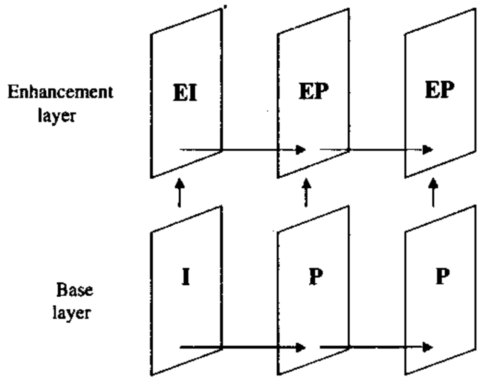


Figure 5.115 SNR scalability [5.176]. ©1997 ITU-T.

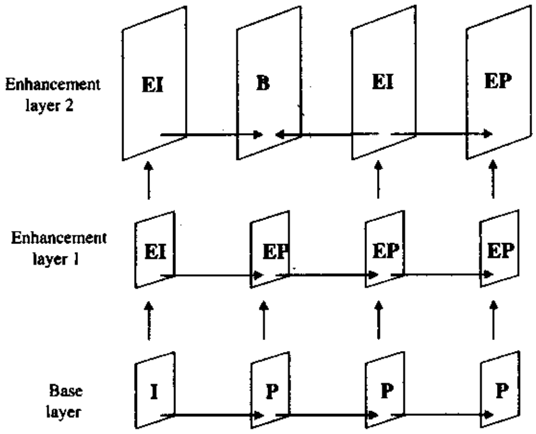


Figure 5.116 Multilayer scalability [5.176]. ©1997 ITU-T.

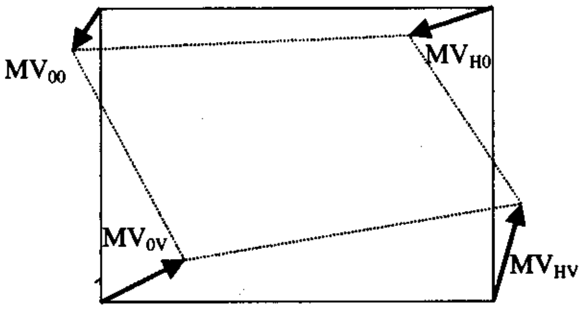


Figure 5.117 Reference picture resampling [5.176]. ©1997 ITU-T.

motion vectors that specify the amounts of offset of the four corners of the reference picture. This mode allows an encoder to switch smoothly between different encoded picture sizes, shapes and resolutions. It also supports a form of global motion compensation and special-effect image warping.

Reduced-resolution update mode—This mode allows the encoding of interframe difference information at a lower spatial resolution than the reference frame. It gives the encoder the flexibility to maintain an adequate frame rate by encoding foreground information at a reduced spatial resolution while holding onto a higher-resolution representation of more stationary areas of a scene.

One important feature of H.263 Version 2 is the usage of supplemental information, which may be included in the bit stream to signal enhanced display capabilities or to provide tagging information for external usage. For example, it can be used to signal a full-picture or partial-picture freeze or a freeze-release request with or without resizing. It can be used to label a snapshot, the start and end of a video segment and the start and end of a progressively refined video. The supplemental information may be present in the bit stream even though the decoder may not be capable of providing the enhanced capability to use it, or even to interpret it properly. In other words, unless a requirement to provide the requested capability has been negotiated by external means in advance, the decoder can simply discard anything in the supplemental information. Another use of the supplemental enhancement information is to specify a chroma key for representing transparent and semitransparent pixels [5.206].

Another round of H.263 extensions has created a third generation of the H.263 syntax, informally called H.263++. Four key technical areas were identified for further investigation toward possible later standardization: variable transform type, adaptive arithmetic coding, error-resilient VLC tables and deringing filtering.

H.263++ Standard Development

The H.263++ development effort is intended for near-term standardization of enhancements to produce a third version of the H.263 video codec for real-time communications and related non-conversional services [5.207]. Key technical areas showing potential for performance gain of H.263++ are the following [5.208, 5.209]:

Error-resilient data partitioning—Creation of a data partitioned and layered protection structure for the coded data and a longer resynchronization codeword to improve the detectability and to reduce the probability of false detection.

4x4 block-size motion compensation—Long-term picture memories; rate-distortion optimization alterations; motion optimization alterations; a new type of deblocking filter; a new type of intraspatial prediction and some VLC alterations for transform coefficients, motion vectors and coded block pattern.

- *Adaptive quantization*—Rate-distortion optimized quantization and truly optimal rate-distortion trellis encoding for an additive distortion measure

- *Enhanced reference picture selection*—Multiframe motion-compensated prediction and modified interframe prediction method
- *Enhanced scalability*—New P-picture types in enhancement layers
- *IDCT mismatch reduction*—Integer inverse transform
- *Deblocking and deringing filters*—Directional classifications and identifications of outlying values of block corner pixels for spatial treatment
- *Error concealment*—Provides error tolerance

H.26L Standard

The long-term recommendation H.26L (previously called H.263L) is scheduled for standardization in the year 2002 and may adopt a completely new compression algorithm. H.26L is an effort to seek efficient video-coding algorithms that can be fundamentally different from the MC-DCT framework used in H.261 and H.263. When completed, it will be a video-coding standard that provides better quality and more functionalities than existing methods. The first call for proposals for H.26L was issued in January 1998. According to the call for proposals, H.26L is aimed at very low bit-rate, real-time, low end-to-end delay coding for a variety of source materials [5.210]. It is expected to have low complexity permitting software implementation, enhanced error robustness (especially for mobile networks) and adaptive rate-control mechanisms. The schedule for H.26L activities is shown in Table 5.29.

Table 5.29 Schedule for H.26L [5.210].

Year	Schedule
Jan 1998	Call for proposals
Nov 1998	Evaluation of the proposals
April 1999	First test model of H.26L (TML1)
1999-2001	Collaboration phase
Oct 2001	Determination
July 2002	Decision

©1998 ITU-T.

The following technical proposals were evaluated in response to the call for proposals for H.26L [5.211]:

- Modified prediction/transform-based method
- Vector quantization with block approximation either by reference to a codebook or by motion compensation from a previous frame
- Loop-filtering method for reducing the blocking artifacts, corner outliner and ringing noise
- Adaptive scalar quantizer scheme using nonzero-level codebooks

- DCT-based embedded video coder using rearrangement of DCT coefficients
- Rough segmentation affine motion compensation scheme, vector quantization and multishape DCT
- Data partitioning using a data-reordering algorithm
- Video coding using long-term memory for multiple reference frames and affine motion-compensated prediction

Multihypothesis (MH) motion pictures are an extension of P-pictures proposed for H.26L. Each block of an MB can be compensated by a linear combination of two motion-compensated blocks. Conventional B-pictures also employ two linearly combined motion-compensated blocks, but one motion-compensated signal (hypothesis) originates from a future reference frame. In contrast to B-pictures, MH pictures use temporally previous pictures for prediction and cause no extra coding delay. In addition, decoded MH pictures are also used for reference to predict future MH pictures.

MH pictures are shown in Figure 5.118. Two blocks of temporally previous pictures are used to predict the current MH picture. MH pictures are also used for reference to predict future MH pictures. MB modes for MH pictures are presented in Figure 5.119. For each block, a Multihypothesis Block Pattern (MHBP) indicates one or two hypotheses. Seven MH MB types are added to the standardized MB types for intercoding. The additional seven types allow MH motion-compensated prediction for seven different block sizes.

MH pictures as well as P-pictures use temporally previous pictures for prediction. Each block can be compensated by one hypothesis (conventional motion compensation) or two hypotheses. An MHBP indicates one hypothesis or two hypotheses for each block. The MHBP is dependent on the MB type. When the MHBP indicates one hypothesis for a block in the MB, motion vector data and a reference frame parameter are specified. When the MHBP indicates two hypotheses for a block, two motion vectors and two reference frame parameters are indicated.

The additional MH MB types use the code numbers of the universal VLC as specified in Table 5.30.

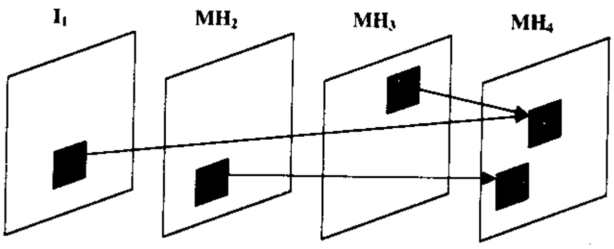


Figure 5.118 M++ pictures [5.212]. ©2001 ITU-T.

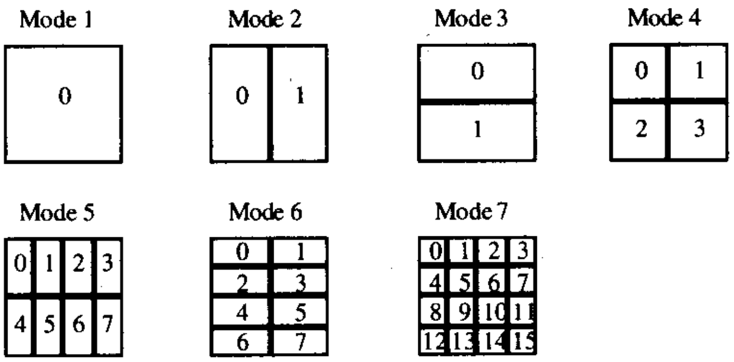


Figure 5.119 MB modes for MH motion pictures [5.212]. ©2001 ITU-T.

Table 5.30 MB types for the MH pictures [5.212].

Code number	MB type
0	Skip
1	16x16
2	MH 16x16
3	MH 16x8
4	MH 8x16
5	MH 8x8
6	MH 8x4
7	MH 4x8
8	MH 4x4
9	8x8
10	16x8
11	8x16
12	8x4
13	4x8
14	4x4
15	Intra4x4

©2001 ITU-T.

Table 5.31 MB types for the MH pictures [5.212].

Code number	MHBP
0	One hypothesis
1	Two hypotheses

The MHBP uses the universal VLC as shown in Table 5.31.

One-hypothesis blocks, as well as two-hypothesis blocks, are independent of the block size individual reference frame parameters and reference frame parameter pairs. The universal VLC code numbers are used for signaling the reference frames.

The encoder has to determine the MH MB type and the number of hypotheses for each block. For a one-hypothesis block, the motion vector data and reference frame parameter are determined by rate-constrained long-term memory motion estimation. An integer-pel accurate estimate for all reference frames is redefined to half-pel and quarter-pel accuracy.

For a two-hypothesis block, the motion vectors and reference frame parameters are determined by rate-constrained MH motion estimation.

Rate-constrained MH motion estimation is performed by an iterative algorithm. The solution fnr the one-hypothesis block is used for initialization. The algorithm works as follows:

1. One hypothesis is fixed and long-term-memory motion estimation is applied to the complementary hypothesis so that the MH rate-distortion costs are minimized.
2. The complementary hypothesis is fixed, and the first hypothesis is optimized.

These two steps are repeated until convergence. Usually, the algorithm converges very fast after one to two iterations.

For the conditional motion estimation, an integer-pel accurate estimate for all reference frames is refined to half-pel and quarter-pel accuracies.

The computational complexity of a two-hypothesis block is just 2 to 4 times the complexity of a one-hypothesis block as the iterative algorithm is initialized with the one-hypothesis solution. However, this complexity can be further reduced by efficient search strategies.

For decoding MH pictures, the decoder has to add two motion-compensated signals. When the MH block pattern indicates a two-hypothesis block, the pixel values of the two motion-compensated blocks are added and divided by 2 (integer division). No additional memory is required.

5.9.4 ITU-T Speech-Coding Standards

The ITU has standardized three speech coders, which are applicable to low-bit-rate multimedia communications. ITU recommendation G.729 8 Kb/s CS-ACELP has a 15 ms algorithmic codec delay and provides network-quality speech [5.213]. It was originally designed for wireless applications, but is applicable to multimedia communication as well. Annex A of recommendation G.729 is a reduced-complexity version of the CS-ACELP coder. It was designed explicitly for simultaneous voice and data applications that are prevalent in low-bit-rate multimedia communications. These two coders use the same bit stream format and can interpolate. The ITU recommendation G.723.1 6.3 and 5.3 Kb/s speech coder for multimedia communications was designed originally for low-bit-rate videophones. Its frame size of 30 ms and one-way algorithmic codec delay of 37.5 ms allow for a further reduction in bit rate compared to the G.729 coder. In applications where low delay is important, the delay of G.723.1 may be too large. However, if the delay is acceptable, G.723.1 provides a lower-complexity alternative to G.729 at the expense of a slight degradation in quality.

An enormous number of new speech coders have been standardized. For example, in the 1995-96 time period, three new international standards (ITU G.729, G.729A and G.723.1) and three new regional standards (enhanced full-rate coders for European and North American mobile systems) have emerged.

Speech quality as produced by a speech coder is a function of bit rate, complexity, delay and bandwidth. Hence, when considering speech coders, it is important to review all these attributes. It is important to realize that there is a strong interaction between all these attributes and that they can be traded off against each other. Additional factors that influence the selection of a given speech coder are availability and licensing conditions the way that the standard is specified. Most speech coders operate at a fixed bit rate regardless of the input signal characteristics. Because multimedia speech coders share the channel with other forms of data, it is better to make the coder variable rate. For simultaneous voice and data applications, a good compromise is to create a silence compression scheme as part of the coding standard. A common solution is to use a fixed rate for active speech and a low rate for background noise. Silence compression consists of two main algorithms. The first is a Voice Activity Detector (VAD), which determines if the input signal is speech or some sort of background noise. If the signal is declared speech, it is coded at the full fixed bit rate. Sometimes no bits are transmitted at all. The second algorithm, Comfort Noise Generation (CNG), is invoked at the receiver to reconstruct the main characteristics of the background noise. Obviously, the performance of the VAD is critical to the overall speech quality. The CNG scheme must be designed in such a way that the encoder and decoder stay synchronized, even if there are no bits transmitted during some interval.

The delay of a speech-coding system usually consists of three major components. Most low-bit-rate speech coders process a frame of speech data at a time. The speech parameters are updated and transmitted for every frame. In addition, to analyze the data properly, it is sometimes necessary to analyze data beyond the frame boundary. This is referred to as look-ahead. Hence, before the speech can be analyzed, it is necessary to buffer a frame's (plus look-ahead) worth of data. The resulting delay is referred to as algorithmic delay. This is the only delay component that cannot be reduced by changing the implementation. All other delay components depend on the implementation. The second major contribution comes from the time that it takes the encoder to analyze the speech and the decoder to reconstruct the speech. This is referred to as processing delay. It depends on the speed of the hardware used to implement the coder. The sum of the algorithmic and processing delays is called the one-way codec delay. The third component is the communication delay, which is the time it takes for an entire frame of data to be transmitted from the encoder to the decoder. The total of these three delays is the one-way system delay. Maximum values of 400 ms for the one-way system delay can be tolerated if there are no echoes. However, new testing methodologies revealed that, for the case of communication, it is preferable if the one-way delay is below 200 ms. In many applications, such as teleconferencing, it is necessary to bridge several callers so that each person can hear all the others. For speech coders, this means decoding each bit stream, summarizing the decoded signals and then re-encoding the sum signals. This process not only doubles the delay, but it also reduces the speech quality due to the multiple encodings.

Speech coders are often implemented on (or share) special purpose hardware, such as DSP chips. Their attributes can be described as computing speed in MIPS, RAM and ROM. Currently, speech coders requiring less than 15 MIPS are thought of as low complexity. Those requiring 30 MIPS or more are considered high complexity. From the system designer's point of view, more complexity results in higher costs and greater power usage. For portable applications, greater power usage means reduced time between battery recharges or using larger batteries, which means more expense and weight. Thus, complexity is an important factor.

In what follows, we give an overview of the standardization process for the three ITU coders. We start with a description of how the requirements are set. This is illustrated by the specifics for each of the speech-coder attributes: bit rate, delay, complexity and quality. In most standardization procedures, it is common to specify the requirements using the Terms of Reference (ToR). This document not only contains a schedule, but also specifies the performance requirements and objectives.

Bit Rate

For G.729, one of the ToR requirements was that the speech coder should operate at 8 Kb/s. This rate was selected in part because it fits the range of first-generation digital cellular standards, from 6.7 Kb/s in Japan to 7.95 Kb/s in the United States to 13 Kb/s in Europe. For G.723.1, the ToR requirement was that the speech coder should operate at lower than 9.6 Kb/s. As it turned out, all the coders tested ranged between 5.0 and 6.8 Kb/s. For the Digital Simultaneous Voice and Data (DSVD) coder, the ToR requirements for bit rate were derived from the amount of speech data that could be carried across a 14.4 Kb/s modem. The bit rates of the five candidate coders submitted for this standard were all near 8 Kb/s. None of the three coders had a silence compression scheme as part of the main body of the recommendation. Subsequent work created silence compression schemes for both G.723.1 and G.729, which are included as Annexes to each recommendation.

Delay

This is a major difference between G.723.1 and G.729. The ToR requirement for delay for G.729 was discussed for more than a year. Initially, it was a maximum one-way codec delay of 10 ms.

Later, the frame size was allowed to grow to 16 ms. G.729 has a 5 ms look-ahead. Assuming a 10 ms processing delay and a 10 ms transmission delay, the one-way system delay for G.729 is 35 ms. The principal application of G.723.1 is low-bit-rate videophones, which typically operate at 5 frames/s or fewer. This rate equates to a video frame period of 200 ms. The final version of G.723.1 has a look-ahead of 7.5 ms, making a one-way system delay of 97.5 ms. In deliberating on the delay requirements for a DSVD coder, SG14 was cognizant of the delay inherent in V.34 modems. These modems often have one-way delays greater than 35 ms.

Complexity

In formulating the requirements for G.729, the trade-off discussed involved delay and complexity. The ITU-R was concerned about creating a coder that would be too complex with too high a delay. Ultimately, they accepted a delay target that allowed a significant reduction in complexity compared with the G.728 coder. The MIPS are reduced to around 17. The amount of RAM required is 3,000 words, 50 % more than G.728. Much of this extra memory usage is due to the use of larger frames. G.723.1 is of lower complexity than G.729 (14.6 MIPS at 5.3 Kb/s and 16 MIPS at 6.3 Kb/s), and uses 2,200 words of RAM. The requirements for use of the DSVD coder were 10 MIPS, 2,000 words of RAM and 10,000 words of ROM.

Quality

Table 5.32 gives speech-quality performance requirements and objectives for G.729. It does not include requirements unrelated to speech quality, such as bit rate, delay and complexity, which are also discussed in the ToR. The first requirement is that, for error-free conditions, a single encoding should be rated not worse than 32 Kb/s (G.726). In separate testing of G.729, G.723.1 and the DSVD version of G.729, all three coders met this requirement. In Degradation Category Rating (DCR) tests, subjects seem to equate different with worse. As a result, G.729 received lower scores than G.726 for DCR testing. However, if Absolute Category Rating (ACR) tests were performed, the MOS of G.729 was never significantly worse than G.726 and was sometimes better. The testing of G.723.1 and G.726A was less extensive. The second requirement concerned speech quality with noisy channels. For 10^{-3} random bit-error rate, the speech quality should again be no worse than G.726 under similar conditions.

All three coders encode music signals, but the quality of the music is poor. The reason for this is that Linear Prediction Analysis by Synthesis (LPAS) coders rely on pitch prediction to achieve high coding efficiency. Most music signals lack a pitch structure, and all the coding burden has to be carried by the excitation and low-order LP.

The overall performance of the three coders was similar. It seemed that the G.723.1 and G.729A (DVD version of G.729) coders are slightly less robust for background noises and tandem conditions. Their performance for clean speech and general robustness is sufficient enough that the ITU sees fit to recommend them for use in simultaneous voice and data applications, such as low-bit-rate multimedia communications.

Table 5.32 Speech quality performance requirements and objectives for G.729 [5.213].

Parameter	Requirements	Objectives
Quality (without bit errors)	Not worse than 32 Kb/s G.726	N/A
Random bit errors $BER<10^{-3}$	Not worse than G.726	Equivalent to 32 Kb/s G.726
Detected frame erasures	No more than 0.5 MOS	N/A
Random and bursty 3%	Degradation from 32 Kb/s ADPCM without errors	
Undetected burst errors	N/A	For further study
Level dependency	Not worse than 32 Kb/s G.726	As low as possible
Talker dependency	Not worse than 32 Kb/s G.726	N/A
Capability to transmit music	N/A	No annoying effects generated
Tandeming capability for speech	Two asynchronous codings with a total distortion of <4 asynchronous 32 Kb/s G.726	3 asynchronous codings with a total distortion <4 Asynchronous 32 Kb/s G.726
Tandeming with other ITU standards	<4 asynchronous 32 Kb/s G.726	Synchronous tandeming property
Tandeming with regional DMR standards	For further study	N/A
Idle channel noise		
Weighted	For further study	Not worse than 32 Kb/s G.726
Single frequency	Not worse than 32 Kb/s G.726	N/A
Capability to transmit signaling/ information tones	DTMF, CCITT Nos.5, 6, 7, CCITT R2, Q.35, Q.23, V.25	Distortion as low as possible

5.9.5 Multimedia Multiplex and Synchronization Standards

ESs, such as audio, video, data, video frame synchronous control and indications signals, each of which may be internationally standardized or private, are multiplexed into a serial packet stream according to H.222.0. H.222.1/H.222.0 functions include multiplexing timebase recovery, media synchronization, jitter removal, buffer management, security and access control, subchannel signaling and trick modes which are mechanisms to support video recorder-like control functionality, for example, fast forward, rewind and so forth. Recommendation H.222.1 specifies elements and procedures from the generic H.222.0 for their use in ATM environments and also specifies codepoints and procedures for ITU-T-defined ESs [5.214]. H.222.1 allows the use of both the H.222.0 program stream and the H.222.0 transport stream. Only single-program transport streams are allowed [5.171]. A particular call may consist of multiple program streams or transport streams, each carried in separate ATM virtual channels and all referring to a common system time clock. Subchannel signaling is the process by which a subchannel for audio, video and other ESs is established and released between send and receive H.222.1 entities. Although H.222.1 specifies an unacknowledged signaling procedure using the Program Stream Map for H.222.0, Program Streams and Program Specific Information for H.222.0 Transport Streams, as well as an acknowledged signaling procedure, it offers improved call-phase synchronization and reliability.

ITU-T Recommendation H.221

This recommendation is entitled "Frame structure for a 64-1920 Kb/s channel in audiovisual teleservices" [5.215]. H.221 is the multiplex and bending protocol for H.320 terminals. Up to 30 ISDN B channels can be bundled together to form a superchannel with a bit rate of *n**64 Kb/s. The media channels for audio and video and the data information are multiplexed onto the supperchannel. For audio and video information, H.221 does not perform any error control, but relies completely on the error resilience of the media coding, which is possible because of ISDN's isochronous nature and its low error rates. The protocol offers only a bit-oriented, unprotected, point-to-point transport service.

ITU-T Recommendation H.223

This recommendation is entitled "Multiplexing protocol for low bit rate multimedia communication" [5.216]. Three different types of ALs are available that have different characteristics in terms of error probability and delay. Low-delay channels allow higher error rates, and reliable channels might have indefinitely long delays. AL1 and AL2 serve different duties; AL3 is designed for the use with coded video. Video data is encapsulated in small, variable-length packets (typically around 100 bytes, although larger packet sizes can be negotiated). A 16-bit Cyclic Redundancy Check (CRC) for each packet allows error detection. The packetization overhead for each packet is 1 to 3 bytes, plus error control information of AL3. AL3 includes an optional retransmission protocol, which sometimes allows the retransmission of a lost or corrupted packet. The retransmission of AL3 relies on the fast arrival of the confirmation messages, which give indications about correctly transmitted packets. These confirmation messages arrive at the sender of the original message with twice the one-way delay because a complete roundtrip of data and confirmation are necessary. The retransmission of the damaged packet after notification will incur a third one-way delay, resulting in a one-way delay three times .

ITU-T Recommendation H.225

Recommendation H.225.0 describes the means by which audio, video, data and control are associated, coded and packetized for transport between H.323 terminals on a nonguaranteed QoS LAN, or between H.323 terminals and H.323 gateways, which in turn may be connected to H.320, H.324 or H.310/H.321 terminals on NISDN, GSTN or BISDN, respectively. This gateway, terminal configuration and procedures are described in H.323, and H.225.0 covers protocols and message formats. The scope of H.225.0 communication is between H.323 terminals and H.323 gateways on the same LAN, using the same transport protocol, as shown in Figure 5.120. This LAN may be a single segment or ring or, if logical, could be an enterprise data network compromising multiple LANs bridged or routed to create one interconnected network. H.225 makes use of RTP/RTCP for media-stream packetization and synchronization for all underlying

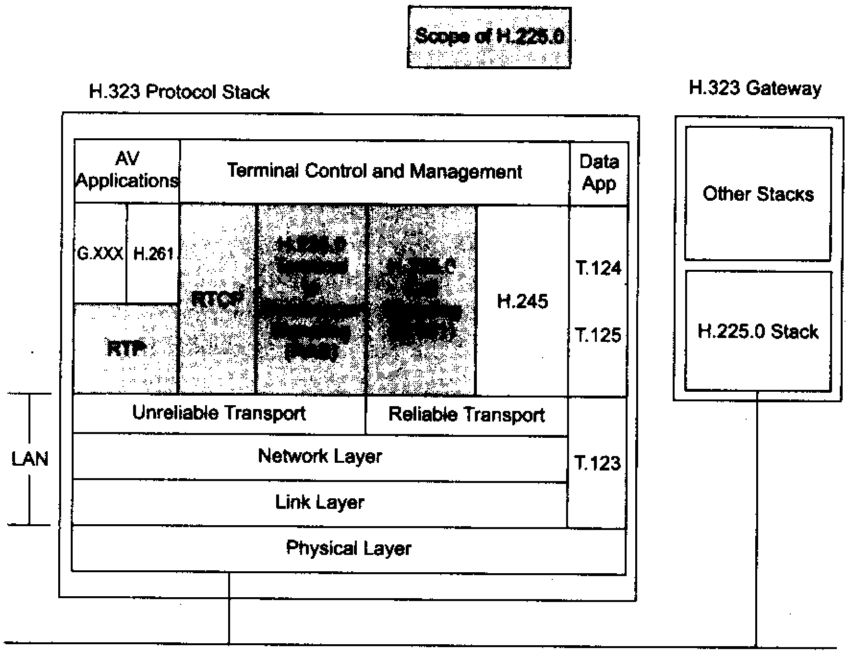


Figure 5.120 Scope of H.225.0 [5.171]. ©1997 IEEE.

LANs. H.225.0 assumes a call model where initial signaling on a non-RTP transport address is used for a call establishment and capability negotiation followed by the establishment of one or more RTP/RTCP connections. Thus, H.323/H.225.0 constitutes a profile for RTP/RTCP using the IETF terminology [5.171]. The general approach of H.225.0 is to provide a means of synchronizing packets that makes use of the underlying LAN/transport facilities. H.225.0 does not require all media and control to be mixed into a single stream, which is then packetized. The framing mechanisms of H.221 are not used for the following reasons:

- Not using H.221 allows each media to receive different error treatments as appropriate.
- H.221 is relatively sensitive to the loss of random groups of bits packetization allowing greater robustness in the LAN environment.

H.225 terminals send audio and video using RTP through unreliable channels to minimize delay. Error concealment or another recovery action must be applied to overcome lost packets; in general, audio/video packets are not transmitted because this would result in excessive delay. It is assumed that bit errors are detected in the lower layers, and errored packets are not sent up to H.225.0. Note that audio/video and call signaling/H.245 control are never sent on the same channel and do not share a common message structure. H.225.0 terminals are capable of sending and receiving audio and video on separate transport addresses using separate instances of RTP to allow for media-specific frame sequence numbers and separate QoS treatment for each medium.

However, an optional mode, where audio and video packets are mixed in a single frame and are sent to a single transport address, is for further study.

Common Control Protocol H.245

This recommendation defines messages and procedures for the exchange of control information between multimedia terminals. H.245 specifies terminal-to-terminal signaling to determine the coding and decoding capabilities of the remote terminal, following the establishment of the network connection, and to coordinate the assignment and release of terminal resources throughout the call. H.245 has been defined as a generic recommendation that is suitable for use in a range of multimedia terminal applications. Recommendations H.310 and H.323 described previously, it has also been adopted in H.324, the terminal for the GSTN and in V.70, which is for the multimedia terminal using modems. H.245 will also be used for the mobile terminal recommendation with working number H.324/M.

H.245 was structured by defining three main sections: syntax, semantics and procedures. In the first of these, ASN.1 notation was used to define generic syntax for messages [5.190]. The semantics section defines the meaning of syntax elements and provides syntactic constraints. Interaction between the different protocol entities is only through communication with the H.245 user. H.245 provides a number of different services to the H.245 user. Services may be applicable to a specific terminal recommendation. Some of these services are as follows:

- *Capability exchange*—Before multimedia communication begins, a terminal must be aware of the remote terminal's capabilities to receive and decode the multimedia signals. Terminal video capabilities may be one of, or a range of, H.261, H.262 or H.263 video. Audio capabilities may be one of the ITU-T G-series recommendations, such as G.711, G.722 or MPEG-1 or MPEG-2 Audio. Data capabilities include the T.120-series of recommendations. Audiovisual multiplexing capabilities are also recognized. Multiplexing capabilities include specification of the H.221 Program Stream or Transport Stream and ATM AL type and parameters.
- *Logical channels signaling procedures*—Following the capability exchange, but before the actual transmission of multimedia signals, the terminal coding and decoding resources are assigned using logical channel signaling. A logical channel number simply represents a specific channel in the system multiplex. Logical channel numbers are unique in each direction of transmission for a particular call.
- *Control and indication signals*—Messages are defined to carry the control and indication signals defined in H.221 and H.320 [5.215}. These are intended for various purposes, including maintenance loops, video/audio active/inactive signals, fast update request, source switching in multipoint applications and so forth.

It was expected from early in its development that H.245 would be a living document and would have additional features added to it from time to time, either to make it suitable for use in new terminal recommendations or simply to provide additional functionality to existing terminal recommendations. Thus, H.245 syntax has been designed to be extensible. This has been achieved by the use of extension markers in the syntax, as well as a protocol identifier field that indicates the version of H.245 that is being used. Extension markers allow syntax to be added so that earlier H.245 decoder implementations can skip the additional syntax, without understanding it, and continue to decode normally [5.218]. The protocol identifier field will be used to indicate more substantive changes to the recommendation, such as the definition of new procedures or the addition of new syntax that must be understood by the remote terminal. When the message containing the protocol identifier field is received and an earlier version of the H.245 is indicated, a terminal must restrict its use of messages and procedures to those of the earlier version. The set of procedures for each protocol entity in H.245 is specified using the Specification and Description Language (SDL) diagrams. The SDL diagrams define not only normal operations, but also actions to be taken in the event of exception conditions, such as the reception of unexpected messages.

5.10 IETF and Internet Standards

The IETF is an open international standardization body of network designers, operators, vendors and researchers focused on the development of Internet standard protocols for use on the Internet and intranets. The IETF is focused on the development of protocols used on IP-based networks. It consists of many working groups and is managed by the Internet Engineering Steering Group (IESG), Internet Architecture Board (IAB) and Internet Society (ISOC). The IETF is different from most standardization bodies in that it is a totally open community with no formal membership [5.219]. One of the strengths of the Internet is its global connectivity [5.220]. For this connectivity, it is essential that all the hosts on the Internet interoperate with one another and understand the common protocol at various layers. The Internet standardization process of IETF under the ISOC is the key to the success of GII over IP-based networks such as the Internet.

5.10.1 IETF Standardization Process

The Internet by definition is a complex mingle of networks based in the TCP/IP protocols. The whole structure of Internet management makes the prediction of its evolution complex. The Internet is largely self-developed by its own users, and it is too complex to predict changes about the relationships between the Internet and other technologies.

The ISOC was officially formed in January 1992. It was formed by a number of people with long-term involvement in the IETF, in order to provide an institutional home and financial support for the Internet standardization process [5.221]. Today, the ISOC is a nonprofit, nongovernmental, international, professional membership society with more than 100 organizations and 6,000 individual members in more than 100 countries. It provides leadership in addressing issues that confront the future of the Internet and is the organization home for the groups responsible for Internet infrastructure standards, including the IETF and the IAB.

ISOC aims to ensure the beneficial, open evolution of the global Internet and its related interworking technologies through leadership in standards, issues and education. The Society's individual and organizational members are bound by a common stake in maintaining the viability and global scaling of the Internet. The Society is governed by a board of trustees elected by its membership around the world, and the Board is responsible for approving appointments to the IAB from among the nominees submitted by an IETF nominating committee.

The IAB is the technical advisory group of the ISOC. It is chartered to provide oversight of the architecture of the Internet and its protocols and to serve in the context of the Internet standards process as a final appealing body. The IAB is responsible for approving appointments to the IESG from among the nominees submitted by the IETF nominating committee.

The IESG is responsible for technical management of IETF activities and the Internet standards' process. As part of the ISOC, it administers the Internet standards' process according to the established rules and procedures. The IESG is directly responsible for the actions associated with entry and movement along the standards track, including final approval of specifications as Internet standards. The IESG is composed of the IETF Area Directors (ADs) and the chairperson of the IETF, who also serves as the chairperson of the IESG. Representative of the increasingly larger span of the Internet is the fact that the IESG has established formal liaison with the ATM Forum and the ITU-T.

The IETF is a loosely self-organized group of people who make technical and other contributions to the engineering and evolution of the Internet and its technologies. It is open to any interested individual. The actual technical work of the Internet is mostly done inside the IETF. It is the principal body engaged in the development of new Internet standard specifications, although it is not itself a part of the ISOC. Much of the work is handled through mailing lists, because the IETF holds meetings only three times per year.

The IETF is composed of individual Working Groups (WGs) , which are organized by topics into several areas, each of which is coordinated by one or more ADs. These are the members of the IESG. Nominations to the IAB and the IESG are made by nominating committee members.

At present, the IETF is organized into the following areas:

- *Applications area*—Issues related to applications, other than security and networks
- *General area*—Internal IETF organization issues
- *Internet area*—Improvements on the TCP/IP protocols for increased usage and versatility
- *Operations and management area*—Concerned with management and operation control of the Internet
- *Routing area*—Internet routing protocol issues
- *Security area*—Support for security across all areas
- *Transport area*—Transport of different payloads across IP and the IP transport by other media
- *User services area*—A forum for people interested in all levels of user services and the quality of information available to users of the Internet

Each area is further divided into WGs, ranging from a couple to several dozen.

The Internet Assigned Number Authority (IANA) is the central coordinator for the assignment of unique parameter values for IPs. The IANA is chartered by the ISOC to act as the clearinghouse to assign and coordinate the use of numerous IP parameters. IANA functions as the top of the pyramid for Domain Name System (DNS) and Internet address assignment, establishing policies for these functions [5.221].

Request for Comments (RFC) started in 1969 as a series of notes about the Internet and then the Advanced Research Agency Network (ARPANET). The specification documents of the IP suite, as defined by the IETF and its steering group (IESG), are now formally published as RFCs [5.222]. There are several categories of RFCs, from informational to standard. Furthermore, there are different standardization degrees [5.223]. For a given RFC to reach the full Internet standard statue, stable implementations in multiple, independent, and interoperable versions are required.

The Internet standardization process is managed by IESG [5.224]. The existence of interoperable running implementations is the key requirement for advancement of the process. A document may take two paths in order to become an RFC. The first path is through the IETF. The IETF WGs develop documents that may be approved for publication as RFCs by ADs. Another path for an RFC to go through is for it to be individually submitted to the RFC editor.

The very first step, however, is for a document to become an Internet draft so that it may be distributed, read and commented on. These drafts, as well as all IETF documents, should be focused, handling few points of doubt. If required, a subject can be separated into different components and each treated separately in a different WG. If it is required, a WG can be created in a very fast way, after an initial session, in order to assess its interest. When created, a WG has a very well-defined charter and publishes its goals and milestones. There is no formal voting process inside the WG, and the results are achieved by consensus, often after discussing results of different demonstrations.

WGs are loosely co-ordinated through their ADs, besides mutual interests that their participants may share. Most of the work is being done by volunteers, and the IETF policy of accepting only working implementations for final standards makes the final approval of a particular WG extremely dependent on its real utility to the overall Internet community. Thus, Internet standards are always de facto standards although their widespread usage in the Internet may vary strongly. The whole structure is based on the active participation and interest of its volunteers, regardless of their individual motivations; it is an extremely fluid process when confronted with a more traditional telecommunications standards fora.

Although existing groups have published goals, these are usually short lived (about 1 to 2 years) and very well defined. The technical implementation to reach these goals is entirely dependent on the individual WG participants and on proven implementations of their proposals. Furthermore, WGs can be created and terminated according to their participants' ideas. Thus, it is complex to predict Internet evolution and future activities.

The whole structure of Internet management makes the prediction of its standards' evolution complex. Nevertheless, some of these areas are of more interest to AD and to issues relevant to cohesion and integration of architectures, and trends can be identified inside these areas. Clearly, the Transport area and the Operations and Management area have more impact on future network integration, although some of the work being progressed in the Internet and Routing and Security areas may be of some influence also.

5.10.2 Internet Network Architecture

Telecommunication networks and computer networks have been developed from different perspectives. Telecommunication networks have relied on circuit switching. The circuits have provided either a constant bandwidth or a constant data rate. When telephone circuit switches became so complicated that new services were required three or four years of switch software modification, the telecommunications industry developed a new architecture called Intelligent Network (IN) to facilitate the introduction of new services. IN defines interface-to-switch call-processing software so that central computers at a service control point can instruct the switch on how to handle a call.

Computer networks have adopted packet switching, which facilitates statistical multiplexing of burst data transmissions of different sources. Furthermore, computer networks have relied on the processing power of customer premises' equipment to control the network.

Although the global telephone network was originally designed to support one service (voice), the Internet architecture was designed to support a broad range of data communications services. In addition, IP was designed to operate across a wide range of network technologies. Like other network architectures, the Internet has a layered set of protocols. IP is simple, and it defines an addressing plan and a packet delivery service. An effort is made to deliver each packet, but there are no guarantees concerning the transmit time or even the reliability. Many protocols can run on top of IP [5.225]. The most common one is TCP, which provides a guaranteed delivery service. IP does not guarantee the delivery of packets, and TCP/IP is subject to unpredictable delays. As the Internet expands, new protocols such as RTP and RSVP are being developed. Another group of IP-based protocols supports multicasting, which increases the efficiency of network utilization for applications such as Internet radio and videoconferencing. Multimedia applications are being transferred from server to client, and people are experimenting with voice and video real-time connections across the Internet [5.6].

The Internet reference network architecture is composed of end nodes (hosts) linked by subnetworks as shown in Figure 5.121. All the hosts belonging to the same subnetwork exchange data directly. The crossing of subnetwork boundaries is enabled by means of intermediate nodes. Hosts and routers exchange data by means of the IP, which is the universal protocol used by the heterogeneous network components to offer a unified abstraction of the network service.

The IP is capable of offering a network service in which the information is packaged in data units named packets or datagrams. The network offers no assurance on the delivery of the packets to the intended recipient (best-effort service). Intermediate nodes decide where to route a packet addressed to a given destination on the basis of routing tables built by exchanging information with other intermediate nodes by means of custom protocols, such as Routing Informa-

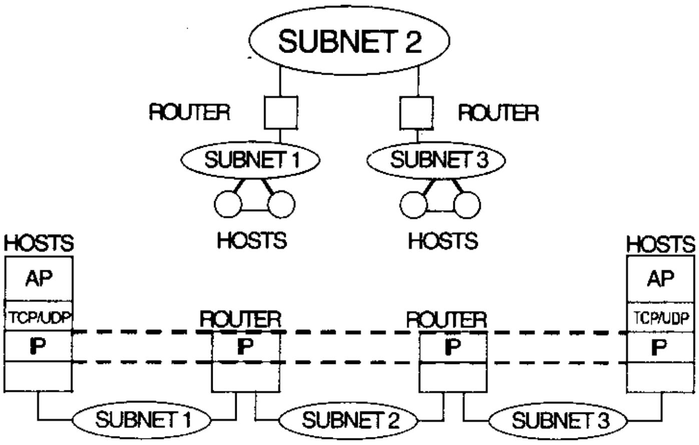


Figure 5.121 Internet network architecture [5.226]. ©1997 IEEE.

tion Protocol (RIP), OSPF and Border Gateway Protocol (BGP). The Internet Control Message Protocol (ICMP) supports IP by offering some basic control capabilities, such as sending reachable packets and asking an upstream packet source to slow down the packet transmission rate in the event of congestion. The end-to-end protocols add the capability to multiplex or demultiplex multiple flows of packets at the end nodes, and they may add reliability. In particular, UDP offers an unreliable service and adds a differentiation of packet flows within a host by means of port numbers [5.223]. The TCP offers a reliable sequenced delivery of byte streams on top of the datagram service offered by the IP. TCP is a connection-oriented protocol that enables connections with significance only at the end nodes. A windowing scheme is applied to enforce flow control, that is, to avoid the overrunning of slow receivers, as well as to allow the traffic source to adapt to network overload. In particular, the number of outstanding packets, that is, the number of packets that a source is entitled to transmit while waiting for an acknowledgement, is timed according to a probing of available bandwidth.

5.10.3 Internet Protocols

The IP has been the foundation of the Internet and virtually all multivendor private internetworks. A protocol, known as IPv6 (IP version 6) has been defined to ultimately replace IP [5.225]. The driving motivation for the adaptation of a new version of IP was the limitation imposed by the 32-bit address field in IPv4. Previous versions of IP (1 through 3) were successively defined and replaced to reach IPv4. Version 5 was assigned to the Stream Protocol, which runs parallel to IPv4 in some routers, which explains the use of the label "version 6."

Until recently, the Internet and most other TCP/IP networks have primarily provided support for rather simple distributed applications, such as file transfer, electronic mail, and remote access using Telnet. However, today, the Internet is increasingly becoming a multimedia, application-rich environment led by the huge popularity of the Web. At the same time, corporate networks have branched out from simple email and file transfer applications to complex client/server environments and intranets that mimic the applications available on the Internet. All of these developments have outstripped the capability of IP-based networks to supply needed functions and services. An interworked environment needs to support real-time traffic, flexible congestion control schemes and security features.

IP provides the functionality for interconnecting end systems across multiple networks. For this purpose, IP is implemented in each end system and in routers, which are devices that provide connection between networks. Higher-level data at a source and system are encapsulated in an IP Protocol Data Unit (PDU) for transmission. This PDU is then passed through one or more networks and connecting routers to reach the destination end system. The router must be able to cope with a variety of differences among networks, including addressing schemes, maximum packet sizes, interfaces and reliability.

The networks may use different schemes for assigning addresses to devices. For example, an IEEE802 LAN uses either 16-bit or 48-bit binary addresses for each attached device. An X.25 public packet-switching network uses 12-digit decimal addresses (encoded as 4 byte/digit for a 48-bit address). Some form of global network addressing must be provided, as well as a directory service.

Packets from one network may have to be broken into smaller pieces to be transmitted on another network, a process known as fragmentation. For example, Ethernet imposes a maximum packet size of 1,500 bytes. A maximum packet size of 1,000 bytes is common on X.25 networks. A packet that is transmitted on an Ethernet system and picked up by a router for retransmission on an X.25 network may have to segment the incoming packet into two smaller ones.

The hardware and software interfaces to previous networks differ. The concept of a router must be independent of these differences.

Various network services may provide anything from a reliable end-to-end virtual circuit to an unreliable service. The operation of the routers should not be defined on the assumption of network reliability. The operation of the router depends on an IP. IP must be implemented in all stations on all networks as well as on the routers.

Example 5.18 Consider the transfer of a block of data from station X to station Y as shown in Figure 5.122. The IP layer at C receives blocks of data to be sent to Y from TCP in X. The IP layer attaches a header that specifies the global Internet address of Y. That address is in two parts: network identifier and station identifier. Let us refer to this block as the IP datagram. IP recognizes that the destination (Y) is on another subnetwork. Therefore, the first step is to send the datagram to a router 1. To accomplish this, IP hands its data unit down to a Logical Link Control (LLC) with the appropriate addressing information. LLC creates an LLC PDU that is sent down to the Media Access Control (MAC) layer. The MAC layer constructs a MAC packet with a header that contains the address of router 1. Next, the packet travels through the LAN to router 1. The router removes the packet and LLC headers and trailers and analyzes the IP header to determine the ultimate destination of the data, in this case Y. The router must now make a routing decision. There are two possibilities:

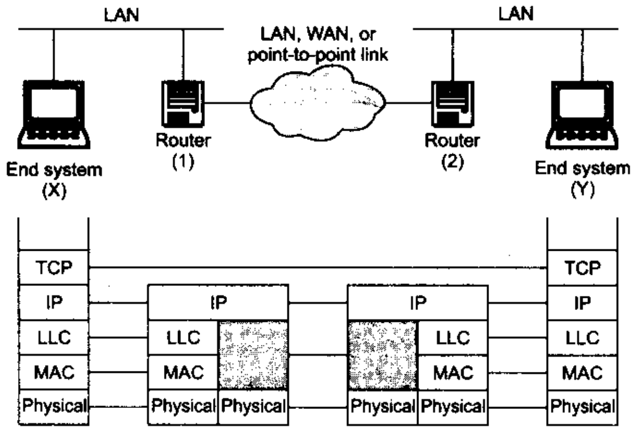


Figure 5.122 Protocol architecture including IP [5.225]. ©1996 IEEE.

- The destination station Y is connected directly to one of the subnetworks to which the router is attached.
- To reach the destination, one or more additional routers must be traversed. In this example, the datagram must be routed through router 2 before reaching the destination.

The IP datagram to router 2 passes to router 1 through the intermediate network. For this purpose, the protocols of that network are used. If the intermediate network is an X.25 network, the IP data unit is wrapped in an X.25 packet with appropriate addressing information to reach router 2. When this packet arrives at router 2, the packet header is stripped off. The router determines that this IP datagram is destined for Y, which is connected directly to a subnetwork to which the router is attached. The router therefore creates a packet with a destination address of Y and sends it out onto the LAN. Finally, the data arrives at Y, where the packet, LLC and Internet headers and trailers can be stripped off. This IP service is unreliable, that is, IP does not guarantee that all data will be delivered or that the delivered data will arrive in the proper order. It is the responsibility of the next higher layer, TCP in this case, to recover from any errors that occur. Because delivery is not guaranteed, there is no particular reliability requirement on any of the subnetwork types. Because the sequence of delivery is not guaranteed, successive datagrams can follow different paths through the Internet. This allows the protocol to react to congestion and failure on the Internet by changing routes.

Classical IP Stack

Figure 5.123 illustrates the classical IP stack, which includes end-user applications such as SMTP for the exchange of electronic mail messages and network-specific applications such as the DNS for the node-naming service [5.6]. Other user application-specific protocols are the remote terminal (Telnet), FTP and Network News Transfer Protocol (NNTP) for exchange of

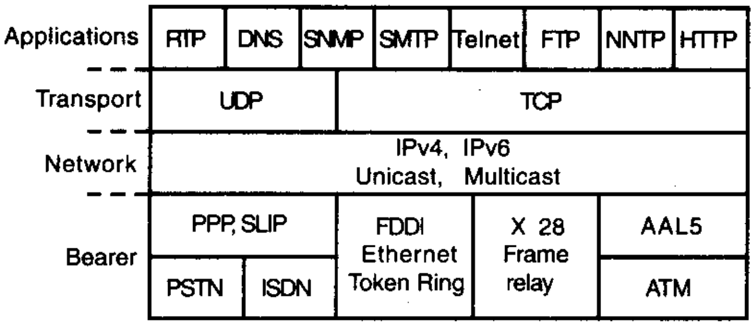


Figure 5.123 IPs [5.226]. ©1997 IEEE.

newsgroup information. The popular Web client/server applications based on the HTTP have been introduced, too. Among the key features of these applications is the capability to access a huge amount of multimedia information distributed worldwide in a transparent and user-friendly manner. The browsing of information is enabled by the user of the hypertext structure, where a document formatted using HTML contains links to other documents. The management of network resources on the Internet is carried out in the frame of the Simple Network Management Protocol (SNMP). That is an application layer protocol running over UDP for resilience designed to exchange management information among network nodes.

IP Version 6

With the shortcomings of the existing IP becoming increasingly evident, a new protocol known as IPv6 (IP version 6) has been defined. An IPv6 protocol data unit (known as a packet) has the general form shown in Figure 5.124. The only header that is required is referred to simply as the IPv6 header. This is of fixed size with a length of 40 octets, compared to 20 octets in the mandatory portion of the IPv4 header.

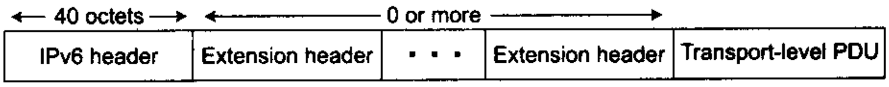


Figure 5.124 IPv6 PDU general form [5.225]. ©1996 IEEE.

The IPv6 header has a fixed length of 40 octets, consisting of the following fields as shown in Figure 5.125.

- *Version (4 bits)*—Indentifies IP version number; the value is 6.
- *Priority (4 bits)*—Indentifies the priority value.
- *Flow label (24 bits)*—May be used by a host to label those packets for which it is requesting special handling by routers within a network.

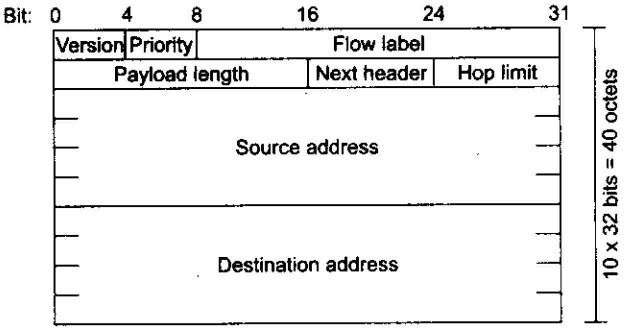


Figure 5.125 IPv6 header [5.225]. ©1996 IEEE.

- *Payload length (16 bits)*—Indentifies the length of the remainder of the IPv6 packet following the header, in octets. In other words, this is the total length of all the extension headers plus the transport-level PDU.
- *Next header (8 bits)*—Identifies the type of header immediately following the IPv6 header.
- *Hop limit (8 bits)*—Identifies the remaining number of allowable hops for this packet. The hop limit is set to some desired maximum value by the source and is decremented by 1 if the hop limit is decremented to zero.
- *Source address (128 bits)*—Identifies the address of the originator of the packet.
- *Destination address (128 bits)*—Identifies the address of the intended recipient of the packet. This may not in fact be intended as the ultimate destination if a routing header is present.

Priority Field

The 4-bit priority field enables a source to identify the desired transmit and delivery priority of each packet relative to other packets from the same source. The field enables the source to identify two separate priority-related characteristics of each packet. First, packets are classified as being part of the traffic for which the source is either providing congestion control or not. Second, packets are assigned one of eight levels or relative priority within each classification.

Congestion-controlled traffic refers to traffic for which the source backs off in response to congestion. An example is TCP. Let us consider what this means. If there is congestion on the network, TCP segments will take longer to arrive at the destination, and, hence, acknowledgments from the destination back to the source will take longer. As congestion increases, it becomes necessary for segments to be discarded en route. The discarding could be done by a router when that router experiences buffer overflow, or it could be done by an individual network allowing the route when a switching node within the network becomes congested. The nature of congestion-controlled traffic is that a variable amount of delay in the delivery of packets, and even for packets to arrive out of order, is acceptable.

Noncongestion-controlled traffic is traffic for which a constant data rate and a constant delivery delay, or at least a relatively smooth data rate and delivery delay, are desirable. Examples are real-time video and audio, for which it makes no sense to retransmit discarded packets. Further, it is important to maintain smooth delivery flow. Eight levels of priority are allocated for this type of traffic from the lowest priority, 8 (most willing to discard), to the highest priority, 15 (least willing to discard). In general, the criterion is how much the quality of the received traffic will deteriorate in the face of some dropped packets. For example, low-fidelity audio, such as a telephone voice conversation, would typically be assigned a high priority. The reason is that the loss of a few packets of audio is readily apparent as clicks and buzzes on the line. On the other hand, a high-fidelity video signal contains a fair amount of redundancy, and the loss of a few packets will probably not be noticeable. Therefore, this traffic is assigned a relatively low priority.

Flow Label

The IPv6 standard defines a flow as a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers. A flow is uniquely identified by the combination of the source address and a nonzero 24-bit flow label. Thus, all packets that are to be part of the same flow are assigned the same flow label by the source. From the source point of view, a flow typically will be a sequence of packets that are generated from a single application instance at the source and that have the same transfer service requirements. A flow may comprise a single TCP connection or even multiple TCP connections. An example of the use of multiple TCP connections is file transfer application, which could have one control connection and multiple data connections. A single application may generate a single flow or multiple flows. An example of the use of multiple flows is multimedia conferencing, which might have one flow for audio and one for graphics windows, each with different transfer requirements in terms of data rate, delay and delay variation.

From the router's point of view, a flow is a sequence of packets that shares attributes that affect how they are handled by the router. These include path, resource allocation, discard requirements, accounting and security attributes. The router may treat packets from different flows in a number of ways, including allocating different buffer sizes, giving different precedence terms of forwarding and requesting different qualities of service from subnetworks.

There is no special significance to any particular flow label. Instead, the special handling to be provided for a packet flow must be declared in some other way. For example, a source might negotiate or request special handling ahead of time from routers by means of a control protocol or at transmission time by information in one of the extension headers in the packet, such as the hop-by-hop options header.

IPv6 Addresses

IPv6 addresses are 128 bits in length. Addresses are assigned to individual interfaces on nodes, not to the nodes themselves. A single interface may have multiple unique unicast addresses. Any of the unicast addresses associated with a node's interface may be used to identify that node uniquely. IPv6 allows three types of addresses:

- *Unicast*—An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.
- *Anycast*—An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address.
- *Multicast*—An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

Unicast addresses. These addresses may be structured in a number of ways. The following have been identified: provider-based global, link-local, site-local, IPv4-compatible, IPv6 and loopback.

A provider-based global unicast address provides for global addressing across the entire universe of connected hosts. The address has five fields after the packet prefix:

- *Registry identifier*—Identifies the registration authoring that assigns the provider portion of the address.
- *Provider identifier*—Identifies a specific service provider that assigns the subscriber portion of the address.
- *Subscriber identifier*—Distinguishes among multiple subscribers attached to the provider portion of the address.
- *Subnet identifier*—Identifies a topologically connected group of nodes within the subscriber network.
- *Interface identifier*—Identifies a single node interface among the group of interfaces identified by the subnet prefix.

Link-local addresses are to be used for addressing on a single link or subnetwork. They cannot be integrated into the global addressing scheme.

Site-local addresses are designed for local use, but are formatted in such a way that they can be later integrated into the global address scheme. The advantage of such addresses is that they can be used immediately by an organization that expects to transition to the use of global addresses.

A key issue in deploying IPv6 is the transition from IPv4 to IPv6. It is not practical to replace all IPv4 routers on the Internet or a private Internet with IPv6 routers and to replace all IPv4 addresses with IPv6 addresses. Instead, there is a lengthy transition period when IPv6 and IPv4 must coexist. IPv4-compatible IPv6 addresses accommodate this coexistence period. Full coexistence can be maintained as long as all IPv6 nodes employ an Ipv4-compatible address. As general IPv6 addresses come into use, coexistence will be more difficult to maintain.

The unicast address 0: 0: 0: 0: 0: 0: 0: 1 is called the loopback address. It may be used by a node to send IPv6 packet to itself. Such packets are not sent outside a single node.

Anycast addresses. An anycast address enables a source to specify that it wants to contact any one node from a group of nodes using a single address. A packet with such an address will be routed to the nearest interface in the group, according to the router's measure of distance. Anycast addresses are allocated from the same address space as unicast addresses. Thus, members of an anycast group must be configured to recognize that address, and routers must be configured to be able to map an anycast address to a group of multicast interface addresses.

An example of the use of an anycast address is within a routing header to specify an intermediate address along a route. The anycast address could refer to the group of routers associated with a particular provider or particular subnet, thus dictating that the packet be routed through the Internet in the most efficient manner.

Multicast addresses. IPv6 includes the capability to address a predefined group of interfaces with a single multicast address. A packet with a multicast address is to be delivered to all members of the group. A multicast address consists of an 8-bit format prefix of all ones, a 4-bit flags field, a 4-bit scope field and a 112-bit group identifier.

Multicasting is a useful capability in a number of contexts. For example, it allows hosts and routers to send neighbor discovery messages only to those machines that are registered to receive them, removing the necessity for all other machines to examine and discard irrelevant packets. As another example, most LANs provide a natural broadcast capability. A multicast address can be assigned that has a scope of link-local with a group ID configured on all nodes on the LAN to be a subnet broadcast address.

Hop-by-Hop Options Header

This header carries optional information that, if present, must be examined by every router along the path. It consists of next header, header extension length and/or more options as shown in Figure 5.126.

Next header (8 bits) identifies the type of header immediately following this header. The length of header extension is 8 bits. A variable-length field consists of one or more option definitions. Each definition is in the form of three subfields: option type (8 bits), which identifies the option; length (8 bits), which specifies the length of the option data field in octets and option data, which is a variable-length specification of the option. It is actually the lowest-order five bits of the option type field that are used to specify a particular option. The higher-order two bits indicate the action to be taken by a node that does not recognize this option type. The conven-

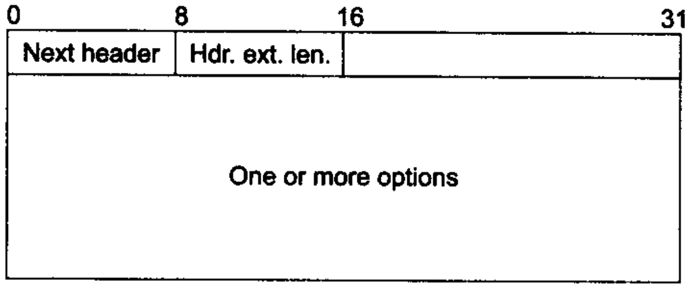


Figure 5.126 Hop-by-hop options header [5.225]. ©1996 IEEE.

tions for the option type field also apply to the destination options header. This header carries optional information that, if present, is examined only by the packet's destination mode.

Fragment Header

IPv6 fragmentation may be performed only by service nodes, not by routers along a packet delivery path. The path-discovery algorithm enables a node to learn the Maximum Transmission Unit (MTU) of the bottleneck subnetwork on the path. With this knowledge, the source node will fragment, as required, for each given destination address. Otherwise, the source must limit all packets to 576 octets, which is the minimum MTU that must be supported by each subnetwork. The fragment header (Figure 5.127) consists of the following:

- *Next header (8 bits)*—Identifies the type of header immediately following this header.
- *Reserved (8 bits)*—Remains for future use.
- *Fragment offset (13 bits)*—Indicates where in the original packet the payload of this fragment belongs. This implies that fragments must contain a data field that is a multiple of 64 bits long.
- *Res (2 bits)*—Remains reserved for future use.
- *M flag (1 bit)*—Identifies that 1=more fragments and 0=last fragment.
- *Identification (32 bits)*—Identifies the original packet uniquely. The identifier must be unique for the packets' source address and destination address for the time during which the packet will remain on the Internet.

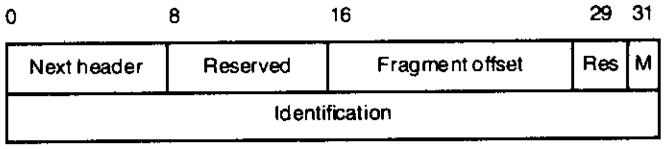


Figure 5.127 Fragment header [5.225]. ©1996 IEEE.

Routing Header

The routing header contains a list of one or more intermediate nodes to be visited on the way to a packet's destination. All routing headers starts with a 32-bit block consisting of four 8-bit fields, followed by routing data specific to a given routing type. A generic routing header is presented in Figure 5.128. The four 8-bit fields are the following:

- *Next header*—Identifies the type of header immediately following this header.
- *Header extension length*—Indentifies the length of this header in 64-bit units, not including the first 64 bits.
- *Routing type*—Identifies a particular routing header variant. If a router does not recognize the routing type value, it must discard the packet.
- *Segments left*—Identifies the number of explicitly listed intermediate nodes still to be visited before reaching the final destination.

0	8	16	31
Next header	Hdr ext len	Routing type	Segment left
Type-specific data			

Figure 5.128 Generic routing header [5.225]. ©1996 IEEE.

IPv6 requires an IPv6 node to reverse routes in a packet that it receives containing a routing header in order to return a packet to the sendeı. Figure 5.129 shows a configuration in which two hosts are connected by two providers, and the two providers are in turn connected by a wireless network. IPv6 has the ability to select a particular provider to maintain connections while mobile and to route packets to new addresses dynamically.

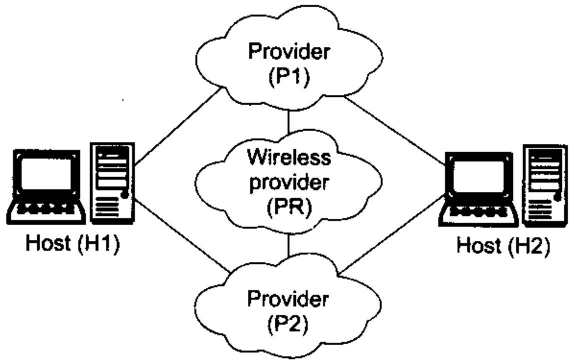


Figure 5.129 Examples of a routing configuration.

IPv6 Security

The Internet community has developed application-specific security mechanisms in a number application areas, including elecetronic mail, network management, Web access and others. However, users have some security concerns that cut across protocol layers. IP-level security encompasses two functional areas: authentication and privacy. The authentication mechanisms ensure that a received packet was in fact transmitted by the party identified as the source in the packet header. In addition, this mechanism ensures that the packet has not been altered in transit. The privacy facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties. In August 1995, the IETF published five security-related proposed standards that define a security capability at the Internet level. The documents provide the following [5.225]:

- *RFC1825*—Overview of security architecture
- *RFC1826*—Description of a packet authentication extension to IP

- *RFC1827*—Description of a packet encryption extension to IP
- *RFC1828*—Specific authentication mechanism
- *RFC1829*—Specific encryption mechanism

Support for these features is mandatory for IPv6 and optional for IPv4. In both cases, the security features are implemented as extension headers that follow the main IP header. The extension header for authentication is known as the authentication header, and the header for privacy is known as the Encapsulating Security Payload (ESP) header.

The authentication header provides support for data integrity and authentication of IP packets. The authentication header is given in Figure 5.130. It consists of the following fields [5.225]:

- *Next header (8 bits)*—Identifies the type of header immediately following this header.
- *Length (8 bits)*—Identifies the length of authentication data field in 32-bit words.
- *Reserved (16 bits)*—Remains for future use.
- *Security parameter index (32 bits)*—Identifies a security association.
- *Authentication data (variable)*—Identifies an integral number of 32-bit words.

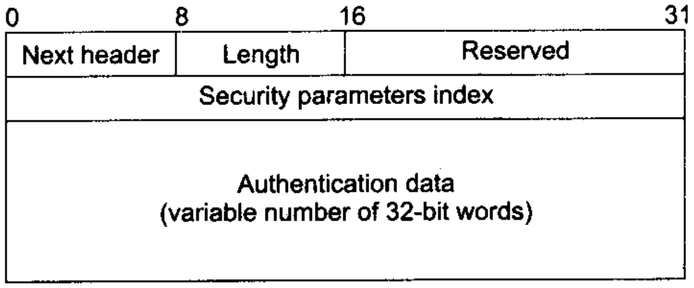
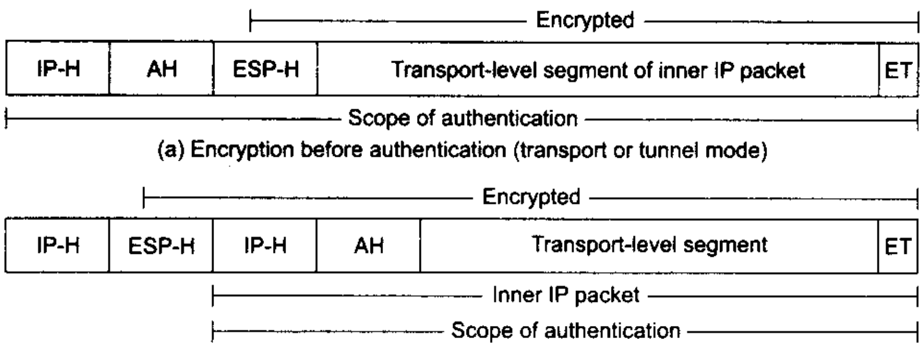


Figure 5.130 Authentication header [5.225]. ©1996 IEEE.

What the authentication data field contains will depend on the authentication algorithm specified. In any case, the authentication data is calculated across the entire IP packet, excluding any fields that may change in transit. Such fields are set to zero for purposes of calculation at both source and destination. The authentication calculation is performed prior to fragmentation at the source and after reassembly at the destination. Hence, fragmentation-related fields can be included in the calculation. The two IP security mechanisms can be combined in order to transmit an IP packet that has both privacy and authentication. Two approaches can be used based on the order in which the two services are applied: encryption before authentication and authentication before encryption.

Figure 5.131a illustrates the case of encryption before authentication. In this case, the entire transmitted IP packet is authenticated, including both encrypted and unencrypted parts. In this approach, the user first applies ESP to the data to be protected. Then, he presents the authen-



(a) Encryption before authentication (transport or tunnel mode)

(b) Authentication before encryption (tunnel mode)

IP-H = IP-based header plus extensions headers
ISP-H = Encapsulating security payload header
E-T = Encapsulating security payload trailing fields
AH = Authentication header

Figure 5.131 Privacy and authentication combination [5.225]. ©1996 IEEE.

tication header and the plain-text IP header(s). There are two subcases: transport-mode ESP and tunnel-mode ESP.

In transport-mode ESP, authentication applies to the entire IP packet delivered to the ultimate destination, but only the transport-layer segment is protected by the privacy mechanisms (encrypted).

In tunnel-mode ESP, authentication applies to the entire IP packet delivered to the other IP destination address. Authentication is performed at that destination. The entire inner IP packet is protected by the privacy mechanisms for delivery to the inner IP destination.

Figure 5.131b illustrates the case of authentication applied before encryption. This approach is appropriate only for tunnel-mode ESP. In this case, the authentication header is placed inside the inner IP packet. This inner packet is both authenticated and protected by the privacy mechanisms. The use of authentication prior to encryption might be preferable for several reasons. First, because the Authentication Header (AH) is protected by ESP, it is impossible for anyone to intercept the message and the AH without detection. Second, it may be desirable to store the authentication information with the message and the destination for later reference. It is more convenient to do this if the authentication information applies to the unencrypted message. Otherwise, the message would have to be re-encrypted to verify the authentication information.

5.10.4 Real-Time Multimedia Transmission across the Internet

The growth of the Internet and intranets has attracted a great deal of attention to the implementation and performance of networked multimedia services that involve the transport of real-time multimedia data streams over nonguaranteed QoS networks based on the IP. Continuing advances in computing technology, together with developments in signal coding and network protocols, have made transmission of real-time multimedia data across the Internet and intranets a viable and important application. An understanding of the Internet multimedia data transmission architecture is beneficial for developing signal-processing applications suitable for this fast growth area. Furthermore, effective design and use of the intermediate protocol layers of this architecture requires in-depth knowledge of both signal processing and networking. Based on their functionalities, the protocols directly related to real-time multimedia data transmission across the Internet can be classified into four categories:

- Signaling
- Session control
- Transport
- Network infrastructure

Signaling

Several protocols can be used for the higher-layer functions of signaling and session control. Signaling includes sending announcements about a multimedia session to prospective participants or inviting selected participants to join a session. In both cases, the details of the session, including the types of compression techniques used for audio and video signals, the number of audio channels, and so forth, may be a part of the signaling message. Generating and handling the responses of the receiver to a signaling message, for example, accept, join, reject, busy, forward and so forth, are handled by signaling protocols, too.

The ability of a receiver to decode the selected payload types and possible negotiations of the capabilities may be covered by signaling. Current protocols supporting signaling for multimedia sessions on the Internet include Session Description Protocol (SDP) for describing multimedia sessions [5.227], Session Announcement Protocol (SAP) [5.228] for announcing the described sessions and Session Initiation Protocol (SIP) for inviting users (human or machine) to participate in multimedia sessions [5.229]. HTTP and URL can be used to announce and describe sessions in a bulletin board format, which may also be considered as a part of a specific type of signaling.

Session Control

This defines the messages and procedures to control the delivery of the multimedia data during an established session. The Real-Time Streaming Protocol (RTSP) addresses tasks such as providing a means for choosing delivery channels and mechanisms, selecting a multimedia data segment for playback and controlling playback or recording properties using controls similar to the familiar ones on VCRs [5.230]. The H.323 standard defined by ITU-T standardizes both signaling and session control for tightly coupled multimedia communications sessions. A discussion of the relation between H.323 and other IPs can be found in [5.231].

Transport

The transport protocol has very tight relationships with the way that the multimedia payload types are organized and used. We discuss here the details of the transport layer based on the RTP [5.232, 5.233].

RTP is designed to deliver various kinds of real-time data across packet networks. It addresses the needs of real-time data transmission only and relies on other well-established network protocols for other communication services, such as routing, multiplexing and timing. RTP is based on the Application Level Framing (ALF) and Integrated Layer Processing (ILP) principles. They dictate using the properties of the payload in designing a data transmission system as much as possible [5.234]. For example, if we know that the payload is MPEG, we should design our packetization scheme based on slices because they are the smallest independently decodable data units for MPEG Video. This approach provides a much more suitable framework for MPEG transmission across networks with high packet loss rates. Also, we can identify and protect the critical information by repeating it frequently or sending it across a reliable channel. The services provided by the RTP are discussed in the following sections.

Payload type identification. The type of the payload contained in an RTP packet is indicated by an integer in a special field at the packet header. The receiver interprets the content of the packet based on this number. Certain common payload types have assigned payload type numbers. For other payloads, this association can be defined externally, for example, through signaling during the starting of a session or with session control protocols. The payload type identification service of the RTP together with the multiplexing services supported by the underlying transport protocol, such as UDP, provides the necessary infrastructure to multiplex a large variety of information effectively. Multicast transmission of several multimedia streams multiplexed together with any other type of information can easily be handled using these services.

RTP allows additional information to be added to its generic headers for each payload type. This information may be used to increase the packet-loss resiliency of the transmission. For example, each RTP packet carrying MPEG Video contains information about the picture type (intra, predictive and bidirectional), motion vector ranges, and so forth, copied from the latest picture header, increasing the decodability of individual packets.

Packet sequence numbering. Each RTP packet that belongs to a stream contains a 16-bit sequence number field that is incremented by one for each packet sent. The sequence numbers make packet loss detection possible because the lower protocol layers need not provide this information. Also, packets received out of order can be reordered using the sequence numbers. The initial sequence number is selected as a random number so that RTP packets do not cause known-plaintext attacks on the encryption that may be used at some later stage of their transmission.

Because the packets may be delivered out of order, receipt of a packet with an out-of-order sequence number does not necessarily imply packet loss. In most applications, a certain number of packets are buffered before starting the playback so that late or out-of-order packets can be used when they arrive. The buffer size depends on the network jitter, and, for interactive real-time applications, the buffer size is limited by the allowed delay.

Time stamping. Each RTP packet carries a 32-bit time stamp that reflects the sampling instant of the first byte in the payload portion of the packet. The interpretation and use of the time stamp are payload dependent. For example, for MPEG ES payloads, the time stamp represents the presentation time of the MPEG picture or audio frame, a portion of which is carried by the packet, based on a 90-KHz clock. It is the same for all packets that make up a picture or audio frame and, in a video stream with B-frames, it is not monotonically increasing. On the other hand, for fixed-rate audio (for example, PCM), the time stamp may reflect the sampling period. If blocks covering *n* audio samples are read from an input device, the time stamp would be increased by *n* for each such block, regardless of whether the block is transmitted in a packet or dropped as silent.

The time stamp, together with the information provided by the associated RTCP packets, is to be used for the following:

- Encoder/decoder clock matching
- Synchronization of several sources
- Measurement of packet-arrival jitter

Similar to the initial value of the sequence numbers, the initial value of the time stamp is random in order to make known-plaintext attacks on encryption difficult.

Source identification. The source of each RTP packet is identified by an integer called Synchronization SouRCe (SSRC) identifier included in the packet header. Each sender initially picks a random number for its SSRC. It is the sender's responsibility to detect and resolve collisions when more than one source picks the same number in the same session. The relation between several sources participating in a session, as well as their characterizing names, is established through RTCP.

The delivery-monitoring function of the RTP is carried out using the associated protocol, RTCP. RTCP is based on periodic transmission of control packets from all participants of a session to all other participants using the same distribution mechanism as the RTP data packets. RTCP's main functions are described in the following sections.

Feedback on the quality of distribution and timing. In an RTP session, each sender and each receiver send periodic reports to each session participant. Part of this report contains information on the quality of reception characterized as the following:

- Fraction of the lost RTP packets since the last report
- Cumulative number of packets lost since the beginning of reception
- Packet interarrival jitter
- Delay since receiving the last sender's report.

Sender and receiver reports contain enough information to determine these quantities at each participant's location. This feedback in reception quality is an integral part of RTP, and it is intended to be used for congestion and flow-control purposes as well as network performance input for the adaptive coding applications. Because RTP does not define an explicit flow control mechanism, an RTP application is capable of generating high traffic rates, causing network congestion. It is important to prevent this by analyzing the RTCP packets coming from the receivers so that other network applications are not disturbed.

Sending the feedback reports to all participants makes it possible to determine the extent of network problems. Additionally, a network management entity may monitor the network performance by observing these reports without actively participating in each session.

As for the timing, each sender's periodic RTCP packets contain 64-bit Network Time Protocol (NTP) time stamps, indicating the wall-clock (absolute) time when the RTCP packet was sent. This information can be used in combination with the timing information returned in reception reports from other receivers to measure roundtrip propagation to those receivers. Additionally, the sender's RTCP packet contains an RTP time stamp that corresponds to the same time as the NTP time stamp, but in the same units and with the same random offset as the RTP time stamps of the RTP data packets. This correspondence is to be used for intra- and intermedia synchronization for sources with synchronized NTP time stamps. A detailed discussion of the clock synchronization procedures can be found in Mills [5.235].

Participant identification. Special RTCP messages are used to establish a connection between the real identification of an RTP source, called by its canonical name (CName), and the current SSRC numbers used by it. CNames are very similar to email addresses following the **user name@host** syntax. Also, identification messages carry additional information about the participants, such as their names, email addresses, phone numbers, and so forth.

Control packet transmission scaled to the number of participants. As the number of the session participants increases, unregulated RTCP message traffic may consume significant bandwidth. In order to prevent this, RTCP scales itself by changing its message transmission interval based on the number of session participants. The suggested RTCP bandwidth is less than 5% of the bandwidth allocated for a session. Algorithms to achieve this are discussed in Schulzinne [5.232]

Minimal session control information. This optional functionality can be used for conveying simple session information, such as names of the participants, to everyone.

Network Infrastructure

All real-time multimedia data transmission applications across the Internet depend on both of the fundamental Internet transport protocols, UDP and TCP, for several functions, such as multiplexing, error control, flow control, and so forth. In turn, TCP and UDP depend on the basic IP for the network services support including network addressing [5.236].

The Point-to-Point Protocol (PPP), which defines a standardized method for sending datagrams across communications links such as telephone and ISDN lines, is an integral part of several real-time data transmission applications, such as Internet telephony. Several other protocols addressing specific requirements of real-time delivery are on their path to becoming standards. RSVP, which defines and implements QoS requirements, is important for multimedia data delivery across the Internet.

The lower layer (network infrastructure) protocols have a fundamental impact on the performance and usability of signal-coding techniques in networked applications. For example, if the network offers some service guarantees, such as delay bounds or guaranteed packet deliveries (no loss), signal-coding techniques with no error resilience can be used. If appropriate data flow control is done at the lower layers, application designers need not worry about network-buffer overflows due to short-term high-output data rates as in the case for I-frames in MPEG Video. In many cases, such additional services offered by the lower layers are not free, and a price-performance compromise may be obtained by using layered coding techniques. In this case, specialized services are needed only for transmitting a portion of the encoded data streams.

Multimedia Data for Network Use

It is well known that delivering real-time data encoded in any form across the Internet is possible. Nevertheless, real-time multimedia streams with the properties described in the following sections are more convenient for networked applications.

Natural breakpoints for packetization. Packetizing a stream that has natural breakpoints can be easier and more efficient. As an example, if a picture is JPEG-coded and is presented to a packetizer, the resulting packets contain arbitrary sections of the encoded data. If one of these packets is lost, it will be practically impossible to decode the remaining packets even if they are received, However, if the same JPEG coded picture contains special restart markers indicating starting of independently decodable blocks, a lost packet will not cause such a problem.

Adjustable packet sizes. Different technologies used as parts of the Internet have different frame (largest data unit) sizes. In order to carry a packet that is larger than the smallest frame size allowed on its path, called an MTU, the packet needs to be fragmented and reassembled. If the size of a packet can be changed based on the MTU, fragmentation can be avoided.

Well-defined high-priority information. If certain parts of a data stream are vital for decoding the rest of it, it is preferable to have these parts in easily identifiable and separable sections so that they can be transmitted more reliably.

Flexible rate control. An encoding scheme that has a rate that can easily be changed is useful in adapting its output to network conditions.

Ease of transcoding. The heterogeneity of bandwidths used for Internet access requires using different rates for the same multimedia material. Data streams that can easily be transcoded to change their bandwidth are definitely preferable.

Layered coding. Layered coding is beneficial for two purposes. The first one is to remove the need for transcoding by providing representations of the same multimedia source at different bit rates without noticeably increasing the overall bandwidth. The second benefit is to obtain a price/performance compromise by sending only a portion of a stream through channels.

Resilience to error propagation. Assuming that the packet losses will be unavoidable in the foreseeable future, techniques that prevent or reduce the propagation of data loss effects are preferable.

5.10.5 MPEG-4 Video Transport across the Internet

MPEG-4 is a standard designed for representation and delivery of multimedia information across a variety of transport protocols. It includes interactive scene management, visual and audio representations and systems functionality like multiplexing, synchronization and object-descriptor framework. The MPEG-4 Systems specification defines an architecture and tools to create audiovisual scenes from individual objects. The scene description and synchronization tools are the core of the Systems specification [5.64].

The scene description is encoded separately and is treated as another elementary bit stream. This separation allows for providing different QoSs for a scene description that has very low or no loss tolerance and media streams in the scene that are usually loss tolerant. The MPEG-4 scene description, also referred to as BIFS, is based on VRML and specifies the spatiotemporal composition of scenes. BIFS update commands can be used to create scenes that evolve over time. This architecture allows creation of complex scenes with potentially hundreds of objects. This calls for a high rate of establishment and release of numerous short-term transport channels with the appropriate QoS. DMIF is a general applications and transport delivery framework specified by MPEG-4 [5.43]. DMIF's main purpose is to hide the details of the transport network from the user as well as to ensure signaling and transport interoperability between end systems. In order to keep the user unaware of underlying transport details, MPEG-4 defined an interface between user-level applications and DMIF called DAI. The DAI provides the required functionality for realizing multimedia applications with QoS support. Although DMIF makes transport-independent application development possible, there is also a need to develop transport dependent mappings for MPEG-4 content to be able to use existing infrastructure and support applications that do not use DMIF. The IETF group is specifying payload formats and synchronization schemes for delivering MPEG-4 presentations using RTP [5.237].

Use of RTP

A number of Internet drafts describe RTP packetization schemes for MPEG-4 data [5.238]. Media-aware packetization (for example, video frames split at recoverable subframe boundaries) is a principle in RTP, so it is likely that several RTP schemes will be needed to suit the different kinds of media, audio, video, and so forth. No matter what packetization scheme is used, they must have a number of common characteristics. However, such characteristics depend on the fact that the RTP session contains a single ES or a FlexMux stream. An RTP session contains a single ES with the following characteristics:

- The RTP time stamp corresponds to the presentation time if the earliest access unit is within the packet.
- RTP packets have sequence numbers in transmission order. The payloads logically or physically have SL sequence numbers, which are in decoding order, for each ES.
- The MPEG-4 time scale (clock ticks per second) is the time-stamp resolution in the case of MPEG-4 systems and must be used as the RTP time scale.

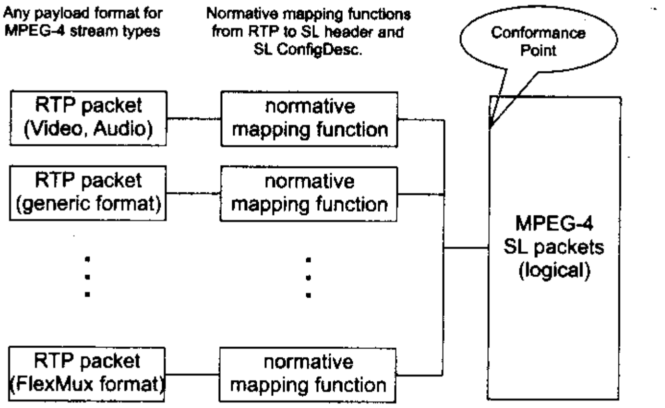

Any payload format for MPEG-4 stream types
Normative mapping functions from RTP to SL header and SL ConfigDesc.
Conformance Point
RTP packet (Video, Audio)
RTP packet (generic format)
RTP packet (FlexMux format)
normative mapping function
normative mapping function
normative mapping function
MPEG-4 SL packets (logical)

Figure 5.132 RTP-packet-to-SL-packet mapping [5.121]. ©2001 ISO/IEC.

- To achieve a base level of interoperability and to ensure that any MPEG-4 stream may be carried, all senders and receivers must implement a generic payload format.
- Streams should be synchronized using RTP techniques (notable RTCP sender reports). When the MPEG-4 Object Clock Reference (OCR) is used, it is logically mapped to the NTP time axis used in RTCP.
- The RTP packetization schemes may be used for MPEG-4 ES standing alone (for example, without MPEG-4 systems, including BIFS). Each RTP stream is passed through a mapping function, which is specific to the payload format used. This mapping function yields an SL packetization stream. RTP-packet-to-SL-packet mapping is shown in Figure 5.132.

There may be a choice of RTP payload formats for a given stream, for example, as an ES, an SL-packetized stream using FlexMux, and so on. The following is recommended:

- Terminals implementing a given subsystem (for example, video) accept at least an ES and the default SL packing of that stream, if they exist.
- Terminals implementing a given payload format accept any stream over that format for which they have a decoder, even if that packing is not normally the best packing.

Future versions of this specification will identify the single standard RTP packing format for each MPEG-4 stream type.

System Architecture

The MPEG-4 system developed is an end-to-end system consisting of an MEPG-4 server, the DMIF component for signaling and session management on an IP network and an MPEG-4 client for media playback and rendering. Figure 5.133 shows the components of the system.

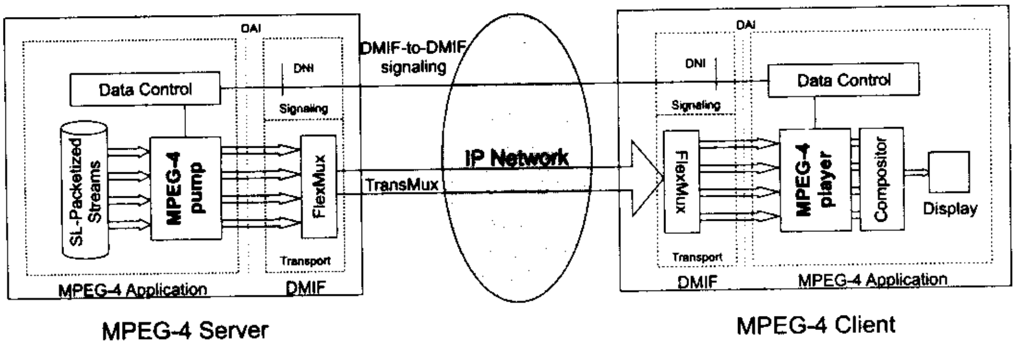


Figure 5.133 MPEG-4 System architecture [5.121]. ©2001 ISO/IEC.

The media data and the media composition data are transmitted to a client as separate streams, typically with different QoS requirements, in the same session. Furthermore, because the number of objects in a presentation can be quite large, the overhead required to manage the session is large. Interactivity makes this problem more complex because the resource required for a session will now depend on the user behavior, essentially when user interaction with objects changes the number of objects in the scene either by adding or deleting objects. The MPEG-4 server consists of an MPEG-4 pump, an object scheduler and a DMIF instance for IP networks. The server delivers SL packets to the DMIF layer, which multiplexes them in a single FlexMux stream and transmits them to the client [5.239]. The complexity of the player (that is, client) has grown as a result of the new features and functionality offered by MPEG-4. The player is responsible for compositing a scene from individual objects in addition to decoding and displaying the object.

A player consists of three logical components: a DMIF instance, ES decoders, and compositor. The DMIF instance is responsible for managing data access from a network or a file. A player typically contains several decoders, each handling a specific ES. ESs are audiovisual streams as well as streams that describe the composition, rendering and behavior of a presentation. Each object in a presentation is carried in a separate ES. Because MPEG-4 presentations can include media objects from several sources, potentially with different clock frequencies, there is an additional burden on the client to track multiple clocks. This is typically done using soft Phase-Locked Loops (PLLs). Because of this additional complexity, a player's performance depends on the complexity of the content. Intelligent resource management and usage are necessary to use the resources, such as memory, efficiently.

The capability to add and remove objects during presentations and to interact with objects differentiates object-based audiovisual presentation. DMIF also supports network independence by providing a DAI. DMIF is also responsible for negotiating the requested QoS for the applications. Typically, streams with the same QoS requirements are multiplexed into a single channel using FlexMux. These FlexMux packets are transported to the other end on the underlying transport network where they are demultiplexed by the peer DMIF entity and passed on to the application. DMIF is also responsible for communicating user interaction commands from Command Ds by means of DAI user-command primitives. These commands are transmitted across the DMIF-to-DMIF signaling channel.

A session is established before a server starts transmitting objects to a client. Session establishment is done by DMIF upon a request from the client. The MPEG-4 presentation to be delivered is selected by the client and communicated to the server as a part of session-establishment messages. MPEG-4 does not specify how a client selects a presentation. As the session establishment continues, the server sends the initial OD to the client. This initial OD contains pointers to the ESs that are part of the session. The client uses this information to request additional channels for the ESs. Each presentation contains at least two ESs, a scene-description stream and an OD stream. In addition to these two ESs, a presentation may include a clock-reference stream, command D stream, IPMP D stream or an OCI stream, in addition to the media stream. Intermedia synchronization is achieved using decoding and composition time stamps contained in SL packets. All ESs received by the client are time stamped. The time stamp indicates the time an access unit is processed by the client.

MPEG-4 Server

Figure 5.134 shows the components of the MPEG-4 server. The server delivers objects in a presentation as scheduled by the scheduler. The ESs, carrying media and media data for the presentation, are stored in the form of SL-packetized streams (SPS). The SL packet header contains the information, such as decoding and composition time stamps, clock references and packet repetition flags.

Scheduling and multiplexing of audiovisual objects in a presentation is a complex problem. Because of the different application domains, no solutions can be directly applied to the problem of scheduling audiovisual objects and also of scheduling in the presence of user interaction that might alter the presentation. The resource constraints that affect the transmission of access units are the channel capacity and buffer capacity at the receiving terminal [5.240].

The scheduler is also useful during the content creation process to determine if the presentation being designed can be scheduled for specific channel rates and client buffer capacity. It may not be possible to find a solution for a given set of resources, that is, the presentation cannot be scheduled with the given resources. In order to create a schedulable presentation, some constraints may be relaxed. In the case of scheduling objects, relaxing a constraint may involve increasing the buffer capacity, increasing the channel capacity, not scheduling some object instances, or removing some objects from a presentation.

It is also necessary to solve the problem online or, in some cases, to compute incremental schedules. Computing a schedule in real time is necessary to support interactive applications. When a user event adds a new object to the presentation, the resulting schedule has to be computed in real time to determine if the event can be supported. A scheduler may also be used offline to determine if an MPEG-4 presentation is suitable for a given channel and buffer capacities.

The MPEG-4 pump talks to a client using DMIF during session setup and delivers access units during the session. The server maintains a list of sessions established with clients and a list

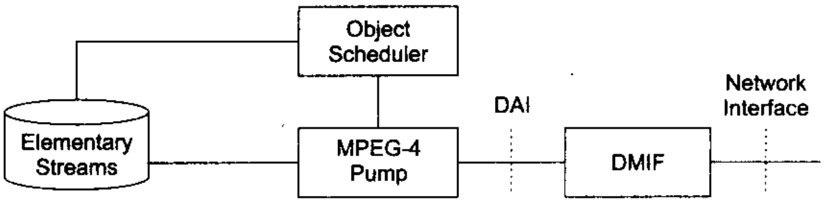


Figure 5.134 MPEG-4 Server components [5.121]. ©2001 ISO/IEC.

of channels for each session. Session identifiers and channel identifiers are used to identify channels in signaling between clients and a server. Network delays and data loss occur in the system if the content is not designed properly. A presentation created without the knowledge of target networks and clients could create long start-up delays, buffer overflows or underflows. This could cause distortion, gaps in media playback and problems with the synchronization of different media streams.

MPEG-4 Client

In order to achieve media synchronization, MPEG-4 defines a system decoder model, which abstracts the behavior of the receiving terminal in terms of synchronization, buffer management and timing for temporally accurate presentations of MPEG-4 scenes. These scenes, which are hierarchical groupings of media objects carrying synthetic or natural content, are composed using the scene description information coded using BIFS. BIFS is augmented by the OD framework. While the scene description declares the spatiotemporal relationship of the media objects, the ODs in the OD framework identify the resources for the ESs that carry these media objects and that associate the streams with the systems decoder model. BIFS and the OD framework form part of the basic building blocks for the architecture of the MPEG-4 client.

Figure 5.135 illustrates the architecture for the implementation of an MPEG-4 client used in the streaming application. The controller manages the flow of control and data information, the creation of buffers and decoders and attaches the transport channels established by the DMIF layer to the decoding buffers. The amount of buffer size to allocate for the decoding buffers is specified in decoder configuration Ds. The SL Manager manages a set of transport channels for receiving BIFS, OD and media streams by binding them to their respective decoding buffers. The SL Manager also provides functionality for forwarding client requests through the DAI to the DMIF process and receiving both control and data information from the server. The compositor encompasses a set of other components for rendering MPEG-4 presentations/scenes and for handling user events from the applications.

First, the client application requests a session establishment with the server and specifies the MPEG-4 presentation to be played. The SL Manager forwards this request to the DMIF layer, which handles the establishment of the session. In case of a successful session establishment, the server provides the initial OD information. This information is passed from the client DMIF through a specific DAI primitive to the SL manager. The initial OD is used for allocating buffers for the scene description (BIFS), OD and command descriptor streams. Then, the presen-

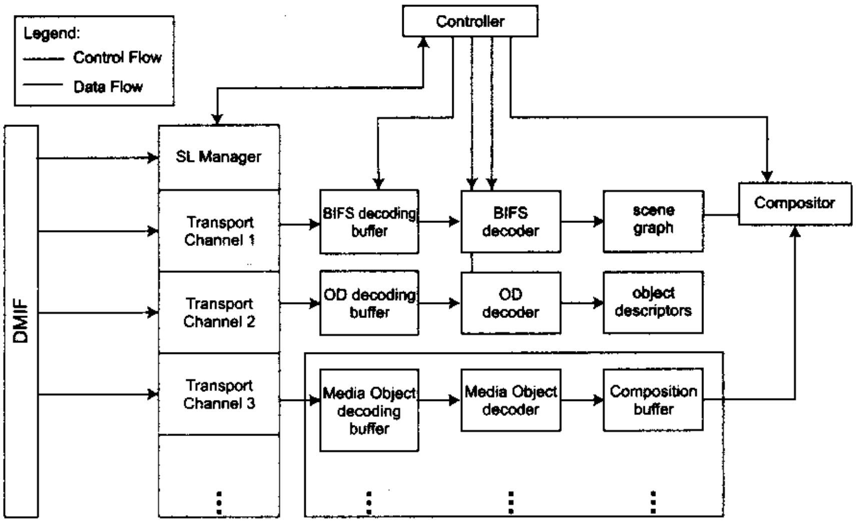


Figure 5.135 Architecture of an MPEG-4 client [5.121]. ©2001 ISO/IEC.

tation controller, which is a part of the compositor module requests, from the SL Manager the establishment of transport channels for the reception of the associated media streams. The SL Manager invokes the respective DAI commands and forwards this request to the DMIF layer. The DMIF establishes the transport channels taking into account the QoS characteristics of the media streams.

After the creation of the transport channels, the client may start receiving the MPEG-4 data. The BIFS data units are decoded to form a scene graph, which is a hierarchical ordering of nodes that describes the media object in a scene. The node attributes define the behavior and appearance of the object as well as their placement in space and time. The OD data units are decoded into a list of ODs used for associating the media streams (that is, ESs) with the media objects and for configuring the system to receive media streams.

The decoded objects in an MPEG-4 presentation are composed and presented by the compositor. The composition process involves positioning the media objects in a scene using scene description and handling the dynamic behavior of scenes.

5.11 Concluding Remarks

Standards play a major role in the multimedia communications because they provide interoperability between hardware and software provided by multiple vendors. However, production of standards is beset by the problem that the many industries having a stake in it have radically different approaches to standardization. The success of the MPEG, ITU-T and IETF standardization approaches is based on a number of concurrent elements.

MPEG is the leading standardization body in audiovisual representation technology. After the great success of the MPEG-1 and MPEG-2 standards, which opened the digital frontiers to audiovisual information and allowed the deployment of high performance services, MPEG-4 standard supports new ways of communication access and interaction with digital audiovisual data, and offers a common technical solution to various services. It also extends to layered coding (scalabilities), multiview (stereoscopic video), shape/texture/motion coding of objects, and animation. Its role extends to the Internet, Web TV, large databases (storage, retrieval and transmission), and mobile networks. The new standard MPEG-7 specifies a standardized description of various types of multimedia information. This description is associated with the content itself, to allow fast and efficient searching for multimedia that is of interest to users. The description can be attached to any kind of multimedia material, no matter what the format of the description is. Stored material that has this information attached to it can be indexed, searched and retrieved. The latest MPEG project MPEG-21 Multimedia Frameworks has been started with the goal to enable transparent and augmented use of multimedia resources across a wide range of networks and devices.

The ITU-T standardization process in multimedia communications deals with video and speech coding as well as multimedia multiplex and synchronization. In the past, most video compression and coding coding standards were developed with a specific application and networking infrastructure in mind. ITU-T Recommendation H.261 was optimized for use with interactive audiovisual communication equipment, e.g., a videophone, and in conjunction with the H.320 series of recommendations as multiplex and control protocols on top of ISDN. Consequently, the H.261 designers made various design choices that limit the applicability of H.261 to this particular environment. The original H.263 was developed for video compression rates below 64 Kbs per second. This was the first international standard for video compression which would permit video communications at such low rates. After H.263 was completed, it become apparent that there were incremental changes that could be made to H.263 that visibly improved its compression performance. It was thus decided in 1996 that a revision to H.263 would be created which incorporated these incremental improvements. ITU-T Recommendation H.263 Version 2 (H.263+) is the very first international standard in the area of video coding which is specifically designed to support the full range of both circuit-switched and packet-switched networks. H.263+ contains functionalities that improve the quality of video transmission in error-prone environments and nonguaranteed quality of service (QoS) networks. H.26L is an ongoing standard activity that is searching for advanced coding techniques that can be fundamentally different from H.263.

The IETF is focused on the development of protocols used on IP-based networks. The IETF is different from most standardization bodies in that it is a totally open community with no formal membership. One of the strengths of the Internet is its global connectivity. For this connectivity, it is essential that all the hosts in the Internet interoperate with one another, understanding the common protocol at various layers. The Internet standardization process of IETF under the Internet Society (ISOC) is the key to the success of multimedia communications over IP-based networks such as the Internet.