Telecommunication Standardization Sector
Original: English
(TSS)

AVC-890v2

Experts Group for Video Coding and Systems in ATM and Other Network Environments

January 16-25, 1996

STUDY GROUP 15
CONTRIBUTION

Source: D. Skran, Editor H.225.0

Title: H.245 Additions for H.323

Date: January 12, 1996

Revision History:

- Rev 2: Jan 17th
    1. Removed proposed H320 loopback commands. Loopbacks requested from a H.320 (H.310, H.324, etc.) terminal will be terminated at the Gateway.
    2. Changed all addresses to type NetworkAddress as defined in H.225
    3. Changed multipointModeSymmetrize to Command as in H.245
    4. Added our extended TerminalCapability Set changes to the H.225 multiplex capability. Changed the conference model specs to centralized vs. distributed parameters for audio, video, data, and control. Added a multicast capability. Added a maximum delay jitter.
    5. Standardized changes to italics non bold, existing H.245 commands in non bold and normal text.

1. Introduction

{Editors Note: It should be noted that these additions are essential for the proper operation of H.323. These changes have not been reviewed and should not be considered final. There have been many recent suggestions for additional H.245 changes; there are also not included but will be considered at Ipswich}

Some of these changes may also be useful to H.324 and H.310 terminals. The primary goal of the H.323 experts is to promote full interoperability with existing and future H.320 terminals.

This proposal contains only proposed additions to H.245 as needed changes have already been put in to the decided version of H.245. New text is italicized non-bold. It is recognized that in some cases needed H.245 text is missing; the ASN.1 is believed sufficient to convey the nature of the additions needed and the text will be provided later.

2. Proposed Additions to H.245 For the First Revision

2.1 PDUs Related to Interworking with H.320

All H.245 PDUs address the H.245 peer which maybe a H.323 terminal, gateway or MCU. Some devices, such as gateways or MCUs may pass on commands to the remote endpoint. Areas where this is needed include ns-cap/ns-com.

{Editor's Note: The general approach described here for dealing with the gateway for encryption and ns-cap/ns-com was not agreed to at the SG15/WP1 meeting by the H.245 experts present. Fortunately, they will all be at Ipswich and hopefully we can come to some conclusion}

PDUs for Ns-cap/Ns-coms and CIC-cap

```
Capability          ::=CHOICE
{
        nonStandard                                NonStandardParameter,
        nonStandardH320                            NonStandardParameter,
        miscellaneous Capability                   MiscellaneousCapability,

        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
```

TD-27

```
                ...
        },

        ...
    }


        NonStandardPDU                          ::=SEQUENCE
        {
            nonStandardData                     NonStandardParameter,
            nonStandardH320Data                 NonStandardParameter,
            ...
        }

        MiscellaneousCapability                 ::=SEQUENCE
        {
            chairControlCapability              BOOLEAN,
            ...
        }
```

When nonStandardH320Data is sent, the gateway sends the information on to the H.320 side rather than acting on the data itself. When nonStandardH320 capability is sent, the H.323 gateway passes the capability on to the H.320 terminal as an ns-cap.

```
DataProtocolCapability                  ::=SEQUENCE
{
    nonStandard                         NonStandardParameter,
    v14buffered                         BOOLEAN,
    v42lapm                             BOOLEAN,        -- may negotiate to V.42bis
    hdlcFrameTunneling                  BOOLEAN,
    transparent                         BOOLEAN,
    v120                                BOOLEAN,        -- as in H.230
    ...
}


MiscellaneousCommand                    ::=SEQUENCE
{
    logicalChannelNumber                LogicalChannelNumber, (how do we indicate all or none??)
    type                                CHOICE
    {
        equaliseDelay                       NULL,       -- like H.230 ACE
        zeroDelay                           NULL,       -- like H.230 ACZ

        multipointModeCommand               NULL,
        cancelMultipointModeCommand         NULL,
        terminalIdRequest                   NULL,       -- translates into TCI or TCS-2
        terminalListRequest                 NULL,       -- same as H.230 TCU

        broadcastMe                         NULL        -- same as H.230 MCV
        cancelBroadcastMe                   NULL,       -- same as Cancel-MCV


        makeTerminalBroadcaster             SEQUENCE            -- same as H.230 VCB
        {
            mcuNumber                       INTEGER(0..192),
            terminalNumber                  INTEGER(0..192),
            ...
        }
        cancelMakeTerminalBroadcaster    NULL,                --same as H.230 cancel-VCB

        sendThisSource                   SEQUENCE             -- same as H.230 VCS
        {
            mcuNumber                    INTEGER(0..192),
            terminalNumber               INTEGER(0..192),
            ...
        }
        cancelSendThisSource             NULL,                --same as H.230 cancel-VCS

        dropTerminal                     SEQUENCE             -- same as H.230 CCD
        {
            mcuNumber                    INTEGER(0..192),
            terminalNumber               INTEGER(0..192),
            ...
        }

        makeMeChair                  NULL,                    -- same has H.230 CCA
        cancelMakeMeChair            NULL,                    -- same as H.230 CIS

        dropConference               NULL,                    -- same as H.230 CCK

        enterH243Password            NULL,                    -- same as H.230 TCS1
        enterH243TerminalId          NULL,                    -- same as H.230 TCS2
        enterH.243ConferenceId       NULL,                    -- same as H.230 TCS3

        requestTerminalId            NULL,                    -- same as TCP
        {
            mcuNumber                    INTEGER(0..192),
            terminalNumber               INTEGER(0..192),
            ...
        }

        videoFreezePicture           NULL,
        videoFastUpdatePicture       NULL,

        videoFastUpdateGOB           SEQUENCE
        {
                                     firstGOB        INTEGER (0..17),
                                     numberOfGOBs    INTEGER (1..18)
        },

        videoTemporalSpatialTradeOff INTEGER (0..31),         -- commands a trade-off value

        videoSendSyncEveryGOB        NULL,
        videoSendSyncEveryGOBCancel  NULL,

        ...
    },

    ...
}

TerminalIdResponse               ::=SEQUENCE
{
    terminalId                   OCTET STRING,            -- as per H.230
    ...
}
```

This could be used to respond to TCP, TCS1/2/3 - but is this the H.245 way? Suggestions?? This would make it equivalent to TIP and IIS.

```
TerminalListResponse                ::=SEQUENCE
{
    terminalNumbers                 SEQUENCE,               -- as per H.230 TIL
    {
        mcuNumber                   INTEGER(0..192),
        terminalNumber              INTEGER(0..192)
    }
    ...
}

VideoCommandReject                  ::=NULL                 -- same as H.230 VCR

MakeMeChairResponse                 ::=NULL                 -- same as H.230 CCR
{
            grantedChairToken       NULL,                   -- same as H.230 CIT
            deniedChairToken        NULL,                   -- same as H.230 CCR
            ...
}

MiscellaneousIndication                   ::=SEQUENCE
{
    logicalChannelNumber                  LogicalChannelNumber,
    type      CHOICE
    {
        logicalChannelActive              NULL,             -- like H.230 AIA and VIA
        logicalChannelInactive            NULL,             -- like H.230 AIM and VIS

        multipointConference              NULL,             --

    cancelMultipointConference            NULL,             --

        multipointZeroComm                NULL,             -- like H.230 MIZ
        cancelMultipointZeroComm          NULL,             -- like H.230 cancel MIZ

        multipointSecondaryStatus         NULL,             -- like H.230 MIS
        cancelMultipointSecondaryStatus   NULL,             -- like H.230 cancel MIS

        videoIndicateReadyToActivate      NULL,             -- like H.230 VIR

        sbeNumber                         INTEGER (0..9),   -- same as H.230 SBE Number

        terminalNumberAssign              SEQUENCE          -- same as H.230 TIA
        {
            mcuNumber                     INTEGER(0..192),
            terminalNumber                INTEGER(0..192),
            ...
        }

        terminalJoinedConference          SEQUENCE          -- same as H.230 TIN
        {
            mcuNumber                     INTEGER(0..191),
            terminalNumber                INTEGER(0..191),
            ...
        }

        terminalLeftConference            SEQUENCE          -- same as H.230 TID
        {
            mcuNumber                     INTEGER(0..192),
            terminalNumber                INTEGER(0..192),
            ...
        }

        seenByAtLeastOneOther          NULL,              -- same as H.230 MIV
        cancelSeenByAtLeastOneOther    NULL,              -- same as H.230 cancel MIV

        seenByAll                      NULL,              -- like H.230 MIV
        cancelSeenByAll                NULL,              -- like H.230 MIV

        terminalYouAreSeeing           SEQUENCE           -- same as H.230 TIN
        {
            mcuNumber                  INTEGER(0..192),
            terminalNumber             INTEGER(0..192),
            ...
        }

        requestForFloor                NULL,              -- same as H.230 TIF

        videoTemporalSpatialTradeOff   INTEGER (0..31),   – indicates current trade-off

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

2.2 PDUs to distribute multicast information

The CommunicationModeCommand is sent by the H.323 MC to specify to the terminal the communication mode - unicast or multicast. If the communicaton mode is multicast, multicast addresses and their associated port numbers and data types are also specified. This command may cause a switch between the centralized model and the decentralized model. A switch may involve closing all existing logical channels and opening new ones. The CommunicationModeRequest is sent by the terminal to the MC to request for the communication mode. The CommunicationModeResponse is sent by the MC to the terminal, in response to the terminal's request, to specify the communicaton mode and its specifics. Note that the MC could potentially have different communication modes with different terminals

Used by the H.323 MC to distribute multicast addresses.

```
CommunicationModeCommand                  ::=CHOICE
{
    unicast                             NULL,
    multicastTable                        SET SIZE (1..256) OF MulticastTableEntry,
     ...
}
```

Used by the H.323 terminal to request the communication mode (multicast vs unicast)

```
CommunicationModeRequest                  ::=NULL
{
      ...
}
```

Used by the H.323 MC to respond to the CommunicationModeResponse

```
CommunicationModeResponse                        ::=CHOICE
{
      unicast                             NULL,
      multicastTable                        SET SIZE (1..256) OF MulticastTableEntry
       ...
}
```

Used by the H.323 MC to distribute multicast addresses and port numbers for their associated data types

```
MulticastTableEntry                         ::=SEQUENCE
{
      DataType                              CHOICE
      {
            videoData                       NULL,
            videoControl                    NULL,
            audioData                       NULL,
            audioControl                    NULL,
            data                            NULL,
            ...
      }
      multicastAddress                      NetworkAddress,
       ...
}
```

```
MCLocationCommand    ::==SEQUENCE
{
       signalAddress          NetworkAddress,
        ...
}
```

2.2 Extensions to fields in existing PDUs

Additions are shown in *italics* and **bold**.

2.2.1 Extend TerminalCapabilitySet Request PDU

The terminal must be capable of centralized control, audio, video and data processing. A multicast-capable terminal is capable of sending and receiving multicast packets but it cannot guarantee an end-to-end multicast infrastructure. If the terminal does not receive multicast streams, it should present its new capabilities to the MC.

```
MultiplexCapability            ::CHOICE
{
       nonStandard             NonStandardParameter,
       h222Capability          H222Capability,
       h223Capability          H223Capability,
       vGMUXCapability         VGMUXCapability,
       h225Capability          H225Capability,
        ...
}
```

```
H225Capability                 ::=SEQUENCE
{
        maximumDelayJitter            INTEGER (0..1023)        -- units in milliseconds
        multicastCapability           BOOLEAN,
        ConferenceCapability          ::=SEQUENCE
        {
                mcCapability                  ::=SEQUENCE
                {
                centralizedConferenceMC    BOOLEAN OPTIONAL,
                multicastConferenceMC      BOOLEAN OPTIONAL,
                ...
                }
                centralizedControl      BOOLEAN,
                distributedControl      BOOLEAN,
                centralizedAudio        BOOLEAN,
                distributedAudio        BOOLEAN,
                centralizedVideo        BOOLEAN,
                distributedVideo        BOOLEAN,
                centralizedData         BOOLEAN,
                distributedData         BOOLEAN,
                ...
        }
        ...
}
```

maximumDelayJitter indicates the maximum peak-to-peak transmission jitter that the transmitter shall cause. It is measured in milliseconds. Transmission jitter is defined as the difference in time of delivery of each audio packet to the network compared to when it would be delivered at a constant bit rate without packetization.

2.2.2 Extend AudioCapability Description

In the comments before Audio Capability, add the following comment:

-- For an H.225 multiplex, the integers indicate the maximum number of audio frames per packet.

{Note: The size of a frame is defined in H.225.}

2.2.3 Extend OpenLogicalChannel Request PDU for RTCP Channel

Note that the RTP headers may change before the receiver processes the logical channel re-open; thus the RTP header is considered the controlling information concerning mode. This issue and its implications require further discussion.

Two data types (video control and audio control) are added to include the video RTCP and the audio RTCP transport connections. The MediaControlCapability associates the RTCP logical channel to the RTP logical channel.

The h225LogicalChannelParameters is added to specify the fields required in the LAN protocols (eg.,TCP, IP). If the logical channel refers to a unicast transport connection then the UnicastParameters include either a guaranteed (eg., TCP) or a non-guaranteed (eg., UDP) transport connection. However, if the logical channel refers to a multicast transport connection then the MulticastParameters include the multicast address and the port number supplied previously by the MC in the CommunicationModeCommand. Those terminals that wish to open a new physical multcast channel should put zeroes in the MulticastParameters fields. The MC will fill in the new multicast address and port number in the OpenLogicalChannelAck PDU and distribute these addresses to all the other terminals in the , open a logical channel to the MC for the specified physical multicast transport connection.

In the centralized conference, each logical channel maps to a unique physical transport connection between the terminal and the MC. In the decentralized conference, many logical channels between the terminals and the MC will map to a single physical multicast transport connection. Even though the logical channels are opened between the MC and the terminal, the physical data flow for a multicast transport connection is not from the terminal to the MC but essentially a bus from which any terminal can transmit or receive data. The MC opens logical channels to all capable terminals, capability determined during capability exchange, in the conference to give them the ability to receive audio/video on the specified physical multicast transport connection. Those terminals that wish to transmit, open a logical channel to the MC for the specified physical multicast transport connection.

```
DataType                              ::=CHOICE
{
    nonStandard                       NonStandardParameter,
    nullData                          NULL,
    videoData                         videoCapability,
    audioData                         audioCapability,
    data                              DataApplicationCapability,
    encryptionData                    encryptionMode,
    videoControl                      MediaControlCapability,
    audioControl                      MediaControlCapability,
    ...
}

MediaControlCapability  ::=SEQUENCE
 {
  AssociatedLogicalChannel      INTEGER(1..65535),
   ...
 }

LogicalChannelMultiplexParameters     ::=CHOICE
{
    h222LogicalChannelParameters      222LogicalChannelParameters,
    h223LogicalChannelParameters      223LogicalChannelParameters,
    h225LogicalChannelParameters      h225LogicalChannelParameters,
    ...
}

h225LogicalChannelParameters              ::=CHOICE
{
    unicastChannel                  UnicastParameters,
    multicastChannel                MulticastParameters,
}

UnicastParameters               ::=SEQUENCE
{
    GuaranteedDelivery                    BOOLEAN       -- select reliable transport
}

MulticastParameters              ::=SEQUENCE
{
    multicastAddress            NetworkAddress,
    ...
}

NetworkAddress          ::=CHOICE
{
        IPAddress               SEQUENCE
        {
                transport       OCTET STRING (SIZE(4)),
                port            INTEGER(0..4294967295),
                ...
        },
        IPXAddress              SEQUENCE,
        {
                node            OCTET STRING (SIZE(6)),
                netnum          OCTET STRING (SIZE(4)),
                port            OCTET STRING (SIZE(2)),
                ...
        },
        IP6Address              SEQUENCE,
        {
                transport       OCTET STRING (SIZE(16)),
                port            INTEGER(0..4294967295),
                ...
        },
        NetBios                 OCTET STRING (SIZE(16)),
        ...
}
```

2.2.4 Extend OpenLogicalChannelACK Response PDU

In the OpenLogicalChannelAck the receiving terminal sends the port number that it is listening on to the transmitting terminal if the communication mode is unicast. If the communication mode is multicast, the receiver also sends the multicast address in addition to the port number.

In the multipoint and multicast communication mode, the transmitting terminal uses the same multicast address and port numbers provided by the MC in the CommunicationModeCommand and the receiving terminal fills the same multicast address and port number in the OpenLogicalChannelAck. However, those terminals that wish to open a new physical multcast channel, not specified by the MC, should put zeroes in the MulticastParameters fields. The MC will fill in the new multicast address and port number in the OpenLogicalChannelAck PDU and distribute these addresses to all the other terminals in the CommunicationModeCommand.

```
OpenLogicalChannelACK                           ::=SEQUENCE
{
        logicalChannelNumber                    LogicalChannelNumber,
        h225LogicalChannelAckParameters         H225LogicalChannelAckParameters,
        ...
}

H225LogicalChannelAckParameters                 ::==SEQUENCE
{
        LANAddress              CHOICE
        {
                unicastAddress          NetworkAddress,
                multicastChannel        multicastParameters,
                ...
        }
        ...
}
```

2.2.5 Add Cause Codes to OpenLogicalChannelReject Response PDU

The H.323 MC will reject the OpenLogicalChannel if the terminal wishes to open an additional multicast channel that is not specified in the CommunicationModeCommand and the MC does not allow additional channels to be opened.

```
OpenLogicalChannelReject                  ::=SEQUENCE
{
    logicalChannelNumber                  logicalChannelNumber
    cause
    {
            unspecified                           NULL,
            dataTypeNotSupported                  NULL,
            dataTypeNotAvailable                  NULL,
            unknownDataType                       NULL,
            multicastChannelNotAllowed            NULL,               -used by MC
            insufficientBandwidth                 NULL,
            ...
    }
    ...
}
```

2.3 *General Comments*

1) In Section 1 Scope, change "H.222.0 and H.223" to "H.222.0, H.223, and H.22Z".
2) In Section 2 References, Add references to H.22Z and H.323.