Media Coding and Distribution over Unreliable Networks : Some Issues.

Deepak Jaiswal
djai@sasken.com

Sasken Communication Technologies Limited,
Bangalore, India.

Abstract

The field of distributed multimedia computing and systems has shown extraordinary growth as witnessed by new technologies and trends. With ever increasing demand for sharing multimedia information across the Internet and in distributed environments, there is a large demand to come up with cost-effective and graceful solutions on Desktops and PCs using the current installed base of networks.

The present paper is meant as an overview of the various issues involved in media coding and its playback at remote receivers after being transmitted on error prone and unreliable networks. Limitation of the current networks in providing the performance required by real-time multimedia streams and the issues which need to be addressed to provide acceptable multimedia playback presentation at remote receivers is also presented. Issues such as synchronization, error resilient mechanisms and end-to-end delay are discussed in detail. The present paper also considers Quality-of-Service (QoS) and Quality-of-Presentation (QoP) requirements.

Introduction

Advances made in audio-video compression techniques and high bandwidth networks are enabling distributed multimedia applications. But, what make the development of such an application a challenge is the real-time requirements of the application, the lack of guaranteed quality of service from the network and the errors in the transmission of coded media streams. Real-time services from current computer systems and guaranteed quality of service from networks are expensive in today's world. To provide an affordable distributed multimedia system, one has to develop systems, which provide acceptable performance on non real-time systems and over non-guaranteed networks such as the Internet.

Rest of the paper is organized as follows. Section 1 looks at the need for media encoding and related issues. Section 2 deals with synchronization of media streams over unreliable networks. Error recovery mechanisms are investigated in Section 3 while end-to-end delay issues are discussed in Section 4. Section 5 discusses the QoS (Quality of Service) and QoP (Quality of Presentation) parameters and the codec parameters that need be controlled for dynamic adaptation. An overview of the current video conferencing standard (for Internet) and its components are briefly discussed in Section 6. The last section concludes the paper.

Keywords: Synchronization, Error resilient mechanisms, End-to-end delay, Quality-of-Service and Quality-of-Presentation.

1 Media Distribution

The need for media distribution arises by the fact that the source generating the media streams and the sink playing back the media streams are typically not on the same machine. They are distributed and thus the need to compress the data generated by media devices to save transmission time and space (disk space, bandwidth). There are essentially three ways in which data distribution takes place.

1. **Off-line data distribution:** The compressed data to be played-back to the end user is available on the host machine before the multimedia presentation starts. Hence, live transmission of data is not required. In this kind of presentation, the key issue is real-time decoding and presentation. Due to the non-real time performance of operating systems, synchronization of temporally correlated media streams poses a challenge. Timely scheduling of all the activities is required. In case of an overload, low priority tasks need to be dropped such that there is a graceful degradation in the presentation. Two formats are possible for local data.
   - **Separate streams:** In this format, each multimedia stream that needs to be presented is stored in a different logical device e.g., files. The temporal and spatial relationships between the streams, if any, are defined separately in a format called a scenario description.
   - **Multiplexed stream:** In this format, various multimedia streams are multiplexed together into a single data stream. The temporal synchronization of the media streams is either implicit by the nature of multiplexing or each data chunk may have time-stamps which state inter and intra stream synchronization requirements.

2. **One way data transmission:** There is a growing need to access multimedia information that is distributed across various data repositories. If a network connects the source (data repository) and the presentation sink, then the data can be fetched live from the source and presented to the end user. This has the following advantages:
   - All the information available in various servers is at your “finger-tips”.
   - Large storage space is not required at the presentation end.
   - Small start-up time. (Start-up time will be large if media streams are down loaded before being played).

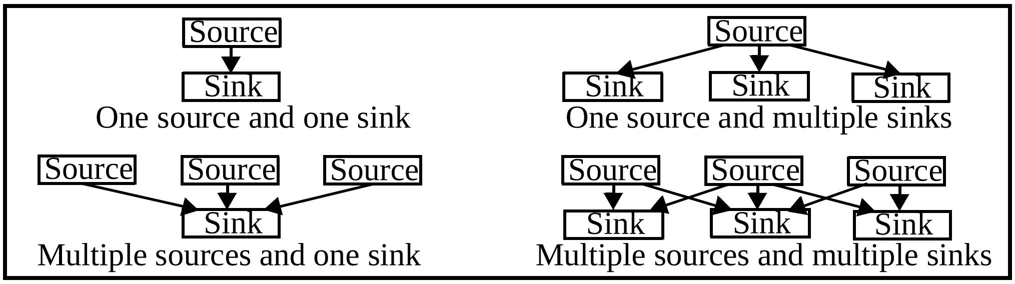

Fig. 1: One way data transmission models

There are number of variations to one-way transmission as shown in Fig. 1. Distributed multimedia applications based on one-way transmission are listed in Table 1.

Table 1: Examples of one-way data transmission.

One-way transmission model	Examples
One source and one sink	Tele-presence and Tele monitoring.
One source and multiple sinks	Tele learning, Video on Demand, DTV and HDTV.
Multiple sources and one sink	Media processors e.g., mixers.
Multiple sources and multiple sinks	Future multimedia terminals.

3. **Two way data transmission:** With the growing need for group activity among geographically distributed people, there is an increasing demand for distributed multimedia applications which facilitate group activity. In this mode of communication, data is transmitted in both directions. There are two scenarios in this type of data communication as show in Fig. 2.

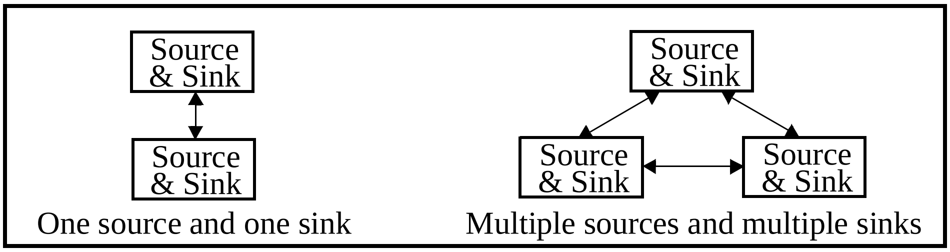

Fig. 2: Two way data transmission models.

Distributed multimedia applications (listed in Table 1 and Table 2) like video-phone and video-conferencing present the most challenging issues to the developers. The challenging issues in this area are divided into two sections. One of which is related to networks (not discussed in this paper) and the other which is related to codecs. Network related issues are:

- Signalling between the various interacting hosts.
- Guaranteed performance by the network based upon requested Quality-of-Service
- Network optimization: Multicasting and reservation issues.

Table 2: Examples of two-way data transmission.

Two-way transmission model	Examples
One source and one sink	Video-phone.
Multiple sources and multiple sinks	Video conferencing and collaborative work applications, e.g., distributed CAD tool.

The codec related issues are:

- **Synchronization of media streams:** The temporal relationship between packets of a media stream is destroyed due to unpredictable delays in the transmitting channel. This leads to annoying (unsynchronized) playback of streams.
- **Error recovery:** A media stream transmitted over an error-prone and unreliable channel may experience packet loss and data corruption. Data loss produces annoying interruptions in the normal presentation of the streams.
- **Reduction in latency:** Maintaining low latency in a full duplex conferencing application is far more critical than in a one-way multicast for video-on-demand application. This is due to the fact that human conversation becomes a challenge when accumulated end-to-end delay exceeds 500ms. For more natural communication to take place, latency should be kept below 250ms.
- **Dynamic adaptation of codec functionality:** During a multimedia session, the Quality-of-Service (QoS) provided by the network or the Quality-of-Presentation (QoP) requested by the user may change. To be able to deliver the best performance from our codecs under a changing environment, we need to change the control parameters of the codecs.

2 Synchronization

Multimedia synchronization [1, 2] is the task of coordinating various time-dependent media streams, and which arises when a variety of distributed media streams with different temporal characteristics are brought together and integrated into a multimedia system such as video conferencing, video on demand, remote learning and collaborative work systems. When media streams are presented for playback in real-time from different sources, either located locally or distributed geographically, the task of maintaining the temporal relationships (as shown in Fig. 3) among these media streams can be difficult but is essential for smooth presentation.

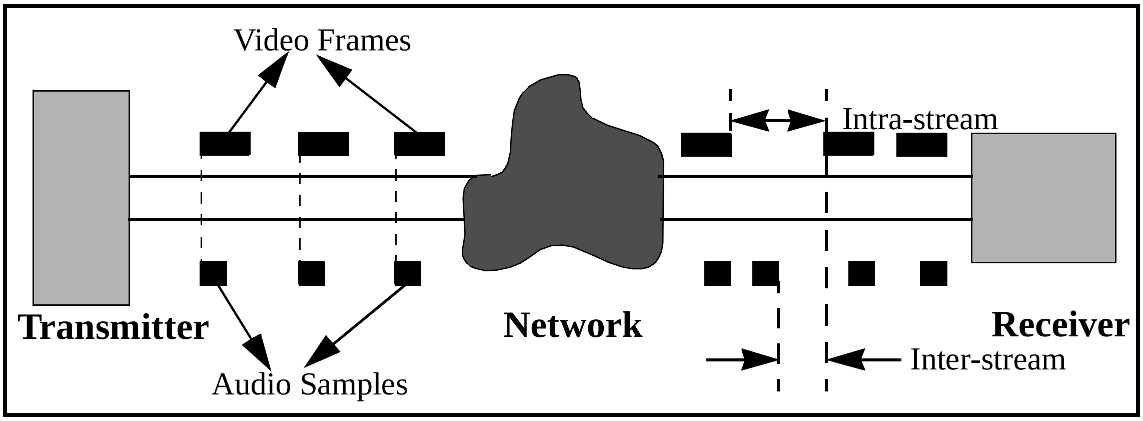

Fig. 3: Inter and Intra Stream Synchronization.

In general, there are three kinds of temporal synchronization issues within a distributed multimedia system framework [3]. They are:

- **Intra-stream:** This stream synchronization refers to the internal stream synchronization between the transmitter and the receiver. This is also known as **serial synchronization,** which defines the timing relationship of a single media over a single connection. In this synchronization, the real trade-off lies between end-to-end delay and glitches (discontinuity in playback).
- **Inter-stream:** Continuous control is required to maintain synchronization across multiple continuous media streams. Also known as **parallel synchronization**, this defines timing relationships for multiple connections or for multiple media interleaved in a single connection. The real issues in this synchronization are:
  - Lip synchronization. (Minimum asynchrony between temporally related streams)
  - Synchronization with minimum glitches in the master stream. The master stream is the stream to which all the other streams are synchronized.
- **Event-based:** Event based synchronization is required for the control of temporal multimedia events, e.g., the coordination of mixed media presentation and user interaction.

2.1 Causes of Asynchrony

The causes of asynchrony (i.e., the failure to keep up the required level of synchronization) are discussed in this section.

2.1.1 Delay jitter

Delay jitter is the variation in end-to-end delay. The three factors contribute to delay jitter [3] are:

1. **Collection delay:** This is the time needed for the transmitter to collect media units and prepare them for transmission.
2. **Transmission and queuing delay:** This is the network delay from the network boundary at the transmitter to the network boundary at the receiver.
3. **Delivery delay:** This is the time the receiver needs to process the media units and prepare them for playback.

Note that none of the delay components are necessarily constant. For example, the network delay in ATM networks varies because different cells may experience different queuing delays due to the unpredictable burst in the network. The collection delay and the delivery delay may also vary from packet to packet due to different processing time (encoding/decoding, segmentation/re-assembly) required and varying load on the host machines.

2.1.2 Asynchrony of the clocks

Asynchrony between the clocks of the transmitter and the receiver clock arises because of drifts between the clocks and the lack of notion of a common global time. Synchronization of media streams becomes a difficult task in the absence of synchronized clocks as each action for media stream synchronization is based on a temporal event e.g., time-outs. One of the efforts to synchronize the transmitter's and the receiver's clocks is to replicate the transmitter's clock at the receiver. In this technique, the clock of the transmitter is calculated from the arrival time of a data packet, its generation time (which is typically transmitted as a time-stamp in the packet) and an estimation of the end-to-end delay experienced by that packet. If there is an error in the estimation of the delay then the clocks will not be synchronized and this will jeopardize the synchronization process. For example, if the delay estimation of the packet was more than actual (i.e., the packet experienced minimum delay and it was assumed that it took average end-to-end delay). In such a case, the playback of media streams (based on the transmitter clock derived from incorrect estimation of end-to-end delay) may experience discontinuity due to data starvation. Hence, the synchronization process should be **adaptive** i.e., self-correcting.

2.1.3 Different Initial Collection Times

When there is more than one transmitter in a group communication, then the transmitters must collect and transmit synchronously, otherwise temporal relationships among media units might be destroyed. For example, consider two media transmitters, one providing voice and the other video. If they start collecting and transmitting their media units at different times, playback of media units of voice and video from two sources at the destination loses semantic meaning. Media units with no temporal correlation are played back simultaneously resulting in ***lip-sync*** failure.

2.1.4 Different Initial Playback Times

The receivers in a group communication must start playing temporally related media units simultaneously, so that each user perceives media units synchronously. If the initial playback times are different for each user, then asynchrony will arise. For some multicast applications in which fairness is the major concern, the playback times of media units of all receivers should be the same, otherwise, the earlier a receiver gets media units, the earlier he can react.

The above causes of asynchrony can be removed if the following requirements are fulfilled:

- All the interacting hosts have the notion of a global synchronized clock.
- Each data packet is transmitted with generation time-stamp.
- All receivers should have sufficient memory to buffer playback media streams.

2.1.5 Transmitter and receiver load variance

Load variance at the transmitter and the receiver also contributes to asynchrony. If the encoding and decoding processes are not scheduled at proper times, then the real-time constraints of these processes are missed, which produces discontinuity in the media streams. One way to avoid this source of asynchrony is to have dedicated real-time systems at the transmitting and receiving ends.

2.1.6 External event interaction

In interactive multimedia systems, playback of media streams needs to be synchronized with external events. For example, user interactions to change the language of the audio/speech stream or to change the angle of the video stream. To fulfill the requirements of new presentation scenarios, a new set of media streams need to be initiated while another set of media streams need to be stopped. This change in presentation requirement may lead to discontinuity in presentation. The challenge lies in responding to the external event as soon as possible without any discontinuity.

2.2 Asynchrony measurement

Asynchrony in a multimedia presentation is measured in terms of skew and end-to-end delay. Skew is defined as the difference in scheduled playback time and the actual playback time of a media stream as shown in Fig. 4.

The asynchrony measurement as reported in [4] was measured using Mean Observation Score by a varying set of people and using a number of test streams

- **In-sync:** Temporally related audio and video streams are said to be in sync if the skew between the two streams is between -80 ms (audio behind video) through +80 ms (audio ahead of video).
- **Out-of-sync:** Temporally related audio and video streams are said to be out-of-sync if the skew between the two streams is outside the range of -160 ms to +160 ms.

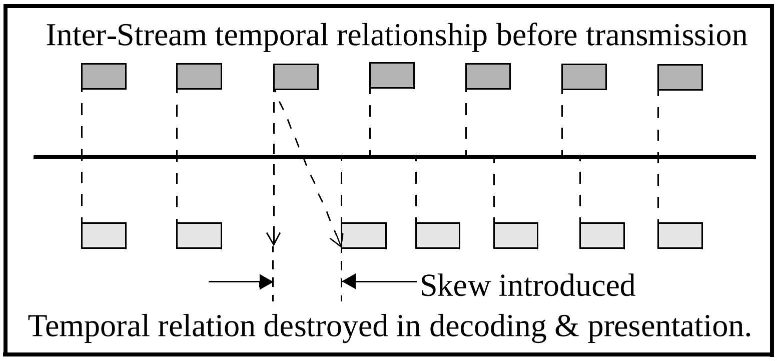
Inter-Stream temporal relationship before transmission
Skew introduced
Temporal relation destroyed in decoding & presentation.

Fig. 4: Skew introduced in network transmission.

2.3 Performance parameters

The parameters that define the desired quality of synchronization in a multimedia presentation has been defined by Bansal and Ravindra [5], and they are:

- **Divergence vector (DV):** This specifies how much of asynchrony can be present between related media streams in a multimedia presentation. This is specified by a divergence vector (DV) $\{l_1,l_2,...,l_n\}$ where (l_j) is the maximum temporal divergence allowed on the j^{th} stream. In other words, the range of acceptable frames at the i^{th} interval for j^{th} stream are $f_j(i)$ - $f_j(i-l_j)$. For an application at i^{th} interval, this indicates how far apart the data frames can be along the temporal axis and still be acceptable to the end user.
- **Inter-glitch spacing (IGS):** A minimum acceptable spacing between two consecutive glitches is specified for each media stream. The **IGS** is related to the human perception level of the glitch and its persistence level in the application. It is a measure of the presentation quality of individual streams.
- **Inter-frame pause (IFP):** It gives the minimum and maximum play rate of a media stream that is acceptable to the end user. It specifies how much a stream can tolerate a change in the pause between the play of $\mathbf{i^{th}}$ and $\mathbf{(i+1)^{th}}$ frames.

2.4 Synchronization actions

2.4.1 Synchronization actions at the receiving/decoding end

A main job of the synchronization process takes place at the receiving end. The various techniques are:

- **Buffer occupancy:** In this technique, the play-rate of the media stream is tuned by the rate of change of buffer occupancy. The playback buffer has two or three watermarks, dividing the play-out buffer into a number of zones. The key principle of tuning is to decrease the play-rate if the present buffer occupancy reduces below the low-water mark and to increase the play-rate if the present buffer occupancy rises above the high-water mark.

- **Arrival time and delay variance:** By studying the arrival time and the delay variance of the incoming data packets, one can estimate the network sub-system condition. Once the state of the network is known, appropriate synchronization actions can be taken. For example, if it is known that the network is congested, then the receiver may request the transmitter to use a lower bit-rate encoding mechanism.

2.4.2 Synchronization actions at the transmitting/encoding End

The synchronization actions that are taken at the transmitter are based upon feedback information received from the network sub-systems, the decoding ends (receivers) or the intermediate routers, mixers or translators. Based upon the feedback, the transmitter may change the control parameters of the codecs. For example, in the event of high packet loss, the transmitter may change the codec control parameter to reduce the output bit-rate. In an RTP[1] [6] framework, the RTCP[2] [7] control message **Receiver Report** gives congestion indication to the transmitter whereas the **Sender Report** (transmitted by the encoding end) gives statistics of transmitted packets to the receivers.

The transmitting/encoding end takes the decision for encoding parameters based upon the results of the following events:

- Collection time failures
- Encoding deadlines missed
- Congestion report from the network
- Packet loss report from the receiver
- Delay variance report from the receiver
- Decoding deadline missed (i.e., synchronization failure) report from the receiver
- Request from the receiver to change the coding format

2.4.3 Synchronization actions at the intermediate mixers and translators

In a group communication, which involves a large number of participants, it becomes quite difficult for each user to receive all the media streams and decode them. In such a scenario, it is much better if all the out going media streams are collected by a powerful Media Processor (MP) (called mixer), which decodes the active media streams and composes them into a single stream. Finally the composed stream is encoded by the mixer and transmitted to all the receivers. Mixer MP is controlled by Multipoint Controller Unit (MCU), which decides which streams are active and need to be mixed. Sometimes, there is also a need to change the coding format of the encoded stream. For example, if a stream is being routed from a high bandwidth network to a low bandwidth network, then there is need to re-code that stream at a low bit-rate. Media processors (called translators) situated at the gateways or network edge devices achieves re-coding functionality. Intermediate mixers and translators can reduce the asynchrony in the media streams by following the procedures followed by the receiving and transmitting ends since they behave as the receiving as well as the transmitting end.

[1] Real-time Transport Protocol: A Transport Protocol for real-time applications.

[2] Real-time Transport Control Protocol: Associated control protocol of RTP.

2.4.4 Synchronization actions at the network sub-system

Although no synchronization action is taken at the network sub-system, certain actions taken in the network sub-system are preventive in nature. For providing guaranteed QoS, the network sub-system reserves adequate buffers at the switches and routers and gives high scheduling priority to packets belonging to delay sensitive media streams. The communication (and the interface) between the codec and network sub-system to activate an adjustment of control parameters is an issue which the codec and networking groups need to investigate.

2.5 Recovery actions

When a synchronization failure is detected, recovery action is performed to overcome its effects. Some of the recovery actions found in the literature are discussed briefly below:

- **Frame interpolation**: If at the i^{th} interval, data to be presented does not arrive before the play-out time and the deadline for that data is missed, then from the previous buffered frames, a frame is predicted and played at the i^{th} interval. This is enabled only if the most recent data slip (if any) occurred farther than inter-glitch spacing (IGS). If the asynchrony of presentation exceeds the tolerance specified by Divergence Vector of the media streams, then either data frames are dropped from leading streams or frames are added (by frame prediction) in the lagging streams. This method of synchronization is called stuffing.
- **Persistent slippage:** If data slippage becomes too frequent it may not be possible for the synchronization mechanism to recover gracefully. It indicates that the QoS provided by the network layer is not sufficient for the current presentation. The synchronization mechanism should recalculate QoS and re-negotiate the QoS parameters. If the negotiation fails then application control is informed, which may decide to degrade the quality of presentation or to abort the presentation.
- **Advancing to the next temporal interval:** If too many deadlines are missed for most of the streams, then after completing the $\mathbf{i^{th}}$ interval, temporal position is advanced to the $\mathbf{(i+1)^{th}}$ interval by restarting the frame time-out and retaining the segments in the range specified by tolerable skew in the playback buffer. Any subsequently arriving packet that belongs to $\mathbf{i^{th}}$ interval is retained for a possible play-out in the $\mathbf{(i+1)^{th}}$ interval if the frame for the later interval does not arrive.
- **Control of frame time out:** Clock drifts between the transmitter and receiver can be compensated by controlling the frame time-out at the destination. The inter-frame pause (IFP) specifies the time by which frame time-out cannot be exceeded. If network delay increases, the destination frame interval times out before packets in that frame arrive. The chance of subsequent frames also missing their time-out increases. Delaying the starting point for the next frame time-out by an appropriate compensatory period increases the probability that subsequent frames are received within their frame interval.

3 Error Resilient Mechanisms

This Section discusses the effects of network errors on video coding. It will examine the nature of these errors and investigate the mechanisms that help minimize the effect of network errors. Error resilient mechanisms enhance video-coding techniques for acceptable performance in erroneous environment.

3.1 Network errors

3.1.1 Data corruption

The primary reason for data corruption in the network is bit-corruption (due to the noise in the communication channel). Wrong synchronization of the transmitter and the receiver clocks also leads to data corruption. The receiver may not be aware that it has lost synchronization with the transmitter and hence may forward garbage data to the higher layer.

If the header of a packet is corrupted, then the data contained in the body of the packet will be meaningless to the receiver. Most of the packets/cells have CRC code to detect bit-corruption. Once the bit-corruption has been detected, it needs to be corrected. One way to recover lost bits is to retransmit the corrupted packet/cell. This does not seem to be a feasible solution for multimedia applications that have real-time constraints and applications that do group (multicast) distribution. Retransmitted packets may be delayed beyond acceptable limits and hence may be discarded at the receiving end. In the case of group distribution, retransmitted packets may flood the network.

Another way to overcome bit corruption is to use FEC (Forward Error Correction), but it has drawbacks of large overhead in terms of channel bandwidth and computation power. Moreover, FEC is not useful when garbage data is forwarded to the application or when a data packet is lost.

3.1.2 Data packet loss

It has been observed that retransmission of lost packets does not seem to be a feasible solution. Such losses should be handled at the application layer. To recover from such losses, additional information is required by the decoding process. The encoding process should generate additional and redundant information (required by the decoding process).

3.1.3 Failure in timely delivery of data packets

The effect of a delayed packet (cell) is similar to that of lost packet. Since multimedia applications have real-time constraints in terms of end-to-end delay, a packet arriving after its playback time can not be presented to the end user and is usually discarded, i.e., not presented to the end user but may be decoded if it is required to decode other frames. In a live communication, there is a stringent end-to-end delay limit but this is not the case for stored media streams. They can have larger end-to-end delay (i.e., playback of stored multimedia steams can afford retransmission) but this delay is limited by the buffering capacity of the receiving end terminals.

Once an error is recognized in the input video bit-stream, various techniques can be applied to conceal the error and minimize its effect. Error resilience techniques assist decoders in decoding erroneous media bit-streams (delivered by the transmitting medium). Before looking into the error resilient techniques we will have a brief overview of MPEG video coding.

3.2 MPEG video coding [8]

The proposed MPEG standard video coding scheme is a motion-compensated discrete cosine transform (DCT) and DPCM (Differential pulse code modulation) hybrid coding algorithm. In MPEG coding, the video sequence is first divided into groups of pictures or frames (GOP) as shown in Fig 6. Each GOP may include three kinds of coding modes of pictures: intra-coded (I) picture, predictive-coded (P) and bi-directional predictive-coded (B) picture. I-pictures are coded by intra-frame techniques only, with no need for previous information. They are used as anchors for forward and/or backward prediction. P-picture are coded using forward motion-compensated prediction from a past I- or P-picture and they are also used as anchors for forward and/or backward prediction for B-pictures. The prediction mode can change for different parts of the picture.

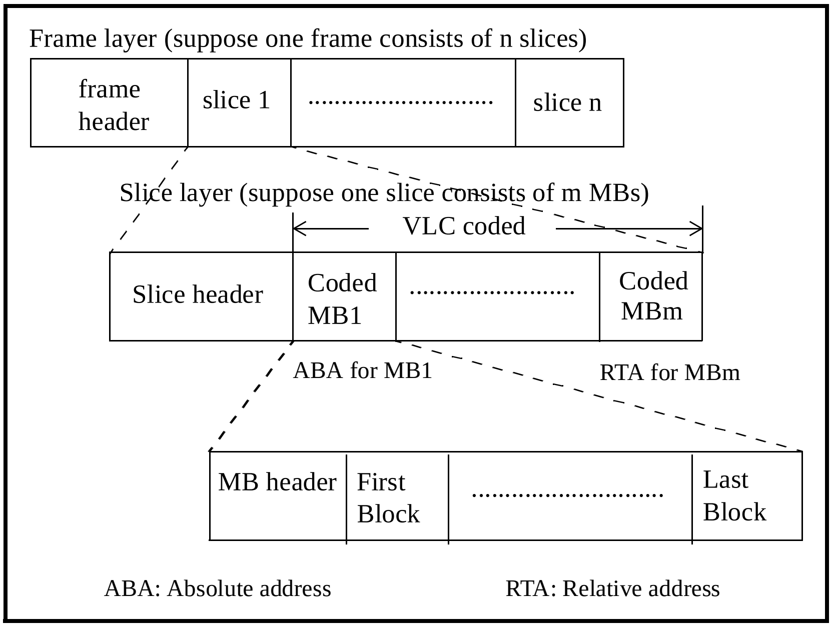

Fig. 5: MPEG video bitstream after VLC coding.

The MPEG algorithm processes the frames of a block-based video sequence. Each input video frame is partitioned into non-overlapping macroblocks (MB). Each macroblock contains blocks of data from both luminance and co-sited chrominance bands: four luminance blocks $\boldsymbol{(Y_1, Y_2, Y_3, Y_4)}$ and two chrominance blocks ***(U,V)***, each with a size of 8x8 pels. Thus, the sampling ratio between ***Y:U:V*** luminance and chrominance pels is 4:1:1. The bitstream syntax of coded data after variable length coding in shown in Fig. 5. Adjacent MB's are grouped into a slice. A frame consists of a number of slices proceeded by a frame header. Similarly, a slice consists of a number of macroblocks proceeded by a slice header. Each macroblock also begins with a header, which includes information on the macroblock location (MB address), and motion vectors for use in the motion compensation prediction. In the first macroblock of each slice, the MB address and motion vector are coded absolutely. In each of the remaining macroblock in a slice, these parameters are coded differentially with respect to the corresponding values in the MB immediately before it.

The structure of MPEG implies if an error occurs within I-picture data, it will propagate through all frames in the group of pictures. Similarly, an error in a P-picture will affect the related P- and B-pictures, while B-picture errors are isolated. Therefore, effective concealment of lost data in I- and P-pictures to avoid error propagation is of critical importance.

3.3 Error concealment [9, 10]

These techniques attempt to conceal errors by taking into account of the remaining spatial correlation in the same frame and temporal correlation in the previous frame of a video sequence.

3.3.1 Temporal and spatial concealment

In areas of the picture that do not change very much with time, it is effective to conceal the effect of packet (cell) loss by temporal replacement, i.e., using the corresponding information from the previous frame. This approach is not effective in high motion areas. In such a case, spatial interpolation tends to be more effective. Both of these techniques are not very effective.

3.3.2 Motion compensated concealment

Motion compensated concealment, which combines temporal replacement and motion estimation can improve the concealment. In this technique, the motion vectors above and/or below the lost macro-block (MB) are used to predict the motion vector on the lost MB, and a motion compensated concealment strategy is used. This improves the concealment in the moving areas of the video sequence, but is unable to conceal errors for a lost MB, which is surrounded by intra-coded macro-blocks. (This is because the intra-coding process does not supply motion vectors.) To overcome this, the encoding process can be extended to include motion vectors for intra-coded MBs. Of course, motion vectors for intra-coded MBs are only for error concealment. This scheme has the drawback of low efficiency in coding. In addition, it will be better if the motion vector and coded information for a particular intra-coded MB are transmitted separately so that the motion vector is still available in the event of a coded macro-block data loss.

3.4 Error localization

In this technique efforts are made to curtail the effects of an error in one MB on other MBs, as shown in Fig. 6.

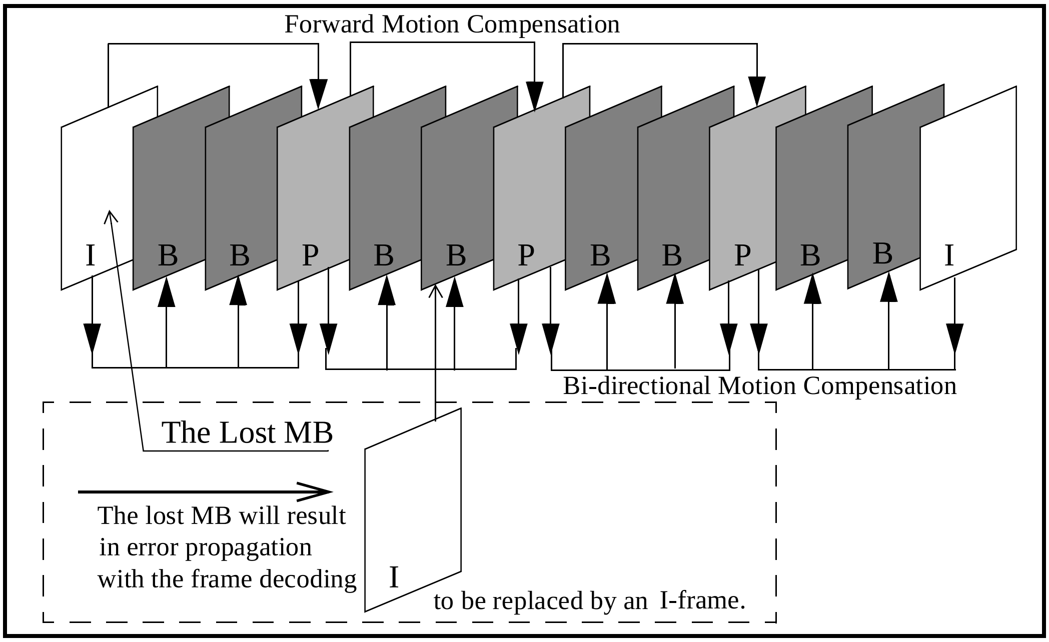

Fig. 6: Frame order in a sequence and error propagation.

3.4.1 Temporal localization

One significant effect of cell loss and erroneous bit-streams on the decoded sequence is error propagation. By considering the processed frame order of MPEG and H.26x coding schemes, it can be seen that errors occurring in I-pictures can propagate into the following P- or B-pictures, since P- and B-pictures are predicted from I-frames. Similarly, errors in P-pictures can propagate into subsequent B- and P-pictures. Temporal localization seeks to minimize error propagation from picture to picture in the temporal domain by providing early re-synchronization of pictures that are coded differentially. The various techniques for temporal localization of error are:

- **Cyclic Intra-coded pictures:** Extra intra-coded I-pictures can be sent to replace B- or P-pictures. Thus error propagation is stopped at the expense of extra-transmitted bits.
- **Cyclic Intra-coded slices/GOBs[1]:** Instead of using intra-coded frames to limit the effect of errors in the image, extra intra-coded slices/GOBs can be used periodically to refresh the frame from top to bottom over a number of P-pictures. The disadvantage is that this partially update of the screen in a frame period will produce the subjectively unpleasant “wind-screen wiper” effect.
- **Feedback from the decoders:** Instead of cyclic intra-coding pictures or slices/GOBs, both of which reduce the coding efficiency and increases the probability of network congestion and erroneous bits, it is better to intra-code only the erroneous macro-blocks, based upon feedback from the decoder. While an erroneous MB is intra-coded and transmitted, temporal/spatial concealment can be used temporarily.

[1] Group of Blocks.

- **No inter-frame coding:** Although completely contrary to the spirit of MPEG and H.26x compressions, turning off inter-frame coding completely removes the possibility of temporal error propagation. It has the drawback of low coding efficiency.

3.4.2 Spatial localization

Spatial localization encompasses methods aimed at minimizing the extent to which errors propagate within a picture by providing early re-synchronization of the elements in the bit-stream that are coded differentially between MBs. Re-synchronization involves two features. The first is an unambiguous indication of the location of macro-blocks in the bit-stream where re-synchronization is possible. The second is to code absolutely those quantities that are normally coded differentially with respect to the previous MB (e.g., motion vectors), for this re-synchronizing MB.

- **Unambiguous indication of the location of the macro-block:** The most basic method for achieving spatial localization of errors is to reduce the number of MBs in a slice/GOB. Suppose that a slice of coded video data is divided into two small slices. If the first cell is lost or corrupted, the decoding procedure can resume at the slice start in the second cell. This protects the second half of video slice data from being discarded. On the other hand, if both slice headers are packed in one cell, for example the first cell, then if data loss occurs in this cell, the simple slice method cannot re-synchronize until the end of the entire slice of data.
- **Absolutely code differential coded elements in the first MB of a packet:** The small slice scheme does not take into account packing of bits into transport stream packets, or of the packing of these packets into smaller cells. To address this point, an improvement over the fixed small slice method can be developed. The first MB coded in every cell will be coded absolutely by the encoder. That is to say, once a cell loss or erroneous bits occur within a cell, the decoding procedure can resume at the first macro-block in the next cell. When ***m*** consecutive cells are lost, the amount of information that becomes unavailable to the decoder will always be less than ***(m+1)*** cells. This technique is known as the MB-synchronization method and it is not compatible with MPEG and H.26x standards. If RTP is used to packetize and transport the coded bit-stream, then information about differential coded elements can be passed to the decoder. A RTP packet header carries the value of the elements (from the MB of the previous packet) that help decode differential coded elements of the first MB of the packet.
- **Adaptive slice/GOB sizes:** The use of adaptive slice sizes is similar to the MB-re-synchronization method, but a change in the slice/GOB size depends on the length of the cell. The encoder will trace the coding process to place the slice start code at the first opportunity in each cell. This technique can achieve essentially the same results as MB-re-synchronization, but at the cost of considerable higher overhead caused by the large size of the additional slice headers that must be transmitted. The overhead of implementing this technique would be between five to eight bytes per cell, depending on the coding mode used.

3.5 Unequal error protection

In video coding, some information is more important than others. For example, a loss of information about the quantization levels or motion vectors is more damaging than a loss of information about DCT coefficients. Even for DCT coefficients, a DC coefficient of lower frequency contributes more to the subjective and objective quality of the video than a higher frequency coefficient. The header of a multiplexed[1] packet usually contains bits describing the type of data contained in the packet. If this header is corrupted, the data in the body of the packet becomes meaningless to any higher level decoding process.

Hence, when the capacity of the channel is limited, it makes sense to give different levels of protection to the data according to their importance. Scalable coding provides unequal error protection, where the bit stream is divided into several layers. The quality of received video increases as data from successively higher layers are received. Information in lower layers is usually more important, and can be given higher protection. This approach works well when different priority levels are available such as in priority-controlled cell-based transmission in the Broadband-ISDN.

Another method[2] to give extra protection to important information like system headers is to code such information using Forward Error Corrections (FEC) codes. This will help in recovering from bit corruption but not from cell loss.

3.6 Use of fixed length codes

The use of synchronization codes and periodic restarting is insufficient to prevent catastrophic losses due to all types of errors. Certain measures have to be introduced to target errors caused by the usage of variable length codes. Variable-length codes allow the propagation of errors because errors in a codeword can cause a loss of synchronization. A simple way to prevent this, possibly at a cost of some coding efficiency, is to use fixed-length codewords. This is done for vector quantization and for coding sparse data. It is also possible to rearrange the variable length codes so that the codewords start at known positions in the bit-streams.

[1] A multiplexed packet contains chunks of data multiplexed from “*n*” number of streams.

[2] Rate-compatible punctured convolution (RCPC) also provides unequal protection.

4 End-to-End Delay [12, 13]

In the following Section, we will discuss the processes (as shown in Fig. 7) that contribute to end-to-end delay and the measures that can be taken to minimize it.

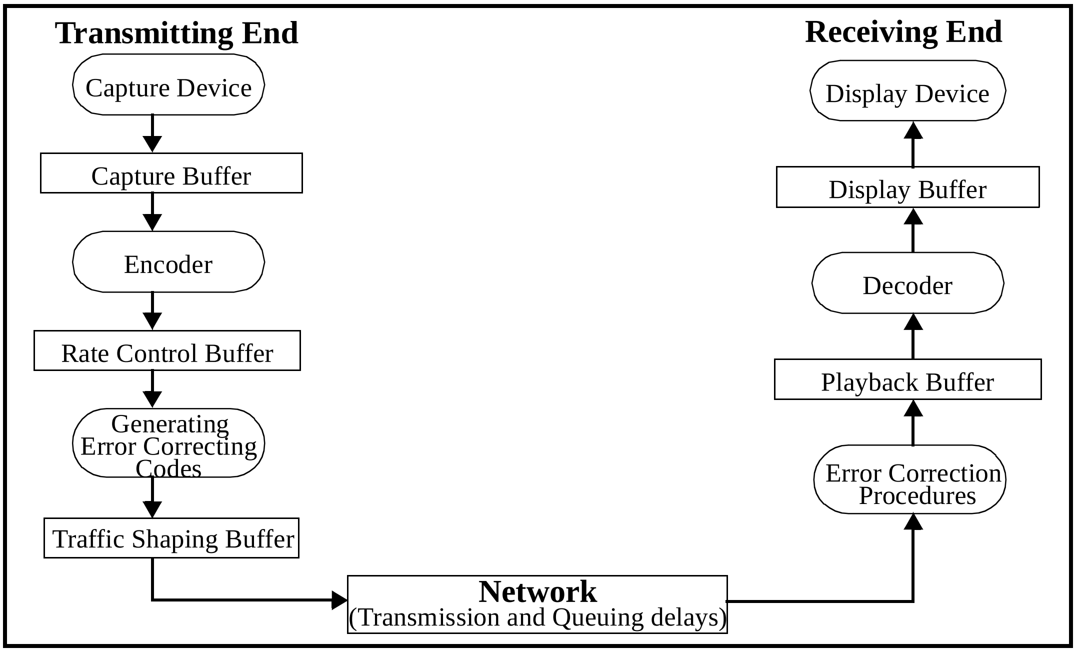

Fig. 7: End-to-end delay components.

4.1 Buffering before encoding (No. of B frames buffered)

Video streams that are coded using compression standards like MPEG and H.263, which use bi-directional prediction, introduce additional delay. As B frames are predicted from two other I/P frames, for the B frame to be coded the future I/P frame from which backward prediction is made should also be captured. The larger the number of B frames between I/P frames, the larger is the end-to-end delay. Decreasing the number of B frames decreases coding efficiency but it reduces the end-to-end delay.

4.2 Audio/Video capture buffer size

If the capture buffer captures ***T*** units of time of media data from input media device (assuming that there is no collection, transmission and presentation delay), then the playback of the media stream by the receiver can start only ***T*** units of time after its generation time. The larger the length of the capture buffer, the larger is the capturing time and hence larger the end-to-end delay. One cannot have very small capture buffer size as capture device may not support capturing small chunks of data and moreover it increases the overhead of context switching.

4.3 Encoding time

Encoding of video frames is a CPU intensive process. It is a time consuming process and adds to the end-to-end delay. If the encoding is done on a general-purpose system then the operating system must have real-time capabilities to reduce the delay in encoding process by giving it due priority. Switching off the advanced features of the encoder can also reduce encoding time, but this will have the drawback of decrease in coding efficiency and picture quality.

One way of reducing effective coding (encoding and decoding) time is by using the pipeline technique. In this technique a video frame is encoded in smaller chunks and each chunk is packetized and transmitted on the network as soon as it is encoded. At the receiving end, decoding starts as soon as the first packet of a video frame is received (the decoding process does not wait for the arrival of all chunks of the frame), but the frame is not displayed until all the chunks of the frame are decoded.

Let (E_t) be the encoding time of a frame, (D_t) be the decoding time of the frame and (n) be the number of chunks into which the frame is fragmented. If each chunk is packetized and transmitted as soon as it is encoded, then the decoding of the first chunk of the frame can start (E_t/n) units of time after the encoding start time (assuming no other delays).

If no fragmentation is done, then encoding and decoding of a frame will take (E_t+D_t) units of time. Assuming that decoding time of a chunk will be less than its encoding time, the time taken for complete encoding and decoding in the fragmented case will be $(E_t+(D_t/n))$ units of time, as the effective decoding time is reduced to (D_t/n) units of time. Hence, end-to-end delay is reduced by $(D_t*(n-1)/n)$ units of time. Although the formulae may suggest usage of large values of (n)**,** but using large value of (n) has a serious drawback. As the number of packets transmitted increases, effective bandwidth decreases due to large overhead of the packet header in each packet.

4.4 Transmit buffer

The output bit-rate of most video encoders is variable and it depends on the type of coding used and the amount of redundancy present in the input video stream, but the encoded data needs to be transmitted over the network at a constant rate conforming to a negotiated bandwidth. The output rate of the encoder is controlled by rate control mechanisms whereas the data generated by the encoder is shaped by traffic shaping algorithms. For traffic shaping and rate control, some data packets need to be buffered before being transmitted over the network. The larger the buffering done, the larger is the end-to-end delay, but better traffic shaping and rate control is achieved. Rate control is based on the watermarks of the transmit buffer and is done at the application level on the user's host machine. Traffic shaping is based on the leaky bucket algorithm and buffering is done at the user to network interfaces. There is a need to come up with mechanisms that will address both these issues and use a common buffer.

4.5 Error correction processing delays

The processing time needed to generate error correcting codes also contributes to the end-to-end delay and the time taken by the error correction process depends upon the complexity of the algorithm used. The error correcting codes that are generated and packetized into packets are decoded at the receiving end.

4.6 Network delay

Network delay is one of the major contributors of end-to-end delay and it is difficult to deal due to its varying nature. Network access time, transmission/propagation time, queuing delay, fragmentation, and reassemble time contribute to the network delay.

4.7 Decoding delay

The decoding process also adds to end-to-end delay. In case of video streams, smaller chunks of a video frame cannot be displayed[1] as soon as they are decoded and hence the saving in the end-to-end delay (due to the pipeline effect of fragmentation) is as shown Section 4.1.3. Such a restriction does not exist in audio streams and they can be played as soon as a chunk is decoded. The smaller the chunk size, the smaller will be the effective decoding delay. But, there are certain restrictions that make it difficult to reduce the frame size beyond certain limits e.g., the capturing device may not support small size frame capturing.

Let (E_t) be the encoding time of a frame, (D_t) be the decoding time of the frame and (C_t) be the capturing time of the frame. Also assume that there are no other delays and there is no restriction on the size of output buffer that can be played by the output audio device. Audio data can be played back $(C_t+E_t+D_t)$ units of time after its generation time. In this case, if the captured frame is divided into (n) smaller chunks, then effective coding time can be reduced to $((E_t/n)+(D_t/n))$ units of time. The audio stream can be played back $(C_t+(E_t/n)+(D_t/n))$ units of time after its generation time (due to pipeline coding mechanism discussed in Section 4.1.3). But, smaller capture frame size has the overhead of large number of context switches and packet headers.

4.8 Playback buffering and synchronization delays

To avoid all glitches in the playback stream, a stream should have a buffering capacity for $(D_{max} - D_{min})$ units of time, where D_{max} is the maximum delay and D_{min} is the minimum delay experienced by data packets.

[1] If small chunks of a video frame are displayed as soon as they are decoded, then the picture will have “block wrapping” effect.

5 QoS and QoP Requirements

Quality of Presentation (QoP) defines the end user's presentation requirements of multimedia streams whereas Quality of Service (QoS) is a specification of the requirements of an application from the network sub-system in order to support a required level of performance.

5.1 Quality-of-Presentation (QoP) [14]

Quality of Presentation is specified in terms of:

- Frame size (width and height in pixels)
- Frame rate
- End-to-end delay (Latency)
- Allowed degree of asynchrony (skew) between temporally related streams
- Allowed rate of glitches
- Allowed inter-frame pause (IFP)
- Picture quality

Table 3 gives estimates of various QoP parameters for different quality profiles:

Table 3: QoP parameters for different quality profiles

Profiles	Frame Rate	Frame Size	Latency	Color Depth	Skew	Glitch Rate	IFP
High	15-25 fps	CIF (352 x 288)	<200 ms	24 bits	<40 ms	<0.1 glitch per minute	<20 ms
Medium	6-15 fps	QCIF (176 x 144)	<400 ms	16 bits	<80 ms	<0.5 glitch per minute	<40 ms
Low	< 6 fps	Sub-QCIF (128 x 96)	<800 ms	8 bits	<200 ms	<2 glitch per minute	<100 ms

The quality of video picture are estimated by the following characteristics:

- **Video artifacts:** These appear as blocks (macrocells), color splotches, image distortions, or patches which are grossly out of focus.
- **Sharpness:** Ideally it should be possible to see individual hairs on the speaker's head. The line of her shoulder should be sharp and smooth. Eyes should be crisp and clear.
- **Contrast, brightness, and color saturation**
- **Stability:** The picture should not shimmer or deform over time.

5.2 Quality-of-Service (QoS) [15]

Quality of Service is specified in terms of:

- Bandwidth
- Error rate
- End-to-end delay
- Delay jitter (variance)

Table 4 gives estimates of various QoS parameters for different quality profiles:

Table 4: QoP parameters for different quality profiles

Profiles	Bandwidth (Average)	Error Rate	End to End Delay	Delay Jitter
High	128-384 kbs	10^{-12}	<150 ms	<50 ms
Medium	64-128 kbs	10^{-10}	<300 ms	100-200 ms
Low	20-32 kbs	10^{-8}	<600 sec	250-500 ms

Networks can be classified into three categories according to the level of guarantee provided.

- Networks with no guarantee, e.g., LANs and Internet. It is most difficult to provide synchronized multimedia presentations in such an environment. All the efforts for synchronizing the presentation should be made in the application layer based upon the current characteristics of the network[1].
- Networks with full guarantee for requested QoS, e.g., ISDN channels and dedicated leased lines. Synchronized multimedia presentation does not pose much of a challenge in such an environment.
- Networks with partial guarantee for requested QoS, e.g., ATM networks, provide the most realistic situation with appropriate cost. For such a network, there is a need to come up with a model which will map the QoP requested by the user and the resources available at the encoding and decoding ends to the minimum QoS requirement from the network. The model should specify the coding (and it control parameters) and synchronization mechanisms (algorithms) to be used to achieve the desired level of performance. There can be a vice-versa requirement too; given the supported QoS from the network and the resources available at the encoding and decoding ends, what will be the maximum QoP that can be delivered to the end user.

Depending upon the QoP requested by the user, QoS supported by the network, and the resources available at the encoding ends, the control parameters of the encoder are tuned to achieve the desired level of performance.

[1] Profiling is needed to find the current characteristics of the network.

The encoder parameters that can be controlled are:

- Quantization level
- Rate at which pictures/slices/GOB/macro-blocks are coded as inter or intra. In case of inter picture, their type - P or B.
- Coding algorithm to be used (it depends upon available CPU power and the desired bit-rate)

The codec control parameters are tuned based upon the following events and parameters:

- QoP requested by the user
- QoS that can be provided by the network
- Processing power of the encoding and decoding ends
- Feedback from the network sub-system and the decoding end

Adjustments that can be made in response to the changing environment are:

- Change of QoP
- Change of QoS (Renegotiations)
- Change of codec parameters

The various trade-off in dynamically adapting the functionality of a media codec is presented in Table 5. It shows that adapting the codec functionality is no trivial task and it needs good understanding and research in this area to come up with a good model to map the changing environment requirements to the control parameters of the codec.

Table 5: Trade-off for the various control parameters

Control Parameter	Trade-off
Quantization level	Picture quality and bit-rate
Picture/MB type	Bit-rate, CPU power and desired resilience to error
Frames skipping	Quality of presentation and bit-rate
Advance mode usage	The CPU power available at the coding ends, bit-rate and desired level of error resilience
Error recovery mechanism	The CPU power available at the coding ends, bit-rate and desired presentation synchronization

6 IP Conferencing Standard [16]

ITU-T Study Group 15 is developing H-series Recommendations that allow internetworking between different audiovisual communication terminals manufactured by different equipment providers. This section focuses on H.323, which is a multimedia conferencing over non-guaranteed networks such as IP packet networks like the Internet or corporate LAN. H.323 covers audio, video, and data conferencing. However, to be H.323-compliant, a device must support voice; video and data support is optional.

H.323 was finalized by the ITU-T in October 1996. H.323 specifies the modes of operation that are required for endpoints from different vendors to intercommunicate with any combination of audio, video, data and graphics. It provides the call-model descriptions, the call-signaling procedures, and the system and component descriptions for packet-based conferencing. H.323 does not directly include standards for guaranteeing Quality of Service (QoS).

6.1 H.323 Components

The H.323 umbrella standard incorporates already established recommendations and protocols to carry out its intention to specify conferencing over packet-switched networks. H.323 uses the following protocols and standards:

- **H.225.0/Q.931:** For call signaling and call setup.
- **H.245:** For conference and call control. It describes the messages and procedures used to negotiate channel usage; for opening and closing logical channels for audio, video, and data; for capabilities exchange; for mode requests; for control; and for indicators.
- **H.225.0/RAS:** Messaging for registration, admission and status with a Gatekeeper.
- **RTP/RTCP:** (Real time protocol/real time control protocol) For media stream packetization and synchronization of streams.
- **Video Codec:** H.261 and H.263.
- **Audio Codec:** G.711, G.722, G.723, G.728, and G.729.

6.2 Network Components

Recommendation H.323 describes the network components that connect to a LAN employed for interaction within a network. It does not describe the LAN itself or the transport layer used to connect various LANs. To implement an IP network-based communication system, H.323 defines these four major components:

- **H.323 endpoints**
- **Gateways**
- **Gatekeepers**
- **Multipoint control units**

6.2.1 H.323 Endpoints

Endpoints on the LAN, whether they are integrated into personal computers or implemented in stand-alone devices, support real-time, bi-directional communication. H.323 endpoints must support H.245, Q.931, RAS, RTP/RTCP, and G.711. By supporting these H.323 standard protocols and through an appropriate gateway, H.323 endpoints can interoperate with these endpoints: H.320 on narrow-band ISDN (N-ISDN), H.321/H.310 on broadband ISDN (B-ISDN) using an asynchronous transfer mode (ATM), H.322 on guaranteed QoS LANs (IsoEthernet), H.324 on general switched telephone network (GSTN).

6.2.2 Gateways

H.323, with its infrastructure of routers and switches, expands on this implementation and embeds gateway technology into the world of standards-based conferencing over IP networks. For H.323, gateways manage inter-operation between ITU-T endpoints by translating the call signaling, control channel messages, audio compression algorithms, and multiplexing techniques between an IP-based endpoint and an endpoint connecting through an ISDN. As a result of gateway services, H.320 systems can communicate with packet-based H.323 systems. H.323 mandates endpoint requirements to minimize the transcoding that the gateway must perform to achieve interoperability. The gateway may not need to transcode audio when conference endpoints communicate with a common mode. In some cases, however, the gateway may perform audio transcoding so that each endpoint can operate with its optimum bandwidth efficiency. H.323 does not mandate the number and types of interfaces that a gateway can manage. In actual practice, a gateway can support several concurrent LAN-CSN (Circuit Switched Network) sessions.

6.2.3 Multipoint Control units (MCU)

The MCU is a server that bridges signaling and media among three or more sites. The H.323 Recommendation specifies the two parts of an MCU: Multipoint controller (MC) and Multipoint processor (MP). Through H.245, the MC negotiates among conference endpoints, determines common audio and video capabilities, and establishes media channels. An MC is required for all multipoint conference types but may be located in an endpoint, a gateway, or a gatekeeper. Whereas the MP mixes and switches audio, video, and data streams. An MCU may consist of an MC or an MC and one or more MPs. H.323 does not standardize communications between the MC and the MP. An MCU can also be known as a multimedia conference server (MCS).

6.2.4 Gatekeepers

Gatekeepers perform management services for an H.323 conferencing zone (A network segment - the collection of endpoints, gateways and MCUs). A gatekeeper is an optional element in H.323. When a gatekeeper is enabled in an IP network, all H.323 endpoints contacting that network must make use of it. The gatekeeper helps to preserve the operational quality of the LAN by performing admissions control and authorizing access to the LAN for H.323 endpoints including gateways and MCUs. The gatekeeper not only limits the amount of bandwidth these entities use on the network, but guarantees access only to recognized entities. The gatekeeper grants permission for both placing and accepting calls from H.323 endpoints. For a connection to be successful, an H.323 endpoint must be recognized by the gatekeeper and must also be registered in the gatekeeper's zone. A zone can have only one gatekeeper. The gatekeeper performs bandwidth management through admissions control and ensures that bandwidth is available within its H.323 zone for email, file transfers, and other designated applications. While the gatekeeper can modify the bandwidth usage during a call, the criteria for doing so is not specified in H.323. The gatekeeper as defined in the RAS specification also performs address translation. It accepts both external E.164 telephone number (addresses received from endpoints outside the LAN) and alias name (addresses from LAN endpoints). It then translates the numbers and names to network recognizable address, e.g., an IP address.

7 Conclusion

In summary, we have presented limitation of current networks (due to unreliable, non-guaranteed and error prone channels) in transmitting real-time multimedia coded streams to remote receivers. We also gave an overview of the issues involved in transmitting coded media streams on current networks. Issues such as synchronization, error resilient mechanisms and end-to-end delay that affect media coding and its transmission to remote receivers were discussed in detail. Quality-of-Service and Quality-of-Presentation issues were also analyzed for acceptable playback performance of media codecs on general-purpose computers. This paper also presented issues, which needs to be researched to have acceptable performance of multimedia presentation given the limitation of current networks.

References

1. G. Blakowski and R. Steinmetz, A Media Synchronization Survey: Reference Model, Specification and Case Studies, IEEE Journal on Selected Area in Communications, Vol. 14, No. 1, pp. 5-35, January 1996.
2. S. Baqai, M. F. Khan, M. Woo, S. Shinkai, A. A. Khokhar, and A. Ghafoor, Quality-Based Evaluation of Multimedia Synchronization Protocols for Distributed Multimedia Information Systems, IEEE Journal on Selected Area in Communications, Vol. 14, No. 7, pp. 1388-1403, September 1996.
3. I. F. Akyidiz and W. Yen, Multimedia Group Synchronization Protocols for Integrated Services Networks, IEEE Journal on Selected Area in Communications, Vol. 14, No. 1, pp. 162-173, January1996.
4. R. Steinmetz, Human Perception of Jitter and Media Synchronization, IEEE Journal on Selected Area in Communications, Vol. 14, No. 1, pp. 61-72, January 1996.
5. K. Ravindra and V. Bansal, Delay Compensation Protocols for Synchronization of Multimedia Data Streams, IEEE Transactions on Knowledge and Data Engineering, Vol. 5, pp. 574-589, August 1993.
6. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, Request for Comments: 1889, Audio-video Transport Working Group, January 1996.
7. H. Schulzrinne, RTP Profile for Audio and Video Conferences with Minimum Control, Request for Comments: 1890, Audio-video Transport Working Group, January 1996.
8. J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, MPEG video compression standard, Chapman & Hall, New York, 1997, pp. 17-30.
9. W. S. Lee, M. R. Pickering, M. R. Frater, and J. F. Arnold, Error Resilience in Video and Multiplexing Layers for Very Low Bit-Rate Video Coding Systems, IEEE Journal on Selected Area in Communications, Vol. 15, No. 9, pp. 1764-1774, December 1997.
10. J. Zhang, M. R. Frater, J. F. Arnold, and T. M. Percival, MPEG 2 Video Services for Wireless ATM Networks, IEEE Journal on Selected Area in Communications, Vol. 15, No. 1, pp. 119-128, January 1997.
11. ITU-T SG XV, Video Coding for low bitrate communications, DRAFT ITU-T Recommendation H.263, May 1996.
12. S. B. Moon, J. Kurose, and D. Towsley, Packet audio playout delay adjustment: performance bounds and algorithms, Multimedia Systems, Vol. 6, No. 6, pp. 17-28, September 1998.
13. P. L. Tien and M. C. Yuang, Intelligent Voice Smoother for Silence-Suppressed Voice over Internet, IEEE Journal on Selected Area in Communications, Vol. 17, No. 1, pp. 29-41, January 1997.
14. K. Nakazono, Frame rate as a QoS parameter and its influence on speech perception,

Multimedia Systems, Vol. 6, No. 6, pp. 359-366, September 1998.

15. A. Hafid and G. V. Bochmann, Quality-of-Service adoption in distributed multimedia applications, Multimedia Systems, Vol. 6, No. 6, pp. 299-315, September 1998.
16. S. Okubo, S. Dunstan, G. Morrison, M. Nilsson, H. Radha, D. L. Skran, and G. Thom, ITU-T Standardization of Audiovisual Communication Systems in ATM and LAN Environments, IEEE Journal on Selected Area in Communications, Vol. 15, No. 6, pp. 965-982, August 1997.