H263VideoCap,
    ...
}

H261VideoCap                ::=SEQUENCE
{
    qcifMPI                 [0]IMPLICIT INTEGER (1..4) OPTIONAL,
    cifMPI                  [1]IMPLICIT INTEGER (1..4) OPTIONAL,
    ...
}
```

{Ed. I have decided to remove cifCapability. If cifMPI is not present, then there is no capability for CIF}

```
H262VideoCap                ::=SEQUENCE
{
    profileAndLevel         BIT STRING
                            {
                                SP@ML       (0),
                                MP@LL       (1),
                                MP@ML       (2),
                                MP@H-14     (3),
                                MP@HL       (4),
                                SNR@LL      (5),
                                SNR@ML      (6),
                                Spat@H-14   (7),
                                HP@ML       (8),
                                HP@H-14     (9),
                                HP@HL       (10),
                                ...
                            },
    videoBitRate            [0]IMPLICIT INTEGER OPTIONAL,
    vbvBufferSize           [1]IMPLICIT INTEGER OPTIONAL,
    samplesPerLine          [2]IMPLICIT INTEGER OPTIONAL,
    linesPerFrame           [3]IMPLICIT INTEGER OPTIONAL,
    framesPerSecond         [4]IMPLICIT INTEGER OPTIONAL,
    luminanceSampleRate     [5]IMPLICIT INTEGER OPTIONAL,
    ...
}

H263VideoCap                ::=SEQUENCE
{
    sqcifMPI                [0]IMPLICIT INTEGER (1..16) OPTIONAL,
    qcifMPI                 [1]IMPLICIT INTEGER (1..16) OPTIONAL,
    cifMPI                  [2]IMPLICIT INTEGER (1..16) OPTIONAL,
    h263Modes               BIT STRING
                            {
                                unrestrictedVector      (0),
                                arithmeticCoding        (1),
                                advancedPrediction      (2),
                                pbFrames                (3),
```

```
                                    ...
                                },
        ...
}

AudioCap                            ::=CHOICE
{
    nonStandard                     [0]IMPLICIT OCTET STRING,
    itu-tAudio1                     [1]IMPLICIT BIT STRING
                                    {
                                        g711Alaw        (0),
                                        g711Ulaw        (1),
                                        g722            (2),
                                        g728            (3)
                                    },
    mpegAudio                       [2]IMPLICIT MpegAudio,
    itu-tAudio2                     [3]IMPLICIT Itu-tAudio2,
    ...
}

MpegAudio                           ::=SEQUENCE
{
    audioLayer                      BIT STRING
                                    {
                                        layer1          (0),
                                        layers2         (1),
                                        layers3         (2),
                                    },
    audioSampling                   BIT STRING
                                    {
                                        at32k           (0),
                                        at44k1          (1),
                                        at48k           (2),
                                    },
    asynchronousCap                 BOOLEAN,
    aAudioCorrectionModes           ENUMERATED
                                    {
                                        mode1           (0),
                                        mode2           (1),
                                        mode3           (2),
                                        allThreeModes   (3),
                                    },
    bitRate                         INTEGER,
    ...
}

Itu-tAudio2                             ::=SEQUENCE
{
    g723Flags                       BIT STRING
                                    {
                                        silenceSuppression      (0),
                                        dtmfTransport           (1),
                                        lowBitRate              (0),
                                        highBitRate             (1),
                                    },
    lengthOfFrame                   INTEGER,
    maxDelayJitter                  INTEGER,
    supervisoryFrameLength          [0]IMPLICIT INTEGER OPTIONAL,
    ...
}

DataCap                             ::=BIT STRING
{
    t120                            (0),
    v42/lapm                        (1),
    v42bis                          (2),
```

```
    dsm-cc                      (3),
    transparentData             (4),
    ...
}

ATMNetworkAdaptCap              ::=SEQUENCE
{
    aal                         BIT STRING
                                {
                                    Aal1        (0),
                                    Aal5        (1),
                                },
    h222Multiplex               BIT STRING
                                {
                                    transportStream  (0),
                                    programStream    (1),
                                },
    bitRate                     INTEGER,
    numberOfVCs                 INTEGER,
    ...
}

H223Capability                  ::=SEQUENCE
{
    numberOfMultiplexTables     INTEGER (0..255),
    multiplexTableNesting       INTEGER (0..255),
    muxPDUSize                  INTEGER (0..65535),
    al3CSSDUSize                INTEGER (0..65535),
    al3SendBufferSize           INTEGER (0..65535),
    maxDelayJitter              INTEGER,
    ...
}
```

4.3 Request Transmit and Receive capabilities

```
RequestCapability               ::=SEQUENCE
{
    requestTxCap                BOOLEAN,
    requestRxCap                BOOLEAN,
    ...
}
```

4.4 Request Mode (Remote control)

```
RequestMode                     ::=SEQUENCE
{
    requestedMode               TermCapSet,
    ...
}
```

4.5 Acknowledgement

```
Acknowledgement                 ::=SEQUENCE
{
    acknowledgementType         ENUMERATED
                                {
                                    ackDeclaredTermCapSet       (0),
                                    nackDeclaredTermCapSet      (1),
                                    ackRequestMode              (2),
                                    nackRequestMode             (3),
                                    ackLogicalChannel           (4),
```

```
                                nackLogicalChannel              (5),
                                ackMultiplexTableDownload       (6),
                                nackMultiplexTableDownload      (7),
                                ackSelectMultiplexTable         (8),
                                nackSelectMultiplexTable        (9),
                                ...
                            },
    declaredTermCapSetFailure   [0]IMPLICIT DeclaredTermCapSetFailure OPTIONAL,
    requestModeFailure          [1]IMPLICIT RequestModeFailure OPTIONAL,
    logicalChannelFailure       [2]IMPLICIT LogicalChannelFailure OPTIONAL,
    tableDownloadFailure        [3]IMPLICIT TableDownloadFailure OPTIONAL,
    tableSelectFailure          [4]IMPLICIT TableSelectFailure OPTIONAL,
    ...
}

DeclaredTermCapSetFailure       ::=ENUMERATED
{
    unspecifiedCause            (0),
    invalidCapSet               (1),
    ...
}

RequestModeFailure              ::=ENUMERATED
{
    unspecifiedCause            (0),
    noRemoteControllability     (1),
    invalidRequest              (2),
    modeUnavailable             (3),
    multipointConstraint        (4),
    requestDenied               (5),
    ...
}

LogicalChannelFailure           ::=ENUMERATED
{
    unspecifiedCause            (0),
    logicalChannelMessageInvalid    (1),
    modeNotDecodable            (2),
    ...
}

TableDownloadFailure            ::=ENUMERATED
{
    unspecifiedCause            (0),
    invalidTable                (1),
    tableOutOfRange             (2),
    ...
}

TableSelectFailure              ::=ENUMERATED
{
    unspecifiedCause            (0),
    tableNotDownloaded          (1),
    tableOutOfRange             (2),
    ...
}
```

{Ed. All references to downloading software have been removed. These were incomplete. DSMCC should provide a complete solution to the downloading of executable software; also T.120 will provide binary file transfer.}

4.6 Encryption

```
Encryption                      ::=CHOICE
{
```

```
    encryptionSE            [0]IMPLICIT EncryptionSE,
    encryptionIV            [1]IMPLICIT EncryptionIV,
    ...
}

EncryptionSE                ::=SEQUENCE
{
    messageIdentifier       INTEGER (0..255),
    content                 OCTET STRING,  -- SEQUENCE INTEGER (0..255) --
    ...
}

EncryptionIV                ::=SEQUENCE
{
    initializationVector    OCTET STRING,  -- SEQUENCE INTEGER (0..255) --
    ...
}
```

4.7 Change or end session

```
ChangeOrEndSession          ::=SEQUENCE
{
    command                 ENUMERATED
    {
        changeSession       (0),
        endSession          (1),
    },
    ...
}
```

4.8 Control and indication

The following control and indication syntax has been defined to maximise compatibility with H.221 [5] and H.230 [9]. Octets have been chosen to have the same value as in these Recommendations wherever possible.

```
C&I                         ::=CHOICE
{
    nonStandard             [0]IMPLICIT OCTET STRING,
    h221C&I                 [1]IMPLICIT H221C&I,
    h230C&I                 [2]IMPLICIT H230C&I,
    h223C&I                 [3]IMPLICIT H223C&I,
    ...
}
```

{Ed. I have changed the way the H.230 C&I is included. By using an octet string, nothing from H.230 needs to be repeated, and no arbitrary choices of which messages to include need to be made. Also, modifications can be made to H.230 without the need to update this document. Mandatory C&I signals should be specified in the appropriate system recommendation, as has been done in H.320.}

```
H221C&I                     ::=SEQUENCE
{
    logicalChannelNumber    INTEGER (0..8191),  -- Limited by 12 bit PID in H.222.0 --
    c&iType                 ENUMERATED
    {
        lca                 (82),  -- Audio Loop Request --
        lcv                 (83),  -- Video Loop Request --
        lcd                 (84),  -- Digital Loop Request --
        lco                 (85),  -- Loopback Command Off --
        vcf                 (80),  -- Video Command: freeze picture request --
        vcu                 (81),  -- Video Command: fast update request --
    },
    ...
}
```

```
H230C&I                 ::=SEQUENCE
{
    logicalChannelNumber    INTEGER (0..8191),  -- Limited by 12 bit PID in H.222.0 --
    c&iData                 OCTET STRING,
    ...
}

H223C&I                 ::=CHOICE
{
    selectMultiplexTable    [0]IMPLICIT INTEGER (0..255),
    logicalChannelSkew      [1]IMPLICIT SEQUENCE
                            {
                                logicalChannelNumber1   INTEGER (0..8191),
                                logicalChannelNumber2   INTEGER (0..8191),
                                skew                    INTEGER (0..4095),
                                ...
                            },
    vcuGOB                  [2]IMPLICIT SEQUENCE
                            {
                                logicalChannelNumber    INTEGER (0..8191),
                                vcuGOBData              INTEGER,
                                ...
                            }
    ...
}
```

4.9 Logical channel signalling

```
LogicalChannelSignalling    ::=SEQUENCE
{
    messageType             ENUMERATED
                            {
                                channelSetUp            (0),
                                channelRedefinition     (1),
                                channelClose            (2),
                                ...
                            },
    logicalChannelNumber    INTEGER (0..8191),  -- Limited by 12 bit PID in H.222.0 --
    channelUse              CHOICE
                            {
                                videoCap                [0]IMPLICIT VideoCap,
                                audioCap                [1]IMPLICIT AudioCap,
                                dataCap                 [2]IMPLICIT DataCap,
                                ...
                            } OPTIONAL,
    pstnSpecifics           SEQUENCE
                            {
                                adaptationLayerType     INTEGER (0..157),
                                privateALDescriptor     [0]IMPLICIT OCTET STRING OPTIONAL,
                                interruptableFlag       BOOLEAN,
                                adaptationLayerSN       BOOLEAN,
                                adaptationLayerCRC      BOOLEAN,
                                adaptationLayerEXT      BOOLEAN,
                                ...
                            } OPTIONAL,
    atmSpecifics            SEQUENCE
                            {
                                ...
                            } OPTIONAL,
    ...
}
```

4.10 User input and other messages

```
UserInputAndOther           ::=CHOICE
{
    alphanumeric            [0]IMPLICIT OCTET STRING,
    numeric                 [1]IMPLICIT OCTET STRING,
    xyFunctions             [2]IMPLICIT SEQUENCEOCTET STRING,
    {
        operation           ENUMERATED
        {
            penUp           (0),
            penDown         (1),
            click           (2),
            doubleClick     (3),
            drag            (4),
            ...
        },
        xAddress            INTEGER (0..4095),
        yAddress            INTEGER (0..4095)
    },
    nonStandard             [5]IMPLICIT OCTET STRING,
    ...
}
```

4.11 Multiplex table download

```
MultiplexTableDownload      ::=SEQUENCE
{
    multiplexTable          OCTET STRING,
    ...
}
```

4.12 Flow Control

```
FlowControl                 ::=SEQUENCE
{
    logicalChannelNumber    INTEGER (0..8191), -- for 12 bit PID--
    noRestrictionFlag       BOOLEAN,
    maximumBitRate          [0]IMPLICIT INTEGER OPTIONAL,
    ...
}
```

5 Messages: semantic definitions

This section provides semantic definitions and constraints on the syntax elements defined in the previous section.

5.1 MultimediaSystemControl Messages

MultimediaSystemControl: a choice of messages. This provides a top level definition of the messages specified in the Recommendation.

5.2 Transmit and Receive capabilities

DeclaredTermCapSet: indicates a terminal's transmit and receive capabilities.

connectionless is a boolean that indicates whether the terminal is able to respond to any upstream signals, whether of the ReceiveCapSet type or requests to transmit in a particular mode. When this boolean is true, the terminal is connectionless and therefore unable to respond to upstream signals; in this case, the TermCapSets shall contain no receiveCapSets, otherwise at least one TermCapSet shall contain receiveCapSet.

remoteControllability is a boolean that indicates whether the terminal's mode of operation can be controlled by the remote terminal. When this boolean is true, the terminal can be remotely controlled.

symmetricalCapSets is a boolean that indicates whether the transmit capabilities are the same as the receive capabilities. When this boolean is true, the transmit capabilities are the same as the receive capabilities, and the TermCapSets shall contain no transmitCapSets.

In addition to the above, table 1/H.245 specifies which combinations of booleans connectionless, remoteControllability and symmetricalCapSets are valid, and for those combinations whether one or more receiveCapSets and transmitCapSets shall be present in the DeclaredTermCapSet.

connection-less	remote-Controllability	symmetrical-CapSets	receive-CapSet	transmit-CapSet
FALSE	FALSE	FALSE	PRESENT	ABSENT
FALSE	FALSE	TRUE	ILLEGAL	
FALSE	TRUE	FALSE	PRESENT	PRESENT
FALSE	TRUE	TRUE	PRESENT	ABSENT*
TRUE	FALSE	FALSE	ABSENT	PREABSENT
TRUE	FALSE	TRUE	ILLEGAL	
TRUE	TRUE	FALSE	ILLEGAL	
TRUE	TRUE	TRUE	ILLEGAL	

Table 1/H.245. The relationship between the boolean fields and the presence of receive and transmit CapSets, and showing the illegal combinations of the boolean fields.
*:transmitCapSet is not present, but transmission capabilities are identical to receive capabilities.

yearOfRecommendation is a NumericString indicating the year of the approval or revision of this recommendation implemented. Its shall consist of four digits, an example being "1995".

Terminal capabilities are indicated by TermCapSet types in addition to the booleans described above. Two types of TermCapSet are defined: independent and dependent TermCapSets. These have the same syntax, but different semantic definitions. It is possible to include a mix of independent and dependent TermCapSets, but at least one independent TermCapSet shall be present.

Independent capabilities: the indication of more than one capability of the same type (video, audio, adaptation or data), such as H.261 and H.262, does not indicate the ability to perform multiple simultaneous processing of the same data type, but list merely the available capabilities. A terminal shall at least be capable of the simultaneous processing of any one indicated independent capability of each type.

Dependent capabilities: this explicitly states the combination of simultaneous capabilities. A terminal shall be capable of the simultaneous processing of all the dependent capabilities indicated in a single TermCapSet.

Note: terminals that can not vary the allocation of resources can indicate their capability completely by use of a single independent TermCapSet.

TermCapSet: a set of receive and transmit capabilities. In general, both are optional, but at least one shall be present.

CapSet: a set of unidirectional capabilities. ATMNetworkAdaptCap may only be present in H.222.0-basedH.222.0-basedATM-based systems. H223Capability may only be present in H.223-basedH.223-basedPSTN-based systems.

nonStandard: is an octet string that can be used to indicate proprietry capabilities. The octet string shall consist of at least four octets, the first four of which are: country code (octet 1 as in T.35 [16]; octet 2*), manufacturer code (next two octets*), *=assigned nationally.

VideoCap: indicates video capabilities. The indication of more than a single capability within a single VideoCap does not indicate simultaneous processing capability. Simultaneous processing capability can be indicated by multiple instances of VideoCap in a dependent TermCapSet.

nonStandard is an octet string that can be used to indicate proprietry video capabilities. The octet string shall consist of at least four octets, the first four of which are: country code (octet 1 as in T.35 [16]; octet 2*), manufacturer code (next two octets*), *=assigned nationally.

H261VideoCap: indicates H.261 [12] capabilities.

qcifMPI indicates the minimum picture interval in units of 1/29.97 for the processing of QCIF pictures and shall take one of the values 1, 2, 3, 4. cifMPI indicates the minimum picture interval in units of 1/29.97 for the processing of CIF pictures and shall take one of the values 1, 2, 3, 4. If cifMPI is not present, then no capability of processing CIF resolution pictures is indicated. When present as a part of a DeclaredTermCapSet, qcifMPI shall be present, but cifMPI shall only be present if CIF capability is being indicated; otherwise exactly one of qcifMPI and cifMPI shall be present.

H262VideoCap: indicates H.262 [13] capabilities.

ProfileAndLevel is a bit string in which a bit set to '0' indicates that operation at the particular profile and level is not possible, while a value of '1' indicates that such operation is possible. An encoder shall produce bitstreams compliant to the specifications of a profile and level for which it has indicated capability, but also within the limitations imposed by the optional fields (see below). A decoder shall be able to accept all bit streams conforming to a profile and level for which it has indicated capability, provided it is within the limitations indicated by the optional fields (see below).

The optional fields indicate limitations to the capabilities declared with ProfileAndLevel. These are integers with the following units:

videoBitRate	:	bits per second
vbvBufferSize	:	bits
samplesPerLine	:	samples per line
linesPerFrame	:	lines per frame
framesPerSecond	:	frames per second
luminanceSampleRate	:	samples per second

H263VideoCap: indicates H.263 [14] capabilities.

sqcifMPI, qcifMPI, and cifMPI indicate the minimum picture interval in units of 1/29.97 for the processing of SQCIF, QCIF, and CIF format pictures respectively. The ability to process the formats SQCIF, QCIF and CIF is indicated by the presence of a corresponding MPI value. When the MPI is present the corresponding format can be processed and when it is absent it can not be processed. Other than when present as a part of a DeclaredTermCapSet, exactly one of sqcifMPI, qcifMPI and cifMPI shall be present.

h263Modes indicates the modes that can be processed.

unrestrictedVector, when '1' indicates that motion vectors may point outside the picture, and when '0' indicates that motion vectors do not point outside the picture.

arithmeticCoding, when '1' indicates that arithmetic coding, and when '0' indicates standard Huffman coding.

advancedPrediction, when '1' indicates separate motion vectors for each luminance block in a macroblock and blocks used in motion compensation overlap, and when '0' indicates one motion vector for each macroblock and non-overlapping blocks.

pbFrames, when '1' indicates that B frames can be decoded, and when '0' that they can not be decoded.

~~A value of '1' indicates that the mode can be processed and a value of '0' indicates that it can not be processed.~~

AudioCap: indicates audio capabilities. The indication of more than a single capability within a single AudioCap does not indicate simultaneous processing capability. Simultaneous processing capability can be indicated by multiple instances of AudioCap in a dependent TermCapSet.

nonStandard is an octet string that can be used to indicate proprietry audio capabilities. The octet string shall consist of at least four octets, the first four of which are: country code (octet 1 as in T.35 [16]; octet 2*), manufacturer code (next two octets*). *=assigned nationally.

itu-tAudio1 indicates the ability to process G.711 A-law, G.711 U-law, G.722 and G.728 coded audio[1][2][4]. A value of '1' indicates that the mode can be processed and a value of '0' indicates that it can not be processed.

MpegAudio: indicates the ability to process MPEG audio [27].

audioLayer indicates which audio coding layers can be processed. A value of '1' indicates that the coding layer can be processed and a value of '0' indicates that it can not be processed. When present as part of a DeclaredTermCapSet, layer1 capability shall be indicated when layer2 capability is indicated, and layer1 and layer2 capability shall be indicated when layer3 capability is indicated; otherwise exactly one of layer1, layer2 and layer3 shall be set to '1'.

audioSampling indicates which sample rates, at32k (32KHz), at44k1 (44.1KHz) and at48k (48KHz), can be processed. A value of '1' indicates that the sample rate can be processed and a value of '0' indicates that it can not be processed.

bitRate is measured in units of bits per second.

{Ed. I have assumed that all MPEG audio is MPEG-1. Is this correct?}

Itu-tAudio2: indicates capability for processing of G723 coded audio [3].

g723Flags indicates various capabilities. A value of '1' indicates that the capability is present and a value of '0' indicates that it is not.

~~silenceSuppression indicates the capability of using discontinuous transmission and noise fill during non-speech intervals.~~

~~dtmfTransport indicates ????~~

~~{Ed. More detail is required here, but what?}~~

lowBitRate indicates the capability to transmit at 5.3 kbit/s.

highBitRate indicates the capability to transmit at 6.4 kbit/s.

Note. Changing between the lower and higher bit rates is not considered to be a change of use of a logical channel, and so may be done without invoking the logical channel signalling acknowledged procedure.

~~lengthOfFrame ????~~

~~{Ed. I don't see why this field is needed, and propose to delete it.}~~

~~maxDelayJitter indicates the maximum peak-to-peak multiplexing jitter that a terminal can process. It is measured in milliseconds. Multiplexing jitter is defined as the difference in time of delivery of the first octet of an audio frame when delivered in the multiplexed stream and when it would be delivered at constant bit rate without a multiplex.~~

~~{Ed. Is this definition OK? Can someone provide better words?}~~

~~supervisoryFrameLength shall be present if and only if silenceSuppression has the value '0'.~~

~~{Ed. What does supervisoryFrameLength indicate?}~~

~~{Ed. Is supervisoryFrameLength needed here?}~~

DataCap: indicates data capabilities. The indication of more than a single capability within a single DataCap does not indicate simultaneous processing capability. Simultaneous processing capability can be indicated by multiple instances of DataCap in a dependent TermCapSet.

t120 indicates the capability to support the t120 protocol stack [18].

v42/lapm indicates the capability to support the LAPM protocol defined in V.42[21].

v42bis indicates the capability to support V.42bis[22].

dsm-cc indicates the capability to support the DSM-CC protocol [28].

T. Gary useful for modem case

{Ed. Is more detail required here?}

transparentData indicates the capability to support transparent data transfer.

{Ed. This looks like a meaningless statement. Is it really possible to transfer data without any specfic protocol? If not, I propose deleting this codepoint, as proprietry techniques would be needed, and there are many means available to indicate proprietry capabilities. Note that user input and other messages provides a complete mechanism for transferring transparent data, but is probably only practical for small amounts of data.}

ATMNetworkAdaptCap: indicates network adaptation capabilities. The indication of more than a single capability within a single ATMNetworkAdaptCap does not indicate simultaneous processing capability. Simultaneous processing capability can be indicated by multiple instances of ATMNetworkAdaptCap in a dependent TermCapSet.

aal indicates AAL capability. It is a bit string in which a bit set to '0' indicates that operation with the particular AAL is not possible, while a value of '1' indicates that such operation is possible.

h222Multiplex indicates H.222.0 multiplex capability. It is a bit string in which a bit set to '0' indicates that operation with the particular multiplex is not possible, while a value of '1' indicates that such operation is possible.

bitRate indicates the maximum bit rate capability of the terminal. If the capability for more than one virtual channel is declared, then this indicates the maximum total bit rate of all of these virtual channels. It is measured in units of 64 kbit/s.

numberOfVCs indicates the maximum number of virtual channels that the terminal can process.

{Ed. This seems incomplete. What happens when there is more than one VC? What can be put in each? Can they have different AALs?}

H223Capability: indicates capabilities specific to the H.223 multiplex [8]. This shall only be present in H.223-basedH.223-basedPSTN-based systems.

{Ed. I have removed all limits from this section as limits are outside the scope of this recommendation, which should be quite generic. However, do we need to specify limits elsewhere, such as H.324? We already do this for the number of tables and for the amount of nesting, but not for the sizes as far as I know. Unless we do, it will be quite legitimate for a terminal to signal an AL3 send buffer of zero size and therefore have no retransmit capability: is this what we want?}numberOfMultiplexTables indicates the maximum number of downloaded multiplex tables that a terminal can process at one time.

multiplexTableNesting indicates the maximum number of levels of nesting in the definition of a multiplex tables that a terminal can process.

muxPDUSize indicates the maximum size of the information field of a MUX-PDU that the terminal shall be capable of processing. It is measured in units of octets.

{Ed. Is this limit needed? I don't see how large MUX PDUs complicate decoders, and so propose to remove this field.}

al3CSSDUSize indicates the maximum number of octets in each CS-SDU of adaptation layer type 3.

al3SendBufferSize indicates the minimum size of the send buffer, B_S, of adaptation layer type 3. It is measured in units of octets.

maxDelayJitter indicates the maximum peak-to-peak multiplexing jitter that a terminal can process. It is measured in milliseconds. Multiplexing jitter is defined as the difference in time of delivery of the first octet of an audio frame when delivered in the multiplexed stream and when it would be delivered at constant bit rate without a multiplex.

{Ed. Is this definition OK? Can someone provide better words?}

5.3 Request Transmit and Receive capabilities

RequestCapability: this message requests the far end terminal to transmit its transmit and/or receive capabilities.

5.4 Request Mode (Remote control)

RequestMode: this message requests the far end terminal to change its transmission mode to the mode indicated. requestedMode is a dependent TermCapSet that contains no transmitCapSet and one receiveCapSet that is consistent with the transmitCapSet of the most recently received DeclaredTermCapSet and the receiveCapSet of the most recently transmitted DeclaredTermCapSet.

5.5 Acknowledgement

Acknowledgement: this message indicates whether the terminal correctly received a message. Explanation shall be given for failure: declaredTermCapSetFailure shall be present when acknowledgementType is nackDeclaredTermCapSet; requestModeFailure shall be present when acknowledgementType is nackRequestMode; logicalChannelFailure shall be present when acknowledgementType is nackLogicalChannel; tableDownloadFailure shall be present when acknowledgementType is nackMultiplexTableDownload; and tableSelectFailure shall be present when acknowledgementType is nackSelectMultiplexTable.

DeclaredTermCapSetFailure: this message indicates why the terminal failed to receive ~~acknowledge receipt of~~ a DeclaredTermCapSet message correctly. Valid responses are: unspecified cause and invalidCapSet (the DeclaredTermCapSet message was syntactically or semantically incorrect).

RequestModeFailure: this message indicates why the terminal was unable to change its transmission mode to the requested mode. Valid responses are: unspecified cause, noRemoteControllability (The remote terminal can not or does not wish to be remotely controlled), invalidRequest (the requested mode was syntactically or semantically incorrect), modeUnavailable (the mode requested was not consistent with the most recently sent TransmitCapSet), multipointConstraint, and requestDenied.

LogicalChannelFailure: this message indicates why the terminal failed to satisfy a request from ~~acknowledge receipt of~~ a LogicalChannelSignalling message. Valid responses are: unspecified cause, LogicalChannelMessageInvalid (the

LogicalChannelSignalling message was syntactically or semantically incorrect) and modeNotDecodable (the mode indicated was not consistent with the most recently sent ReceiveCapSet).

Note. A LogicalChannelFailure message indicating modeNotDecodable may be sent by a terminal at any time, and not only in response to a LogicalChannel message, to indicate that it can not decode the in-coming signals.

TableDownloadFailure: this message indicates why the terminal failed to accept ~~acknowledge~~ the downloading of a Multiplex Table. Valid responses are: unspecified cause, invalidTable (the table contained a syntax error), and tableOutOfRange (the number of the downloaded table is greater than the number of tables that the terminal can receive).

TableSelectFailure: this message indicates why the terminal failed to accept ~~acknowledge~~ a selectMultiplexTable command. Valid responses are: unspecified cause, tableNotDownloaded (the selected table is not the default and has not been successfully downloaded) and tableOutOfRange (the number of the selected table is greater than the number of tables that the terminal can receive).

Note. A TableSelectFailure message may be sent by a terminal at any time, and not only in response to a selectMultiplexTable command, to indicate that it can not demultiplex the in-coming signals.

5.6 Encryption

Encryption: this message conveys information relating to confidentiality in audiovisual services. The definition and use of these syntax elements is given in H.233 and H.234 [10][11].

5.7 Change or end session

ChangeOrEndSession: this message indicates a change of session or the end of a session.

A terminal shall transmit endSession immediately before clearing down. Thereafter it shall transmit no in-band messages and shall ignore any received.

{Ed. What happens on change of session?}

5.8 Control and indication

nonStandard: this is an octet string that can be used for proprietry control and indication messages. The octet string shall consist of at least four octets, the first four of which are: country code (octet 1 as in T.35 [16]; octet 2*), manufacturer code (next two octets*). *=assigned nationally.

logicalChannelNumber: this indicates the logical channel number to which the control or indication message applies. In H.222.0-basedH.222.0-based~~ATM-based~~ systems it indicates the PID in a Transport Stream and the stream_id in a Program Stream.

h221C&I: this allows the control and indication messages specified in H.221 [5] to be transmitted. The definitions of the fields in c&iType are given in H.221.

h230C&I: this allows the control and indication messages specified in H.230 [9] to be transmitted. The c&iData octet string is a sequence of one or more H.230 C&I codes. Each C&I code consists an octet chosen from table 1/H.230, followed by any required SBE characters. The escape code (111)[17] shall not be included.

h223C&I: this shall only be present in H.223-basedPSTN-based systems.

selectMultiplexTable is a command to the far-end terminal to change the multiplex table.

logicalChannelSkew is an indication to the far-end terminal of the average amount of time skew ~~in the multiplex~~ between two logical channels. logicalChannelNumber1 and logicalChannelNumber2 are logical channel numbers of open logical channels. skew is measured in milliseconds, and indicates the delay that must be applied to data belonging to logicalChannelNumber2 as measured at the multiplexbefore decoding, to achieve synchronisation with ~~after decoding, when the delay incurred by the decoding process is the same for the data from~~ logicalChannelNumber1 as measured at the multiplex~~and logicalChannelNumber2~~. The actual delay necessary for synchronisation is dependent on decoder implementation, and is a local matter for the receiver.

vcuGOB is a command to the far-end video encoder to perform a fast update of one or more GOBs. The appropriate video stream is specified by the logicalChannelNumber, which shall be an open logical channel used for H.261 or H.263 video. vcuGOBData is an integer between 0 and 255 inclusive, the least significant four bits of which indicate

the number of the first GOB to be updated, and the most significant four bits indicate the number of GOBs to be updated.

5.9 Logical channel signalling

{Ed. The mode indication message and procedures have been removed, and replaced with logical channel signalling message and procedures. Stuart: perhaps you would like to try to incorporate Annex A of H.222.1?}

This message is used for logical channel signalling acknowledged procedures.

messageType indicates the purpose of the message: the initial set up of a logical channel; the change of use of the logical channel, for example to change from video data to audio data, or simply to change the coding scheme; and the closing of a logical channel.

logicalChannelNumber indicates the logical channel number of the logical channel to which the message applies. In H.222.0-basedH.222.0-based~~ATM-based~~ systems it indicates the PID in a Transport Stream and the stream_id in a Program Stream.

channelUse is a CapSet that indicates the data that is to be conveyed on the logical channel after set up or after redefinition. This shall be present when messageType is channelSetUp or channelRedefinition.

pstnSpecifics shall be present in H.223-based~~PSTN-based~~ systems, and shall not be present in H.222.0-basedH.222.0-based~~ATM-based~~ systems.

The adaptationLayerType shall indicate the adaptation layer type to be used for the logical channel as specified below, refer to H.223 [8]. SN indicates that the sequence number (SN) field is present in CS-PDUs. CRC indicates that the CRC field is present in CS-PDUs. EXT indicates that the extension (EXT) field is present in CS-PDUs. RTX indicates that retransmission may be used.

adaptationLayerType	Adaptation Layer
0	AL1
1	AL2
2	AL2 + SN
3	AL2 + CRC
4	AL2 + SN +CRC
5	AL3
6	AL3 + EXT
7	AL3 + RTX
8	AL3 + EXT + RTX
9..14	Reserved
15	Private

privateALDescriptor shall only be present if adaptationLayerType has the value 7. It is used to describe the proprietry adaptation layer type, and shall consist of country code (octet 1 as in T.35 [16]; octet 2*), manufacturer code (next two octets*), *=assigned nationally. Additional octets may also be present.

interruptableFlag, when equal to true indicates that the channel is designated to be interruptable, and when equal to false indicates that the channel is designated to be non-interruptable.

~~adaptationLayerSN is a boolean that indicates whether the sequence number (SN) field is present in CS-PDUs. When this boolean is true, the SN shall be present. This boolean may only be true when the adptation layer is of type 2.~~

~~adaptationLayerCRC is a boolean that indicates whether the CRC field is present in CS-PDUs. When this boolean is true, the CRC shall be present. This boolean may only be true when the adptation layer is of type 2.~~

~~adaptationLayerEXT is a boolean that indicates whether the extension (EXT) field is present in CS-PDUs. When this boolean is true, the EXT shall be present. This boolean may only be true when the adptation layer is of type 3.~~

~~{Ed. Should retransmission capability in AL3 be included here, or is it always expected?}~~

atmSpecifics shall be present in H.222.0-basedH.222.0-based~~ATM-based~~ systems, and shall not be present in H.223-based~~PSTN-based~~ systems.

5.10 User input and other messages

T.64

include an explicit table with restrictions

alphanumeric: this is a string of characters coded according to T.50 [17].

numeric: this is a string of binary coded numbers, each in the range 0 to 255 inclusive.

{Ed. Should this be limited to the range 0..223 as in H.230?}

xyFunctions: ~~this shall be exactly four octets. The first octet is used to indicate function (pen up/down, click, double-click, drag, ...), the other three octets indicate the X-Y address (0..4095 each way).~~this used to indicate xy functions, such as moving pens and mouse clicks.

nonStandard: country code (octet 1 as in T.35 [16]; octet 2*), manufacturer code (next two octets*), *=assigned nationally. Additional octets may also be present.

5.11 Multiplex table download

This message is used to download multiplex tables. It shall only be present in H.223-based~~PSTN-based~~ systems.

multiplexTable is an octet string of T.50 (ASCII) characters [17] describing a sequence of MUX-SDUs and repeat counts for each table entry. The syntax used to define multiplex tables is specified in H.223 [8].

Each multiplex table is transferred in a single MultiplexTableDownload message.

5.12 Flow control

H.320 multipoint case

This message is used to specify the upper limit of bit rate of a logical channel.

When the logicalChannelNumber is zero, the limit applies to the whole multiplex.

~~noRestrictionFlag, when true indicates that any previous restriction on the bit rate for the channel is no longer applicable, and when false indicates that the limit specified in maximumBitRate supercedes any previous limit, whether higher or lower.~~

maximumBitRate is measured in units of 100 bit/s averaged over non-overlapping consecutive periods of one second. ~~It shall be present when noRestrictionFlag is false.~~When this is present, the specified limit supercedes any previous limit, whether higher or lower. When it is not present any previous restriction on the bit rate for the channel is no longer applicable.

Each transmission of this command affects a specific logical channel or the entire multiplex. More than one such command may be in effect at the same time, up to the number of open logical channels plus one, for the overall multiplex limitation.

{Ed. This was added for the H.223-based~~PSTN-based~~ system, but I see no reason why it should be disallowed in H.222.0-basedH.222.0-based~~ATM-based~~ systems, in fact it may be beneficial. The units have been chosen to be 100bit/s and the range coded with 24 bits for consistency with H.221 and MPEG.}

6 Procedures

This section defines multimedia system control procedures.

6.1 Point-to-point working

The following procedures relate to point-to-point sessions.

6.1.1 Initial transmission

Prior to receipt of the first DeclaredTermCapSet containing receiveCapSet from the remote end-point, transmission is as specified in the appropriate system recommendation.

6.1.2 Capability exchange

For conversational and speech services, both ends shall send DeclaredTermCapSet messages. The syntax element connectionless shall have the value false to indicate that communication is not connectionless.

On receiving a DeclaredTermCapSet message, a terminal shall acknowledge this by sending an acknowledgement message with acknowledgementType indicating ackDeclaredTermCapSet. It is mandatory to recognise acknowledgement messages.

If, having transmitted a DeclaredTermCapSet message, the acknowledgement is not received within five seconds or an acknowledgement message with acknowledgementType indicating nackDeclaredTermCapSet has been received, the DeclaredTermCapSet message shall be retransmitted, unless a DeclaredTermCapSet message with connectionless equal to true has been received. A DeclaredTermCapSet message with connectionless equal to true indicates connectionless communication; in this case the procedures for point-to-multipoint working apply.

If the terminal has not received acknowledgement of its DeclaredTermCapSet message, and has not received a DeclaredTermCapSet message with connectionless equal to true, after transmitting its DeclaredTermCapSet message three times, it shall send a ChangeOrEndSession message with command equal to endSession and clear down the call.

Action related to any capability value received may not be taken until the complete message is received and validated. Such action may be to transmit a corresponding Mode, or send a request for the remote end-point to do so.

All terminals intended for use in connection-oriented applications shall be able to identify a DeclaredTermCapSet and its structure, and such capability values therein that are mandatory for those applications; any unrecognised capability values shall be ignored, and no fault shall be implied.

6.1.3 Logical channel signalling acknowledged procedures

5 sec → another document

In H.223-basedPSTN-based systems logical channels shall be set up, redefined and closed according to the following procedures. These procedures are optional in H.222.0-basedH.222.0-basedATM-based systems.

{Ed. Perhaps we should consider mandating these procedures in bidirectional connections in H.222.0-basedH.222.0-basedATM-based systems?}

{Ed. In H.222.0-basedH.222.0-basedATM-based systems, should we mandate that if one of these messages is ever sent, then all logical channels shall be handled according to these procedures?}

Before a logical channel can be used, a LogicalChannelSignalling message, with message type equal to channelSetUp, and containing channelUse indicating the data that will be conveyed on the logical channel, shall be sent. Transmission on the logical channel shall not start until an acknowledgement message with acknowledgementType equal to ackLogicalChannel is received. The logical channel is considered to be open from the time that this acknowledgement is received.

Before any change of use of a logical channel, transmission on that channel shall stop, and then a LogicalChannelSignalling message, with message type equal to channelRedefinition, and containing channelUse indicating the data that will be conveyed on the logical channel, shall be sent. Transmission on the logical channel shall not restart until an acknowledgement message with acknowledgementType equal to ackLogicalChannel is received. The logical channel is considered to be open throughout this procedure.

A logical channel may be closed by sending a LogicalChannelSignalling message, with message type equal to channelClose. The logical channel shall be considered to be open until an acknowledgement message with acknowledgementType equal to ackLogicalChannel is received. The transmission of data on the logical channel shall stop before the LogicalChannelSignalling message is sent.

A terminal shall not attempt to open a logical channel that is already open. The use of an open channel can be changed either by closing it and reopening it, or by using the channel redefinition procedure above.

On receiving a LogicalChannelSignalling message, a terminal shall send an acknowledgement message with acknowledgementType equal to ackLogicalChannel if it is able to satisfy the request, taking into account the resources required to process the data that is to be carried on the logical channel in the cases of set up and redefinition. If it is unable to satisfy the request, it shall send an acknowledgement message with acknowledgementType equal to nackLogicalChannel and including LogicalChannelFailure.

If the terminal initiating the request has not received an acknowledgement within five seconds, it may retransmit the request.

Note. Fast mode switching may be achieved by setting up a number of logical channels before they are to be used, and switching from one to another as desired.

{Ed. There may be a problem with this: if a terminal requests logical channels for, say, G.711 and G.728, the far-end terminal may refuse the second request on the grounds that it can not perform both at the same time.}

6.1.4 Mode switching

At any time after receipt of a DeclaredTermCapSet message containing one or more receiveCapSets from the remote end-point, a mode switch may be made - other/changed media streams may be transmitted, provided that they are processable by the remote end-point, as notified by the receiveCapSets within the latest received DeclaredTermCapSet. The change of mode shall follow the above procedures for logical channel signalling as appropriate.

Note. From receipt of the first DeclaredTermCapSet containing receiveCapSet until receipt of a ChangeOrEndSession message with command equal to endSession, all transmitted signals shall be in the range indicated by the receiveCapSets in the latest received DeclaredTermCapSet and by the transmitCapSets, if any, in the latest transmitted DeclaredTermCapSet, that is, valid capsets always replace the previous ones. An end-point shall store the latest received capsets as the "currently valid capsets".

If a terminal is incapable of processing the incoming signals, it shall send an acknowledgement message of type nackLogicalChannel, and including LogicalChannelFailure indicating modeNotDecodable. It shall then send a DeclaredTermCapSet containing one or more receiveCapSets. While it remains incapable of processing the incoming signals, it shall repeat these two messages every five seconds. If it is still incapable of processing the incoming signals after repeating this sequence three times, it shall send ChangeOrEndSession message with command equal to endSession and clear down the call.

6.1.5 Request Mode

If the currently valid received DeclaredTermCapSet message has remoteControllability equal to true, an endpoint may select a mode that it prefers to have transmitted to it, by sending a RequestMode message. An end-point whose latest transmission of DeclaredTermCapSet has remoteControllability equal to true, and which is in receipt of RequestMode, shall comply and shall transmit an acknowledgement message with acknowledgementType indicating ackRequestMode; if for any reason it is unable to do so, it shall send an acknowledgement message with acknowledgementType indicating nackRequestMode and including RequestModeFailure.

A terminal whose most recently sent DeclaredTermCapSet had remoteControllability equal to false shall not receive RequestMode, that is, when remoteControllability is equal to false, the terminal does not wish to, and shall not, be remotely controlled. If such a terminal does receive RequestMode, it may comply and acknowledge as above, but if it doesn't comply, it shall send an acknowledgement message with acknowledgementType indicating nackRequestMode, with noRemoteControllability as the cause of failure.

6.1.6 Change of capability

A changed DeclaredTermCapSet may be sent at any time. An unchanged capset shall not be sent unless:

incoming signals are not processable, that is, they lie outside the range of the transmitted receiveCapSets;

acknowledgement of the most recently transmitted DeclaredTermCapSet has not been received within five seconds;

nackDeclardedTermCapSet was received after the most recent transmission of DeclaredTermCapSet;

a RequestCapability message has been received; or

a fault has occurred (for further study).

If a change of capability has the result that the current mode is no longer receivable/decodable, there shall be a mode switch as soon as possible to a mode that can be received and decoded.

6.1.7 Request Capability

A RequestCapability message may be sent at any time to elicit these details from the remote end-point.

A RequestCapability message may be sent if necessary, following an interruption or other cause for uncertainty; such messages shall not be sent repetitively or otherwise without strong cause.

A terminal that receives a RequestCapability message shall respond by sending a DeclaredTermCapSet message. A terminal that receives a request for transmit capabilities shall comply, but need not send remoteControllability equal to true.

6.1.8 Multiplex table download

This applies only to H.223-based~~PSTN-based~~ systems. It shall not be used in H.222.0-based~~ATM-based~~ systems.

At any time after receiving acknowledgement of a sent DeclaredTermCapSet message, a customized H.223 multiplex table [8] may be downloaded to the remote end-point. Different tables may be used in each direction of transmission. A table is downloaded by sending a MultiplexTableDownload message.

Each multiplex table is referred to by a unique integer, the "table number". Table 0 is the default table, and shall not be modified. The ability to receive additional tables is signalled by MultiplexTableCap. When a multiplex table with a given number is downloaded, it replaces and invalidates any previously downloaded table with the same number.

The table that is currently in use shall not be downloaded. To change the table that is currently in use, it is necessary to select another table, and then download the table after completion of the table selection procedure.

On receiving a MultiplexTableDownload message, a terminal shall acknowledge this by sending an acknowledgement message with acknowledgementType indicating ackMultiplexTableDownload. It is mandatory to recognise acknowledgement messages. If the MultiplexTableDownload message can not be understood, or is outside the range of tables that the terminal can use, it shall send an acknowledgement message with acknowledgementType equal to nackMultiplexTableDownload and indicating the cause of the failure.

6.1.9 Multiplex table selection

This applies only to H.223-based~~PSTN-based~~ systems. It shall not be used in H.222.0-based~~ATM-based~~ systems.

At any time after receiving acknowledgement of a MultiplexTableDownload message, the downloaded table may be selected for use by sending a selectMultiplexTable type of H223C&I type of C&I message.

On receiving a selectMultiplexTable command, a terminal shall acknowledge this by sending an acknowledgement message with acknowledgementType indicating ackSelectMultiplexTable. It is mandatory to recognise acknowledgement messages. If the selectMultiplexTable command can not be understood, or if the table has not been downloaded, it shall send an acknowledgement message with acknowledgementType equal to nackSelectMultiplexTable and indicating the cause of the failure.

The terminal shall stop using the current multiplex table from the time that it sends the selectMultiplexTable command, and shall not start to use the newly selected table until it has received an acknowledgement message with acknowledgementType indicating ackSelectMultiplexTable. In the time between these events it may use only those MUX codes that are identical in both tables.

If, having transmitted a selectMultiplexTable command, the acknowledgement is not received within five seconds or an acknowledgement message with acknowledgementType indicating nackSelectMultiplexTable has been received, the terminal may either transmit the same or a different selectMultiplexTable command and follow the above procedure again; or revert to using the previous multiplex table.

{Ed. Is this an acceptable failure procedure? There will be problems if a terminal is very slow to respond, but then it can send a nackSelectMultiplexTable and recover.}

6.1.10 Flow control

A terminal may send a FlowControl command to restrict the bit rate that the far-end terminal sends.

A terminal that receives a FlowControl command shall comply with the command if it is capable of doing so.

{Ed. Do we need acknowledgements here? What if the rate can not go lower, for example commanding G.723 to go at 100 bit/s would be impossible?}

6.1.11 Control and indication messages

At any time after receiving acknowledgement of a sent DeclaredTermCapSet message, other messages may be sent, for example, C&I and UserInputAndOther messages.

A terminal may send a vcuGOB command to the far-end H.261 or H.263 video encoder to perform a fast update of one or more GOBs, following an interruption or other cause for uncertainty; such commands shall not be sent repetitively or otherwise without strong cause. A terminal that receives a vcuGOB command shall comply as soon as it is practical to do so.

6.1.12 Change or end of session

If it is desired to change from conversational to another class, a ChangeSession message shall be sent.

Change of service class is for future study.

Prior to clearing down, an end-point shall transmit EndSession, and thereafter transmit no multimedia system control messages, and ignore any received.

6.1.13 Encryption and Key Management

The procedures for Encryption and Key Management are specified in the appropriate system recommendation.

6.1.14 Round-trip delay determination

{Ed. Write something here.}

6.2 Point-to-Multipoint working

Where one source is feeding several receivers, it will be unable (and does not try) to respond to any upstream signals, whether of the ReceiveCapSet type or requests to transmit in a particular medium/mode, RequestMode, etc. The source shall transmit its TransmitCapSets periodically (eg less than T1 intervals), including only the values actually in use in that session, together with RequestModeFailure and DeclaredTermCapSet with connectionless equal to true. The TransmitCapSet shall not contain any options (NB: this does not mean that it cannot send two or more video signals in different formats, for example).

{Ed. RequestModeFailure should only be sent in case of receipt of RequestMode?}

{Ed. What does TransmitCapSet shall not contain any options mean?}

A receiving terminal shall recognise the DeclaredTermCapSet with connectionless equal to true as indicative of a connectionless communication, and shall transmit no further messages. It shall, however, remain sensitive to incoming DeclaredTermCapSets which could indicate a change of orientation.

ANNEX A

(This annex forms an integral part of this Recommendation)

Protocol Stack for H.222.0-basedATM-based systems

This annex defines two protocol stacks for use with this Recommendation for H.222.0-basedATM-based systems.

{Ed. The definitions of how to use the protocol layers has assumed connected-oriented applications. If this Recommendation is to be used for connection-less (broadcast) applications, further text will be needed.}

A.1 General

Figure A.1/H.245 shows two protocol stacks for use with this Recommendation for H.222.0-basedATM-based systems. Two modes of operation are envisaged. Figure A.1(a)/H.245 shows the protocol stack when the messages defined in this Recommendation are transported in a separate ATM VC to that used for multimedia data. Figure A.1(b)/H.245 shows the protocol stack when the messages defined in this Recommendation are transported in the same ATM VC as that used for multimedia data.

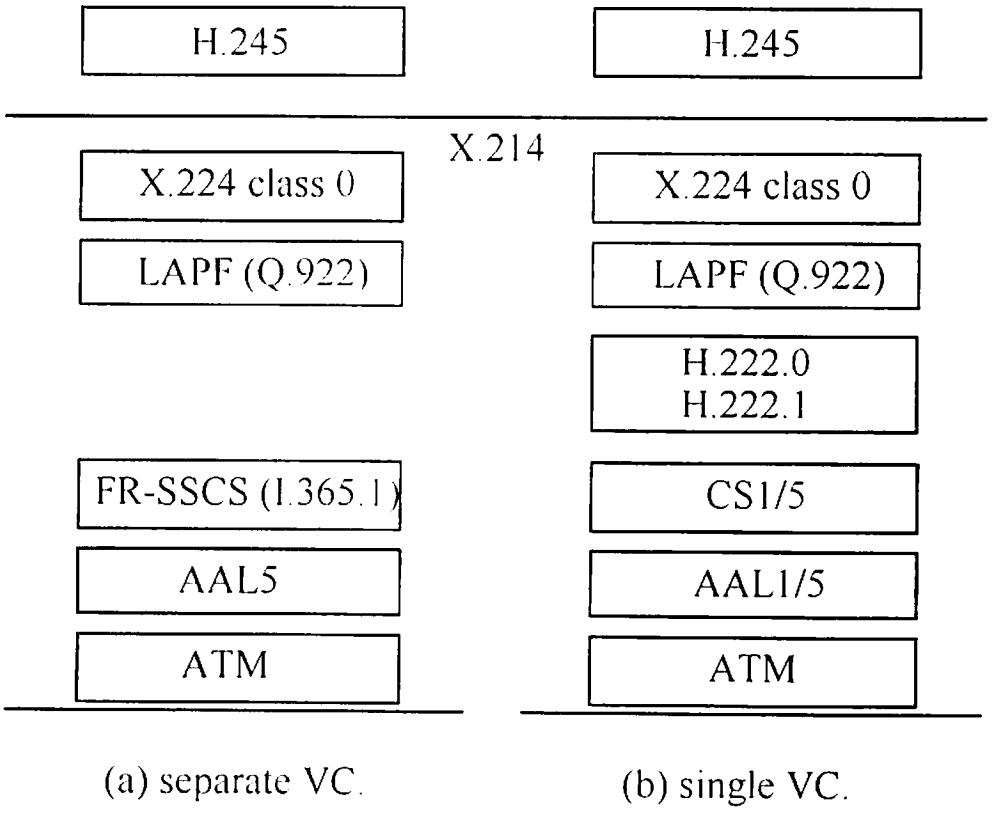

Figure A.1/H.245. H.245 Protocol stack for H.222.0-basedATM-based systems.

A.2 X.214

A connection-mode service shall be used for the transport of MultimediaSystemControl messages as defined in X.214 [23].

A single transport connection (TC) shall be used for the transport of MultimediaSystemControl messages.

{Ed. X.214 allows one or more TCs between the same pair of TS users, but to allow this seems unnecessarily complicated as message transfer would no longer be serial.}

The encoded representation of individual MultimediaSystemControl messages shall be transported in single transport service data units (TSDUs). The bits produced by the ASN.1 encoding process shall be put into TSDU octets, with the first bit generated going into the most significant bit (MSB) of the octet, and progressing down to the least significant bit (LSB).

{Ed. Although this is strictly not necessary, I beleive that this would increase the error resilience of transmission as the loss, corruption or insertion of TSDUs would not result in loss of synchronisation/alignment at the ASN.1 decoding layer.}

A.3 X.224

X.214 service shall be provided by X.224 class 0 [24].

A.4 LAPF (Q.922)

Q.922 [15] shall be used to provide a network connection of Type A defined in clause 5.4.3 of X.224, that is, a connection with acceptable residual error rate and acceptable rate of signalled errors. This shall be done by use of information (I) frames.

A.5 FR-SSCS (I.365.1) and H.222

Both FR-SSCS (I.365.1) and H.222 [6] provide octet transmission with structure. The transparency procedure specified in clause 2.6 of Q.922 shall not be used, that is, no zero bits shall be inserted after any sequence of five consecutive 1 bits. No flags shall be present.

A.5.1 FR-SSCS (I.365.1)

{Ed. Need to say something similar here to that for H.222, but I have not yet read I.365.1.}

A.5.2 H.222.0 and H.222.1

The octets of Q.922 shall be transported in PES packets defined in H.222.0 so that the first octet of a PES packet is the first octet of the address field and the last octet of the PES packet is the last octet of the FCS field, as defined in clause 2.2 of Q.922.

{Ed. I have changed this so that flags are not transported, and PES packets are used to convey the Q.922 structure.}

ANNEX B

(This annex forms an integral part of this Recommendation)

Protocol Stack and initialisation procedures for H.223-basedPSTN-based systems

This annex defines one protocol stack for use with this Recommendation for H.223-basedPSTN-based systems.

B.1 General

Figure B.1/H.245 shows the protocol stack for use with this Recommendation for H.223-basedPSTN-based systems.

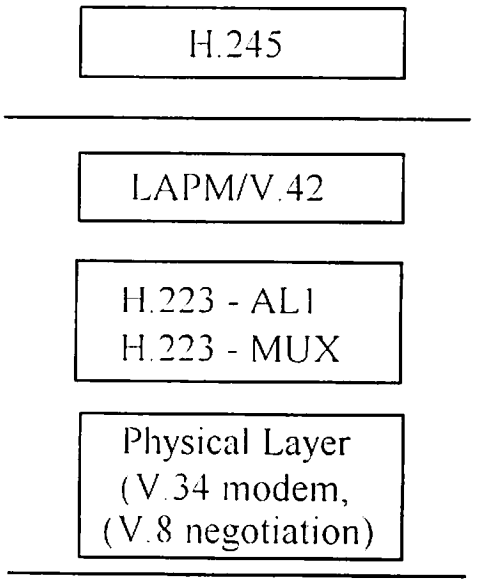

Figure B.1/H.245. H.245 Protocol stack for H.223-basedPSTN-based systems.

B.2 LAPM/V.42

The Link Access Procedure for Modems (LAPM) error control protocol defined in V.42 [21] shall be used.

The transparency procedure specified in clause 8.1.1.2 of V.42 shall not be used, that is, no zero bits shall be inserted after any sequence of five consecutive 1 bits. No flags shall be present. The frame structure is shown in figure B.2/H.245.

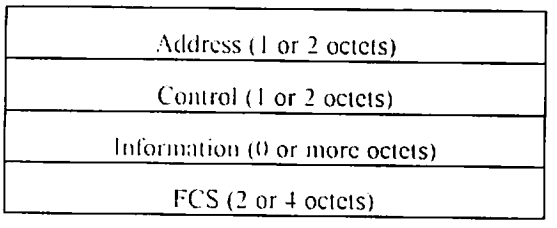

Figure B.2/H.245. Frame structure for LAPM/V.42 AL-SDUs.

Address field shall be one octet with the 6-bit DLCI field set to 000000.

The frame check sequence field shall have length 16 bits.

{Ed. These two statements came from H.246. Do we need to say this?}

Two means of transporting MultimediaSystemControl messages using LAPM/V.42 are defined: XID frames and Information frames. In both cases, bits produced by the ASN.1 encoding process shall be put into LAPM octets, with the first bit generated going into the most significant bit (MSB) of the octet, and progressing down to the least significant bit (LSB).

B.2.1 XID frames

All terminals shall support the transfer of MultimediaSystemControl messages using XID frames.

The format identifier 'xxxxxxxx' shall be used to indicate XID frames containing MultimediaSystemControl messages.

{Ed. The value of this is tbd.}

{Ed. It is not clear how the ASN.1 encoded messages fit into the group and parameter structure of XID frames. The following is a very tentative suggestion: alternatives include GIs being arbitrary, and being fixed at one value independent of the message.}

The encoded representation of individual MultimediaSystemControl messages shall be transported in single XID groups, with the Group Identifier (GI) equal to application label of the included message, as defined in clause 5.1. This is shown in figure B.3/H.245.

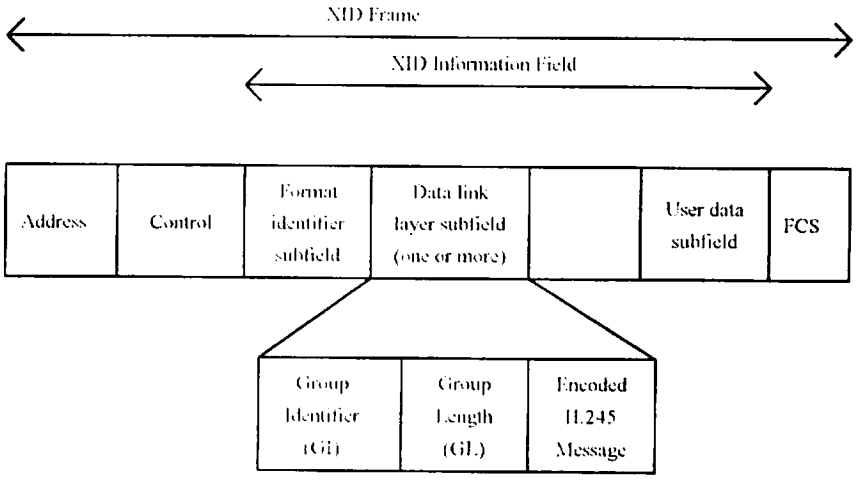

Figure B.3/H.245. Format of XID information field for XID frames with MultimediaSystemControl messages.

XID frames containing MultimediaSystemControl messages shall be considered to be commands for the purpose of setting the C/R (command/response) bit in the address field. Receipt of a valid command XID frame shall be acknowledged by the return of an XID response frame. The contents of the information field of the response frame is outside the scope of this Recommendation, and may be null.

Note. An changed XID frame can not be transmitted until acknowledgement of the previous is received, as the response does not identify the frame to which it refers.

B.2.2 Information (I) frames

Terminals may support the transfer of MultimediaSystemControl messages using Information (I) frames.

The procedures specified in V.42 are used to determine whether there is capability to support I frame transmission and to initiate this.

XID frames may be used to transfer MultimediaSystemControl messages before I frame transmission is initiated and after it is terminated.

{Ed. Both mechanisms could be used simultaneously, should be forbid this, or say nothing?}

The encoded representation of individual MultimediaSystemControl messages shall be transported in single I frames.

B.3 H.223

H.223 [8] provides octet transmission with structure.

Data adaptation layer Type 1 shall be used to transfer LAPM/V.42 frames between the LAPM/V.42 protocol and the MUX layer of H.223. The Packet marker field shall be used to mark the end of LAPM/V.42 frames.

APPENDIX I

(This appendix does not form an integral part of this Recommendation)

Examples of DeclaredTermCapSet messages

{Ed. This appendix will contain some examples of DeclaredTermCapSet messages. The text below was previously in the main part of the document.}

{Ed. Example for clarification of intentions: A terminal that can process G.711 and QCIF, G.728 and QCIF, or G.711 and CIF simultaneously, but not G.728 and CIF simultaneous could indicate any of the following.

independent TermCapSet 1 :	G.711, G.728, QCIF
independent TermCapSet 2 :	G.711, QCIF, CIF
dependent TermCapSet 1 :	G.711, CIF
dependent TermCapSet 2 :	G.728, QCIF

Note: the inclusion of, say, G.711 and G.728 in a dependent capability would indicate that both can be performed simultaneously.}

resource id.

a typical case be desired □

APPENDIX II

(This appendix does not form an integral part of this Recommendation)

Overview of ASN.1 syntax

{Ed. This appendix has been provided by Dan Helman. The revision marks have been removed to improve readability.}

II.1 Introduction to ASN.1

Abstract Syntax Notation One (ASN.1) is a data specification language. Itwas originally standardized as part of the X.400 electronic mail series as X.409. This evolved to X.208 and most recently X.680. ASN.1 allows unambiguous specification of complex data structures including those with variable-length fields, optional fields and recursion.

The above Recommendations deal only with the syntax and semantics of ASN.1 specifications. The binary encoding of data structures is covered in other Recommendations, notably X.690 (basic encoding rules or BER) and X.691 (packed encoding rules or PER). BER allows data to be deciphered by systems that have general knowledge of ASN.1 but do not know the details of the specification used to form the data. In other words, the data types are encoded along with the data values. PER is much more efficient since only data values are encoded and the coding is designed with very little redundancy. This method can be used when both the transmitter and the receiver expect data to adhere to a known structure.

H.245 is implemented using the packed encoding rules. Since both sides of a call know that messages will conform to the H.245 specification it is not necessary to encode that specification into the messages. For decoding simplicity, the aligned variant of PER is used. This forces fields that require eight or more bits to be aligned on octet boundaries and to consume an integral number of octets. Alignment is done by padding the data with zeros before large fields.

II.2 Basic ASN.1 data types

The simplest data type is BOOLEAN, which represents the values FALSE and TRUE. These are encoded in a single bit as 0 and 1 respectively. For example,

Definition	BOOLEAN	
Encoding	FALSE	0
	TRUE	1

The most fundamental data type is INTEGER, which represents whole number values. Integers can be unconstrained as in:

```
videoBitRate    INTEGER
```

or they can be constrained to a range of values, for example

```
muxPDUSize    INTEGER (0..65535)
```

Constrained integers are encoded differently depending on the size of the range. Suppose N is the number of integers in the range, i.e. theupper limit minus the lower limit plus one. Depending on N, the constrained integer will be encoded in one of five ways:

N	Encoding

1	no bits needed
2-255	an unaligned field of 1 to 8 bits
256	an aligned 8-bit field
257-65536	an aligned 16-bit field
larger	as the minimum number of aligned octets preceded by the above encoding of the number of octets

In all cases, the number that is actually used is the value to be encoded minus the lower limit of the range. In these examples "pad" represents zero to seven 0 bits that are added to the encoding so that the following field will start on a 8-bit boundary.

Definition	adaptationLayerType	INTEGER (0..7)
Encoding	0	000
	3	011
Definition	multiplexTableNesting	INTEGER (0..255)
Encoding	3	pad 00000011
	254	pad 11111110
Definition	skew	INTEGER (0..4095)
Encoding	3	pad 00000000 00000011
	4095	pad 00001111 11111111

Unconstrained integer (2's complement) values that can be represented in 127 octets or less are encoded in the minimum number of octets needed. The number of octets (the length) is encoded as an aligned octet that precedes the number itself. For example,

Encoding	-1	pad 00000001 11111111
	0	pad 00000001 00000000
	128	pad 00000010 00000000 10000000
	1000000	pad 00000011 00001111 01000010 01000000

The ENUMERATED data type represents a number chosen from a specific set of values. New values can be added later by putting an ellipsis (extension marker) at the end of the list. For example,

```
TableDownloadFailure ::= ENUMERATED
{
  unspecifiedCause   (0),
  invalidTable (1),
  tableOutOfRange    (2),
  ...
}
```

Encoding ENUMERATED data is simple. If the extension marker is present one bit is output to indicate if the selection is from the original list (extension root) or from the added values. The N values in the enumeration are sorted

and assigned an index from 0 to N-1. The appropriate index is then encoded as a constrained integer. To continue the above example,

Encoding	0	000
	2	010

ASN.1 supports a variety of string data types. These are variable-length lists of bits, octets or other short data types. They are typically encoded as a length followed by the data. The length can be encoded as an unconstrained integer or as a constrained integer if the SIZE of the string is specified. For example,

```
yearOfRecommendation        NumericString (SIZE(4) FROM("0123456789"))
```

In this case the size of the string is fixed at four (digits). Each digit will be encoded as its index into the permitted alphabet, in this case a 4-bit number from 0 to 9. Since the total number of bits in the string is not greater than sixteen, the four 4-bit fields will not be octet-aligned.

```
h263Modes     BIT STRING
              {
                unrestrictedVector      (0),
                arithmeticCoding        (1),
                advancedPrediction      (2),
                pbFrames                (3),
                ...
              }
```

This bit string is a fixed size. It will start with one bit for the extension marker followed by the four data bits. No length information is encoded (unless the extension field is present).

```
nonStandard    OCTET STRING
```

Since the length of the octet string is not bounded, it will have to be encoded as a "semi-constrained whole number" (has a lower bound, but no upper bound). First, the data is padded so that the encoding will be aligned. The rest of the code is as follows:

length	encoding
0 to 127	8-bit length followed by the data
128 to 16K-1	16-bit length with the MSB set, then the data
16K to 32K-1	11000001, 16K octets of data, then code the rest
32K to 48K-1	11000010, 32K octets of data, then code the rest
48K to 64K-1	11000011, 48K octets of data, then code the rest
64K or more	11000100, 64K octets of data, then code the rest

This method is called "fragmentation". Note that if the length is a multiple of 16K, then the representation will end with an octet of zero indicating a zero-length string.

II.3 Aggregate data types

ASN.1 includes several aggregate or container data types that are similar in concept to C's union, struct and array types. These are, respectively, CHOICE, SEQUENCE and SEQUENCE OF. In all cases they are encoded with some bits specific to the container followed by the normal encodings of the contents.

CHOICE is used to select exactly one of a group of data types. For example,

```
VideoCap        ::= CHOICE
{
   nonStandard          [0] IMPLICIT OCTET STRING,
   h261VideoCap         [1] IMPLICIT H261VideoCap,
   h262VideoCap         [2] IMPLICIT H262VideoCap,
   h263VideoCap         [3] IMPLICIT H263VideoCap,
   ...
}
```

The values in brackets are called tags. They are just used to order the choices so that they can be unambiguously encoded (the word IMPLICIT only has meaning for the basic encoding rules, not PER). The tags are sorted, and index numbers are assigned to each choice. The index of the actual data is encoded as a constrained integer. The index is followed by the encoding of the actual selection. If the extension marker is present (as above), the index is preceded by a bit that is zero if the actual choice is from the original list.

SEQUENCE is simply a grouping of dissimilar data types. Individual elements of the sequence may be OPTIONAL. The encoding is very simple. If there is an extension marker the first bit indicates the presence of additional elements. This is followed by a series of bits, one for each optional element that indicates if that data is present. This is followed by the encodings of the components of the sequence. For example,

```
H263VideoCap          ::= SEQUENCE
{
   sqcifMPI           INTEGER (1..16) OPTIONAL,
   qcifMPI            INTEGER (1..16) OPTIONAL,
   cifMPI             INTEGER (1..16) OPTIONAL,
   h263Modes          BIT STRING,
                   { ... as above ... },
   ...
}
```

The encoding has one bit for the extension marker, three bits for the optional fields, four bits each for any optional field that is present, five bits for the bit string and then any extension data. Note that in this sequence has no padding for octet alignment.

The SEQUENCE OF type describes a collection of similar components (an array). It can have a SIZE constraint or an unconstrained number of elements. If the number is known a priori and is less than 64K, it is not encoded. Otherwise the actual number of components in the sequence is encoded as a constrained or semi-constrained length. This is followed by the encoding of the data. If the length is at least 16K and is encoded then the list of data will be broken into fragments like the octet string. In this case the fragments are broken after some number of component fields (16K, 32K, etc), not after some number of octets. For example,

```
nonStandard    SEQUENCE OF OCTET STRING
```

Here the number of strings is arbitrary, so it will be encoded as an octet-aligned length (8 bits, 16 bits or fragments) followed by that number of arbitrary-length octet strings.

APPENDIX X

(This appendix does not form an integral part of this Recommendation)

Outstanding editorial and technical work

{Ed. This appendix is temporary. It specifies the outstanding editorial and technical work.}

Aspect Ratio capability signalling and requests: would seem beneficial to include this for H.222.0-based~~ATM-based~~ systems in line with MPEG video. Could also be used for PSTN?

T.84 and T.434 codepoints need to be added for theH.223-based~~PSTN-based~~ case. What is T.84? If these points are to be of general use, I think more information is needed, such as protocol stack. Can interested people please contribute text.

PSTN adaptation layer capabilities: do these need to be signalled in the capability sets, and if so, how? Defaults already exist, and operation in other modes that are not supported and prevented by the logical channel set up procedure.

The use of ASN.1 for the PSTN multiplex table description. This would allow it to be intergrated with the control messages and would allow more efficient transmission of tables. Problems include recursive ASN.1 syntax, specifying the default table and limiting the size of tables.

Multi-link capability in PSTN: messages need to be defined. Applicability to multiple VCs in the ATM case could also be considered.

Fast mode switching.

Round Trip delay determination.

END