Telecommunication Standardization Sector
Original: English
(TSS)

AVC 757

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

May 15-18, 1995

STUDY GROUP 15
CONTRIBUTION

Source: D. Skran, Editor, H.22Z.
email: dls@mtgbcs.att.com
voice: +1 (908) 957-5988
fax: +1 (908) 957-5627

Title: H.22Z, Media Stream Synchronization and Time Base recovery on Non-Guaranteed Quality of Service LANs

Date: April 28, 1995

Summary: This recommendation describes how audio, video, data, and control information on a non-guaranteed quality of service LAN can be managed to provide conversational services in H.323 equipment.

Notes on reading: ***{Editors notes are generally in braces}*** while underlined text is new. ~~The deleted text has strikethrough font.~~ Please ignore all references to other sections both inside and outside this document; these will be updated in the final version.

1. Scope

This recommendation describes the means by which audio, video, data, and control are synchronized and passed between H.323 terminals on a non-guaranteed quality of service LAN, or between H.323 terminals and H.320, H.324, or H.310/H.321 terminals on N-ISDN, PSTN, or B-ISDN respectively via a LAN/WAN gateway. This gateway is described in H.323. Communication via an H.323 gateway for guaranteed quality of service (QOS) LANs to an H.322 gateway and thus to H.322 endpoints is also possible.

H.22Z is intended to operate over a variety of different LANs, including IEEE 802.3, Token Ring, etc., To support this, H.22Z is described with reference to an general LAN interface (GLI). It is expected that a convergence sublayer would exist between H.22Z and the underlying real LAN. The characteristics of this sub-layer are left to the manufacturer. Thus, the scope of H.22Z communication is between H.323 terminals and H.323 gateways on the same LAN, using the same convergence sub-layer. This LAN may be a single segment or ring, or it logically could be the entire Internet. The possible means by which quality of service might be assured on this LAN, or on the Internet in general is beyond the scope of this recommendation.

H.22Z supports only single channel operation; it is assumed that the H.323 gateway may also operate as an H.244 channel aggregator. Put in H.320 terminology, H.22Z supports only, for example, the 128 Kbps transfer rate and not the 2*64 transfer rate. The primary rationale is to put complexity in the gateway rather than the terminal.

H.22Z is designed so that, with an H.323 gateway, interpretability with H.320(1990), H.320(1993), and H.320(1996) terminals is possible. However, some features of H.22Z may be directed toward allowing enhanced operations with future versions of H.320. It is also possible that the quality of service on the H.320 side may vary based on the features and capabilities of the H.323 gateway.

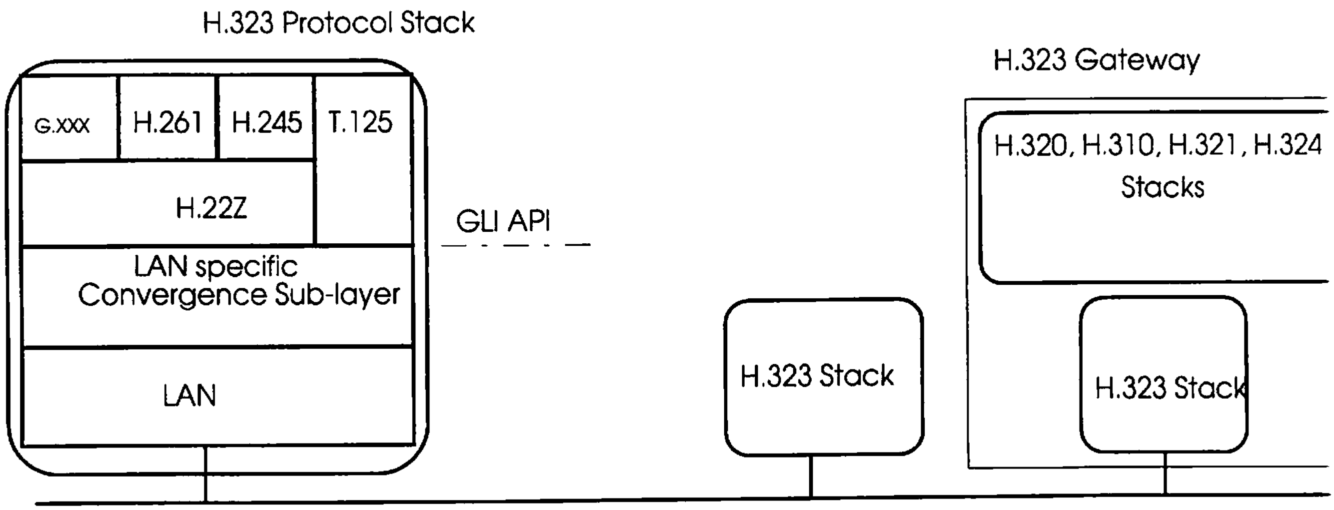

Figure 1/H.22Z
Scope of H.22Z

The general approach of H.22Z is to provide a means of synchronizing packets that makes use of the underlying LAN facilities. H.22Z is not a "multiplex" and does not attempt to pack all media and control into a single stream, which is then packetized. The framing mechanisms of

H.221 are not utilized since no need exists to coordinate more than one stream of multiplexed media information. This approach allows each media to receive different error treatment as appropriate.

Several modes of multipoint operation are supported:

- WAN Mode: In this mode, each H.323 endpoint on the LAN is connected via the H.323 gateway to H.320 endpoints or H.231 MCUs on the N-ISDN, for example. The H.323 gateway does not provide any MCU functions itself. This mode of operation is mandatory on the terminals, and requires the H.323 endpoint conform to H.230 multipoint commands such as MCC, VCF, etc.

- LAN Mode(Isolated): In this mode, the combination of the H.323 gateway, the LAN itself, and the H.323 endpoints operates like an H.231 MCU, with the LAN being used to broadcast video and audio. In this mode, the H.323 endpoint may receive and display more than one video stream, but the H.323 gateway is responsible for producing exclusive audio sums for each endpoint***(Comment: the alternative is that each endpoint does the summing; we may wish to allow for both alternatives).*** The H.323 gateway is responsible for running T.122/T.125 data conferencing**{Some have experssed a desire to replace T.122/T.125 with a distributed equivalent},** and for performing the "traffic manager" function. Note that in this mode the H.323 gateway does not need to be connected to a WAN. However, a "gatekeeper" is always required to coordinate traffic; in this limited mode it may simply be a program on an existing LAN server. This mode is optional.

- LAN Mode(Non-isolated): This mode is similar to LAN Mode(Isolated), with the addition that the H.323 gateway can operate as an H.231 MCU, and participate in cascades with other H.231 MCUs over the WAN. This mode is optional.

In both the LAN MCU modes, the H.323 gateway may optionally be a one-way device, and the H.323 terminals receive only. ***[Do we need a different recommendation for this???]***

2. References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; 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. CCITT Recommendation G.711, *Pulse Code Modulation of 3kHz Audio Channels,* November 1988

2. CCITT Recommendation G.722, *7kHz Audio Coding within a 64 kbit/s Channel*, November 1988}

3. CCITT Recommendation G.728, *Coding of Speech at 16 kbit/s Using Low-delay Code Excited Linear Prediction (LD-CELP),* May 1992

4. ITU-T Recommendation H.221(1993)[1], *Frame Structure for a 64 to 1920 kbit/s channel in audiovisual teleservices.*

[1]Previously CCITT Recommendation

5. ITU-T Recommendation H.230(1993)[2], *Frame Synchronous Control and Indication Signals for Audiovisual Systems,* December 1990

6. ITU-T Recommendation H.233(1993) *Confidentiality System for Audiovisual Services,* March 1993

7. ITU-T Recommendation H.242(1993)[3], *System for Establishing Communication Between Audiovisual Terminals Using Digital Channels up to 2 Mbit/s.*

8. ITU-T Recommendation H.243(1993), *Procedures for Establishing Communication Between Three or More Audiovisual Terminals using Channels up to 2 Mbit/sec.*

9. ITY-T Recommendation H.320(1993)[4], *Narrowband Visual Telephone Systems and Terminal Equipment.*

10. ITU-T Recommendation T.122(1993), *Multipoint Communication Service for Audiographics and Audiovisual Conferencing Service Definition*

11. ITU-T Recommendation T.123(1993), *Protocol Stacks for Audiovisual and Audiographic Teleconference Applications.*

12. ITU-T Recommendation T.125(1994), *Multipoint Communication Service Protocol Specification.*

13. H.324.

14. H.321

15. H.322

16. H.310

17. More needed

3. Definitions

Multipoint Control Unit (MCU): a multi-port device, by means of which two or more audiovisual terminals may intercommunicate in a conference call; See recommendations H.231 and H.243 for details. The MCU is used to connect terminals over a wide area, and especially to connect two or more terminals entering the WAN via an H.323 gateway into a WAN conference.

4. Conventions

In this document, "shall" refers to a mandatory requirement, while "should" refers to a recommended but optional feature or procedure.

---

[2]Previously CCITT Recommendation
[3]Previously CCITT Recommendation
[4]Previously CCITT Recommendation

5. Abbreviations

BAS	Bit rate Allocation Signal
CIF	Common Intermediate Format
ECS	Encryption Control Signal
FAS	Frame Alignment Signal
FAW	Frame Alignment Word
GLI	General Lan Interface
GOB	Group of Blocks
H-MLP	High speed Multi-Layer Protocol
HSD	High Speed Data
LAN	Local Area Network
LD-CELP	Low Delay - Code Excited Linear Prediction
LSB	Least Significant Bit
LSD	Low Speed Data
MBE	Multi-Byte Extension
MCC	Multipoint Command Conference
MCN	Multipoint Command Negating
MCS	Multipoint Command Data Symmetrical Data Transmission
MF	MultiFrame
MLP	Multi-Layer Protocol
MPI	Minimum Picture Interval
MSB	Most Significant Bit
NS	Non-Standard
PCM	Pulse Code Modulation
QCIF	Quarter Common Intermediate Format
QOS	Quality of Service
SBE	Single Byte Extension
SC	Service Channel
SCM	Selected Communications Mode
SMF	Sub-MultiFrame
SP	Still Picture
VCF	Video Command "Freeze Picture Request"
VCU	Video Command "Fast Update Request"

6. General Lan Interface(GLI)

The general LAN interface(GLI) allows H.22Z to operate without reference to the specific features and characteristics of particular LANs. The GLI has the following features:

- A reliable, acknowledged means of sending data that may incur delay in the presence of errors. This means is assumed to supply data both reliably and in the sequence sent.

- An unreliable, unacknowledged (datagram) means of sending data which does not incur additional delay in the presence of errors. ***Question: Do we assume the presence of a CRC or checksum, or is this something we must add in H.22Z?***

- A maximum and a minimum packet size, which may be different for the reliable and the datagram modes of operation.

- A means of addressing endpoints on the LAN, of an unspecified format and content. This address will be referred to as the "LAN Address" of the H.323 terminal. H.323/H.22Z does not process this address; the only requirement is that the address be unique in the space served by the H.323 gateway.

6.1. GLI API

{Editorial Note: It may be more appropriate to specify this interface in terms of PDU contents; I have take the API approach as one method of clarifying the interface}

{Note e also that since the destination LAN address appears in the GLI API, in effect this is enveloped around the common PDU structure. However, it seems to make sense to bring this outside the common PDU since it may vary greatly from LAN to LAN.}}

6.1.1.Reliable Send

The call is

rsend(packet_size, location_of_data, location_of_error_value,destination LAN address)

The error values are

1	lan cannot accept a packet this large
2	lan cannot accept a packet this small
3	lan is congested; cannot accept more data(busy)
4	LAN address is incorrect

6.1.2. Datagram Send

The call is

ursend(packet_size, location_of_data, location_of_error_value,destination LAN address)

The error values are

1	lan cannot accept a packet this large
2	lan cannot accept a packet this small
3	lan is congested; cannot accept more data(busy)
4	LAN address is incorrect

6.2. GLI QOS Metrics

The following metrics are used to characterize the general LAN:

Maximum Bit Rate(MBR): The bit rate at which packets are sent on the LAN. For example, 10 Mbps for IEEE 802.3 is a common rate.

Instantaneous Packet Delay(IPD): The delay on a packet due to any arbitration or collisions that may take place. This delay varies from moment to moment, and often increases with load on the LAN.

Packet jitter(PJ): The variation in IPD from packet to packet that must be compensated for by buffering on the receive side. Generally, the higher the jitter, the more delay across the LAN.

Media Bandwidth: On the LAN, the MBW is a product of the Packet size time the Packets/sec. For example, 64 kbit/sec PCM might be sent as 80 packets/sec of 100 bytes each.

Minimum Media Delay: On the LAN, the minimum delay that will be experienced is that needed to fill a packet plus the time needed for the packet to transit the LAN. Thus, to fill a 100 octet packet with PCM requires 12.5 msec. On a 10 Mbps LAN, sending the 800 bit packet requires about 80 usec. Whether we must double the packing delay to get overall delay depends on whether a checksum is computed on each packet

Actual Packet Delay: This is the sum of MMD(Minimum Media Delay) plus IPD(Instantaneous Packet Delay) plus Packet jitter (PJ). Note that jitter may raise or lower the total delay.

7. Synchronization Mechanism

7.1.General Approach

The general approach is to send data(T.120) and control(H.245) using an underlying reliable means of transport(**rsend**). The impact of varying delay on these types of data is not as strong as the impact on audio and data. While data and audio are in use, the maximum packet size allowed for data and control is limited the packet size used for audio and video. If audio and video are turned off, as during a file transfer, large packet sizes may be usd to rapidly transfer control or data.

Audio and video are sent via an unreliable means**(ursend)** to minimize delay. If the minimum packet size supported by the underlying LAN is insufficient to suitably minimize delay, audio and video may be placed in the same packet. Either forward error correction or error concealment must be applied to overcome lost packets/bit errors; in general audio/video packets are not re-transmitted since this would result in excessive delay in the LAN environment.

Quality of service is dynamically measured by H.22Z using timestamps on packets and feedback from the receiver in the control channel. This feedback enables the sender to compensate for increased LAN congesion in the following ways:

- Reducing the frame rate for a short period of time. This may result in the H.323 gateway sending additional H.261 fill frames in the away from the LAN direction to compensate for the reduced frame rate. This should be the method initially attempted.

- Reducing media bit rate(e.g. switching from 384 kbit/sec to 256 kbit/sec). This may involve a simple instruction to the encoder in a terminal, or it may involve the use of a rate reducer function in the H.323 gateway.

- Turning off media of lessor importance (e.g. turning off video to allow a large amount of T.120 traffic).

- Returning a busy signal (adaptive busy) to the receiver as an indication of LAN congestion. This may be combined with turning off a media, or even all media other than the control channel.

The same time stamps used to measure QOS are also used to synchronize audio and video in the receiver. Audio and video packets contain signals such as freeze picture release that must be closely coordinated with the media itself. In some cases signals carried by the BAS channel in H.221 are carried with the media stream in H.22Z. No effort is made to tightly coordinated the control channel (H.245) or T.120 with the audio/video media streams.

{Editorial Note: A question has been raised concerning whether T.120 should fit within the H.22Z PDU structure, or whether it should run directly on resend. The suggestion is that the H.22Z PDU structure adds no value for T.120. One possible answer is that the program id is useful to associated T.120 streams with audio/video streams. This requires consideration}

7.2. PDU Structure

The general PDU structure is as follows:

Field		Octet Number
Media Type(7 bits)	E	1
Sequence Number(high)		2
Sequence Number(low)		3
Timestamp (high bits)		4
Timestamp		5
Timestamp		6
Timestamp		7
Program Id		8

Type: This 7 bit field has the values:

0	Reserved.
1	G.711
2	G.728
3	G.722
4	G.723
5	Reserved for the new Low-Complexity Coder
6-10	Reserved
11	H.261
12	H.263
13	Mixed Audio/Video
14-20	Reserved
21	**T.120 (no T.123)[needed???]**
22	**Transparent Data[needed???]**
23	H.245 control
24-127	Reserved

E bit: When set to 1, indicates additional octets after the normal header.

Sequence Number: This 16 bit sequence number counts H.22Z PDUs. The sequence number increments by 1 for each PDU sent. The sequence number is used by the receiver to detect PDU loss and to restore sequence. The initial value of the sequence number is random to make know-plaintext attacks on encryption more difficult. **{Editor's note: Essentially taken from RTP} {Do we need a CRC on the header???}{Should the CRC only be present in ursend mode?**

Timestamp: This 32 bit value is the clock time at which the PDU was presented to the H.22Z GLI interface. Each PDU will have a different timestamp **{Editors note: This DOES NOT follow RTP}**. The absolute accuracy is not as vital as the relative accuracy of this timestamp. This timestamp will be used by the receiver to measure jitter and LAN delay. **{Editor's note: Do we need an RTP-like stamp as well that is related to the time at which the media packets were generated, and which may be the same for packets generated logically at the same time?}**

Program Id: Used to associated audio/video packets with common source at a single LAN address **{Editors Note: Do we need this??}**

7.2.1. Audio PDUs

7.2.1.1. G.711 PDUs

{Editors Note: A major issue is whether the H.323 terminal is expected to ever talk directly to a PSTN terminal without an echo cancellor. If this is the case, a very small packet size is needed, or the gateway must have echo cancellers. This section has been written assuming the more loose constraint of taking to an H.320 terminal that does have an echo cancellor, and can handle 100s of msecs of delay. This issue should be taken up in Stockholm}

Since G.711 has no frame structure (it is just a series of octets, or sub-octets), the major issue is how many octets per PDU. For the H.320 case we could target a LAN delay of 100 msec (very annoying without echo cancellors). Since 100 octets are generated every 12.5 msec. it appears the PDU could contain several 100 octets.

If the underlying GLI indicates a bad packet, the receiver is expected to either fill with silence or replicate the last octet(this usually gives a good result) This is also done if the sequence number indicates a lost PDU. **{Editors Note: Need for CRC or checksum should be considered}**

{Editors Note: We need to consider packings for 48/56 kbit/sec PCM}

7.2.1.2. G.728 PDUs

For further study. Note that G.728 has an internal structure we probably shouldn't break across PDUs.

7.2.1.3. G.722 PDUs

For further study.

7.2.2. Video PDUs

{Editors note: This is borrowed with some changes from the IETF RTP document suite}

The purpose of this section is to specify how H.261 video streams can be carried on H.22Z.

7.2.2.1.Structure of the packet stream

H.261 codecs produce a bit stream. In fact, H.261 and companion recommendations specify several levels of encoding:

1. Images are first separated in blocks of 8x8 pixels. Blocks which have moved are encoded by computing the discrete cosine transform (DCT) of their coefficients, which are then quantatized and Huffman encoded.
2. The bits resulting of the Huffman encoding are then arranged in 512 bits frames, containing 2 bits of synchronization, 492 bits of data and 18 bits of error correcting code.
3. The 512 bits frames are then interlaced with an audio stream and transmitted over px64 kbps circuits according to specification H.221.

When transmitting over the Internet, we will directly consider the output of the Huffman encoding. We will not carry the 512 bits frames, as protection against errors can be obtained by other means. **Similarly, we will not attempt to multiplex audio and video signals in the same packets, as UDP and RTP provide a much more efficient way to achieve multiplexing. {Editors Note: We may wish to deviate from the IETF here}.**

Directly transmitting the result of the Huffman encoding over an unreliable stream of UDP datagrams would however have very poor error resistance characteristics. The H.261 coding is in fact organized as a sequence of images, or frames, which are themselves organized as a set of Groups of Blocks (GOB). Each GOB holds a set of 3 lines of 11 macro blocks (MB). Each MB carries information on a group of 16x16 pixels: luminance information is specified for 4 blocks of 8x8 pixels, while chrominance information is only given by two color difference components 8x8 "red" and "blue" blocks. These components and the codes representing their sampled values are as defined in the CCIR Recommendation 601.

This grouping is used to specify information at each level of the hierarchy:

- At the frame level, one specifies information such as the delay from the previous frame, the image format, and various indicators.
- At the GOB level, one specifies the GOB number and the default quantifier that will be used for the MBs.
- At the MB level, one specifies which blocks are present and which did not change, and optionally a quantifier, as well as precisions on the codings such as distance vectors.

The result of this structure is that one needs to receive the information present in the frame header to decode the GOBs, as well as the information present in the GOB header to decode the MBs. Without precautions, this would mean that one has to receive all the packets that carry an image in order to properly decode its components. In fact, the experience has shown that:

1. It would be unrealistic to carry an image in a single packet: video images can sometimes be very large.
2. GOB information typically fits in a packet. In fact, several GOBs can often be grouped in a packet.

Once we have take the decision to correlate GOB synchronization and packetization, a number of decisions remain to be taken, due to the following conditions:

1. The algorithm should be easy to implement when packetizing the output stream of a hardware codec.

2. The algorithm should not induce rendition delays -- we should not have to wait for a following packet to display an image.

3. The algorithm should allow for efficient resynchronization in case of packet losses.

4. It should be easy to depacketize the data stream and direct it to an hardware codec's input.

5. When the hardware decoder operates at a fixed bit rate, one should be able to maintain synchronization, e.g. by adding padding bits when the packet arrival rate is slower than the bit rate.

6. The fragmentation process should break the frame up so that packets start on GOB boundaries when possible.

The H.261 Huffman encoding includes a special "GOB start" pattern, composed of 15 zeroes followed by a single 1, that cannot be imitated by any other code words. That pattern marks the separation between two GOBs, and is in fact used as an indicator that the current GOB is terminated. The encoding also includes a stuffing pattern, composed of seven zeroes followed by four ones; that stuffing pattern can only be entered between the encoding of MBs, or just before the GOB separator.

The first conclusion of the analysis is that the packets should contain all the GOB data, including the "GOB start" pattern that separate the current block from its follower. Actually, as this pattern is well known, we could as well use a single bit in the data header to indicate that a GOB-start pattern must be added at the decoder side.

Not encoding the GOB-start pattern has two advantages:

1. It reduces the number of bits in the packets, and avoids the possibility of splitting packets in the middle of a GOB separator.

2. It authorizes gateways to hardware decoders to insert the stuffing pattern in front of the GOB, in order to meet the fixed bit rate requirement.

Another problem posed by the specificities of the H.261 compression is that the GOB data have no particular reason to fit in an integer number of octets.

7.2.2.2. Specification of the packetization scheme

The data header will thus contain two three-bits integers, EBIT and SBIT:

- SBIT indicates the number of bits that should be ignored in the first (start) data octet.

- EBIT indicates the number of bits that should be ignored in the last (end) data octet.

Although only the EBIT counter would really be needed for software coders, the SBIT counter was inserted to ease the packetization of hardware coders output.

At the receiving sites, the GOB synchronization can be used in conjunction with the synchronization service of the RTP protocol. In case of losses, the decoders could become desynchronized. The "S" bit of the H.261 option field will be set to indicate that the packet includes the beginning of the encoding of a GOB. A GOB start code must be prepended to that packet before decoding. If several GOBs are encoded inside a same packet, all the GOBs except the first one contain the GOB start code. The receiver will detect losses by looking at the RTP sequence numbers. The receiver is recommended to resequence out of order packets in order to limit the packet loss effect. Some misordering of packets in the network seems likely, even when there is no loss, and one would not want to drop a frame because of that. In case of losses, it will ignore all packets whose "S" bit is null. Once an S bit packet has been received, it will prepend the GOB start code to that packet, and resume decoding. The "E" bit of the H.261 header will be set to indicate that the packet contains the end of a GOB, so after resequencing, it can safely be decoded without having decoder state hanging between packets. The "fragment offset" field (2 bytes) is set to the byte offset of the current packet into the frame. It is used to estimate how much memory shall be left for the possible delayed packets. A 2- bytes length for the "fragment offset" field is enough since H.261 recommendation specifies a maximal size of 256 kbits for CIF and 64 kbits for QCIF images.

7.2.2.3. H.22Z PDU Structure

The H.22Z timestamp encodes the date at which the PDU was passed to the GLI. The timestamp consists of the middle 32 bits of a 64-bit NTP timestamp, as defined in RFC 1305 [2]. That is, it counts time since 0 hours UTC, January 1, 1900, with a resolution of 65536 ticks per second. (UTC is Coordinated Universal Time, approximately equal to the historical Greenwich Mean Time.)

The very definition of this settings implies that the beginning of an image shall always be synchronized with a packet. The H.22Z sequence number can be used to detect missing packets. In this case, one shall ignore all incoming packets until the next synchronization mark is received ("S" bit set in the H.261 header). The marker "M" bit of the H.22Z header can be used as a flag to trigger display the new image on the screen. This marker has a value of one in the last packet of a video frame. The H.261 data will follow the H.22Z PDU header when the type field is set to 2 with two additional octets.

The H.261 options field(some number of extra octets in the PDU header) is defined as following:

S (1 bit)	Start of GOB. Set if the packet is a start of GOB
SBIT (3 bits)	Start bit position number of bits that should be ignored in the first data octet. SBIT must be null if S is unset.
E (1 bit)	End of GOB. Set if the packet is an end of GOB
EBIT (3 bits)	End bit position number of bits that should be ignored in the last data octet. EBIT must be null if E is unset
I (1 bit)	Full Intra Image flag. Set if it is the first packet of a full intra image.
V (1 bit)	Movement Vector flag. Set if movement vectors are encoded. All V bits of the same frame must be identical.
MBZ (3 bits)	Must Be Zero.
SIZE (3 bits)	Image format: QCIF, CIF or number of CIF in SCIF.
fragment offset (16 bits)	Byte offset of the current packet into the frame

The image format (3 bits) is defined as following**(table deleted since we will rework):**

{Editors Note: IETF uses a super-CIF format as well as CIF & QCIF We will probably want to rework these bits}

7.2.2.4. Usage of H.245 control packets as they apply to Video PDUs

When sending or receiving H.261 streams through the H.22Z protocol, the endpoints should be ready to:

(1) process or ignore all generic H.245 control packets.

(2) send or receive H.261 specific H.245 control packets, to request a video refreshment. (specifically, the fast update request of H.320).

{Editors note: where should we put H.245 related procedures? Perhaps this entire section should be moved}

7.2.2.4.1. Controlling the reverse flow

Support of the reverse H.245 control packets by the H.261 sender is optional; in particular, early experiments have shown that the usage of this feature could have very negative effects when the number of sites is very large. Thus, reverse H.245 control packets should be used with caution. The aim of these packets is to speed the refreshment of the video when it is possible. Videoconferencing applications do not require reliable multicast packet delivery such as whiteboard applications. Reliable multicast protocols can use similar NACK H.245 control packets but in this case, the main purpose is to provide reliable data transfer to the receivers packets with minimal throughput. A few reliable multicast protocols use random delays to prevent NACK implosion problem [3]. On the other hand, it is more efficient in videoconferencing applications to send NACK control packets as soon as possible, i.e. as soon as a loss is detected, without adding any random delays. In this case, multicasting NACKs control packets generates useless traffic between receivers since only the coder will use them. But this method is only efficient when the number of receivers is small. e.g. in IVS [5] reverse H.245 control packets are used only if there are less than 10 participants in the conference.

A site may distinguish reverse H.245 packets from forward H.245 packets by their arrival port. Reverse H.245 packets arrive on the same port that the site uses as a source port for forward (data) H.22Z packets. **{Editors note: Most of this probably does not apply}**

7.2.2.4.2. Negative ACKnowledgements (NACK) H.245 packet

Packets lost are detected using the RTP sequence number. After a packet loss, the receiver will resynchronize on the next GOB. However, as H.261 uses differential encoding, parts of the images may remain blurred as long as all corresponding MBsare not encoded in INTRA mode; i.e. absolute encoding withoutrelation to previous frame. There are several ways to put it right.

The fastest way is to request a refreshment. As all GOB belonging to a given video image carry the same time stamp,the receiver can determine a list of GOBs which were really received for that time stamp, and thus identify the "missing blocks". Requesting a specific reinitialization of these missing blocks is more efficient than requesting a complete reinitialization of the image through the Fast Update. When it is impossible to use NACK, e.g. if the number of receivers is large or if the coder does not handle NACK,another method consists to periodically force INTRA encoding each MB. The INTRA refreshment rate can be raised in order to speed the recovering when the loss rate measured is important.

{Editors Note: We need to consider whether we need this NACK in H.245}

7.2.3. Data PDUs

The general approach is that T.122/T.125 PDUs should be mapped directly into GLI PDUs using the **rsend** interface. This seems needed to generalize away from any particular LAN stack. The major constraint H.22Z provides is that while audio and/or video is in use, the PDU size used for data shall not exceed that used for audio/video. When both audio and video have been turned off, any PDU size can be used. **{Editors note: Do we need to make turning video off alone a special case with some other rule?**

{Editors Note: The issue of whether T.120 has a media header, or directly uses the GLI has been raised. Direct use of the GLI seems appropriate, but the implications for control and signaling require consideration}

{Editors Note: The issue of whether T.122/T.125 should be used, or whether a new layer that provides the same services in a distributed fashion should be used has been raiased. Until specific proposals are bought forward, T.122/T.125 will be the default approach. Note that this only applies to the "isolated LAN" "MCU" operation; in the "direct gateway" approach, the H.323 terminals on the LAN must use T.122/T.125 in order to participate in the WAN conference.

7.2.4. Control PDUs

H.245 PDUs are simply mapped directly onto the rasend interface. **An issue is whether any are going to be longer than the Audio/Video PDUs.**

{Editors Note: It is assumed that a point-to-point H.245 link exists between each H.323 endpoint on the LAN, and the H.323 gateway, even if there is no WAN connection. The broadcast case requires further consideration.}

7.2.5. Mixing Audio and Video in a PDU

It may be desirable to mix Audio/Video in one PDU to lower audio delay. +

8. Mechanisms for maintaining QOS

8.1. Measuring QOS

This will cover how the various metrics are computed from the timestamps.

8.2. Maintaining QOS

This will cover details for the procedure for dealing with LAN congestion.