Multimedia Networks and Communication

Shashank Khanvilkar, Faisal Bashir, Dan Schonfeld, and Ashfaq Khokhar

University of Illinois at Chicago

INDEX

Paul Baran from the RAND Corporation first proposed the notion of a distributed communication network in 1964 [33] [42]. The aim of the proposal was to provide a communication network that could survive the impact of a nuclear war and employed a new approach to data communication based on packet switching. The Department of Defense (DoD) through the Advanced Research Projects Agency (ARPA) commissioned the ARPANET, in 1969. ARPANET was initially an experimental communication network that consisted of only four nodes: UCLA, UCSB, SRI, and the University of Utah. It's popularity grew very rapidly over the next two decades and by the end of 1989, there were over 100,000 nodes connecting research universities and government organizations around the world. This network later came to be known as the *'Internet'* and a layered protocol architecture (i.e. TCP/IP ref. Model) was adopted to facilitate services such as remote connection, file transfer, electronic mail, and news distribution over it. The proliferation of the Internet exploded over the past decade to over 10 million nodes since the release of the World Wide Web.

The current Internet infrastructure, however, behaves as a 'Best-Effort' delivery system. Simply put, it makes an honest attempt to deliver packets from a source to its destination, but provides no guarantees on the packet either being actually delivered and/or the time it would take to deliver it [22]. While this behavior is appropriate for textual data that requires correct delivery rather than timely delivery, it is not suitable for time-constraint multimedia data such as video and audio. Recently there has been a tremendous growth in demand for distributed multimedia applications over the Internet, which operate by exchanging 'multimedia' involving a myriad of media types. These applications have shown their value as powerful technologies that can enable remote sharing of resources or interactive work collaborations, thus saving both time and money. Typical applications of distributed multimedia systems include Internet based radio/television broadcast, video conferencing, video telephony, real-time interactive and collaborative work environments, video/audio on demand, multimedia mail, distant learning, etc.

The popularity of these applications has highlighted the limitations of the current best-effort Internet service model and viability of its associated networking protocol stack (i.e. TCP/IP) for the communication of multimedia data. The different media types exchanged by these applications have significantly different *traffic requirements* – such as *bandwidth, delay, jitter and reliability* – than the traditional textual data and demand different constraints or service guarantees from the underlying communication network to deliver an acceptable performance. In networking terminology, such performance guarantees are referred to as *Quality of Service* (QoS) guarantees, and can be provided only by suitable enhancements to the basic Internet Service model[22]. Circuit-switched networks, like the telephony system, Plain Old Telephone Service (POTS), have been designed from the ground-up to support such QoS guarantees. However, this approach suffers from many shortcomings like scalability, resource wastage, high complexity and high overhead [25]. Another approach, known as Asynchronous Transfer Mode (ATM), relies on cell switching to form virtual circuits that provide some of the QoS guarantees of traditional circuit-switched networks. Although ATM has become very popular as the backbone of high-bandwidth and local networks, it has not been widely accepted as a substitute for the protocol stack used on the Internet. Providing QoS in packet-switched Internet, without completely sacrificing the gain of statistical multiplexing, has been a major challenge of multimedia networking. In addition to the QoS guarantees, distributed multimedia applications also demand many *functional requirements* – such as support for *multicasting, security*, *session management,* and *mobility* – for effective operation, and these can be provided by introducing new protocols residing above the traditional protocol stack used on the Internet [48]. In this chapter, we discuss two popular protocol architectures (H.323 [44] [26] and SIP [17] [34]) that have been specifically designed to support distributed multimedia applications.

Apart from the Internet, cellular networks have also seen an unprecedented growth in its usage [13] and consequent demand for multimedia applications. The 2nd Generation (2G) cellular systems like GSM, IS-95, IS-136 or PDC, which offered circuit-switched voice services, are now evolving towards 3rd Generation (3G) systems that are capable of transmitting high-speed data, video and multimedia-traffic, to mobile users. IMT-2000 is composed of several 3G standards under development by the International Telecommunication Union (ITU) that will provide enhanced voice, data, and multimedia services over wireless networks. We discuss the layered QoS approach adopted by IMT-2000 to provide end-to-end QoS guarantees.

Section II starts with a general classification of media types from a networking/communication point of view. In this section, the reader is introduced to some common media types like text, audio, images, and video and a discussion about their *traffic* and *functional* requirements ensues. Section III discusses the inadequacy of the current best-effort Internet model to satisfy the multimedia *traffic requirements* and describes three enhanced architectures– Integrated Services [46], Differentiated Services [5], and Multi-Protocol Label Switching [30]– that have been proposed to overcome these shortcomings. Section IV presents some standard approaches to meet the functional requirements posed by multimedia traffic. Later in this section, we introduce the reader to two protocol architectures (H.323 and SIP) that have been introduced for the Internet protocol stack to satisfy these requirements. Section V describes current efforts to support multimedia traffic over the cellular/wireless networks and illustrates issues related to Inter-networking between wired and wireless networks.

II: Introduction to Multimedia

The term 'multimedia' refers to diverse classes of media employed to represent information. Multimedia traffic refers to the transmission of data representing diverse media over communication networks. Fig. 1 shows the diversity of the media classified into three groups: (i) text, (ii) visuals, and (iii) sound. As illustrated in Fig. 1, the symbolic textual material may include not only the traditional unformatted plain text, but also formatted text with numerous control characters, mathematical expressions, phonetic transcription of speech, music scores, and other symbolic representations such as hypertext. The visual material may include line drawings, maps, gray-scale or colored images and photographs, as well as animation, simulation, virtual reality objects, video- and tele-conferencing. The sound material may include telephone/broadcast-quality speech to represent voice, wideband audio for music reproduction, and recordings of sounds such as electrocardiograms or other biomedical signals. Other perceptory senses such as touch and smell, which can very well be considered as part of multimedia, are considered out of scope of this chapter.

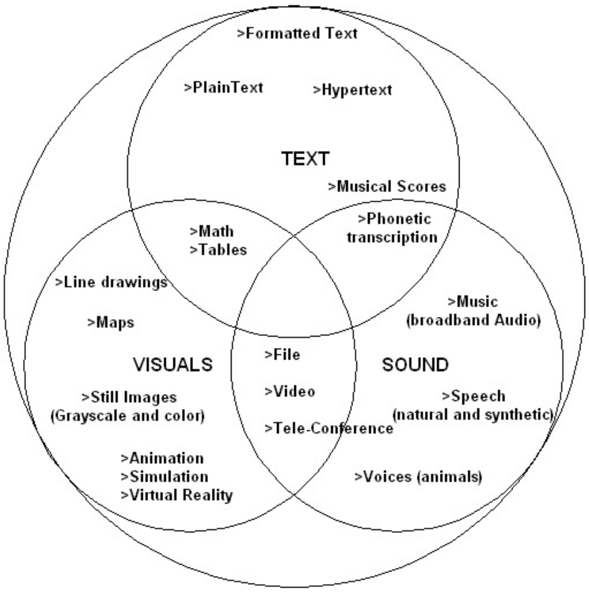


Fig. 1: Diversity of Multimedia Data Signals [20].

Text is inherently digital, while other media types like sound and visuals can be analog and are required to be converted into digital form using appropriate analog to digital conversion techniques. We assume that all media types that we consider here, have been suitably digitized and the reader is invited to read the book by Chapman and Chapman [8], which gives an excellent introduction to the principles and standards used to convert many such analog media to digital form. In this chapter, we focus on typical characteristics of different media types when transported over a network. In this regard, multimedia networking deals with the design of networks that can handle multiple media types with ease and deliver scalable performance.

2.1: Multimedia Classification

From a networking perspective, all media types can be classified as either Real-Time (RT) or Non Real-Time (NRT), as shown in Fig. 2. RT media types require either hard or soft bounds on the end-to-end packet delay/jitter, while NRT media types, like text and image files, do not have any strict delay constraints, but may have rigid constraints on error. There are basically two approaches to error-control [25]: (i) Error detection followed by Automatic Retransmission reQuest (ARQ) – request's retransmission of lost/damaged packets. – This approach is used by TCP (Transport Control Protocol), a transport layer protocol in the TCP/IP protocol stack, to provide reliable connection-oriented service. Applications that require an error-free delivery of NRT media, typically use TCP for transport. (ii) Forward Error Correction (FEC) – provide sufficient redundancy in packets so that errors can be corrected without the need for re-transmissions. – This approach can be used by UDP (User Datagram Protocol), another transport layer protocol in the TCP/IP protocol stack that provides connectionless un-reliable service. Applications that exchange error-tolerant media types (both RT and NRT) typically use UDP for transport as it eliminates time lost in re-transmissions. Leigh et. al. [24] have conducted experiments in the use of FEC along with UDP over a global high-bandwidth communication network, ***STARTAP***.

The RT media types are further classified as Discrete media (DM) or Continuous media (CM), depending on whether the data is transmitted in discrete quantum as a file or message, or continuously as a stream of messages with inter-message dependency. The real time discrete type of media has recently gained high popularity because of ubiquitous applications like MSN/Yahoo messengers (which are error-intolerant) and instant messaging services like stock quote updates (which are error tolerant).

The RT continuous type of media can further be classified as delay tolerant or delay intolerant. We cautiously use the term 'delay tolerant' to signify that such media type can tolerate higher amounts of delay than their delay intolerant counterparts, without significant performance degradation. Examples of RT, continuous, and delay-intolerant media are audio and video streams used in audio or video conferencing systems, and remote desktop applications. Streaming audio/video media, used in applications like Internet webcast, are examples of delay-tolerant media types. Their delay-dependency is significantly diminished by having an adaptive buffer at the receiver that downloads and stores a certain portion of the media stream before starting playout. The entire classification has been carefully illustrated in Fig. 2.

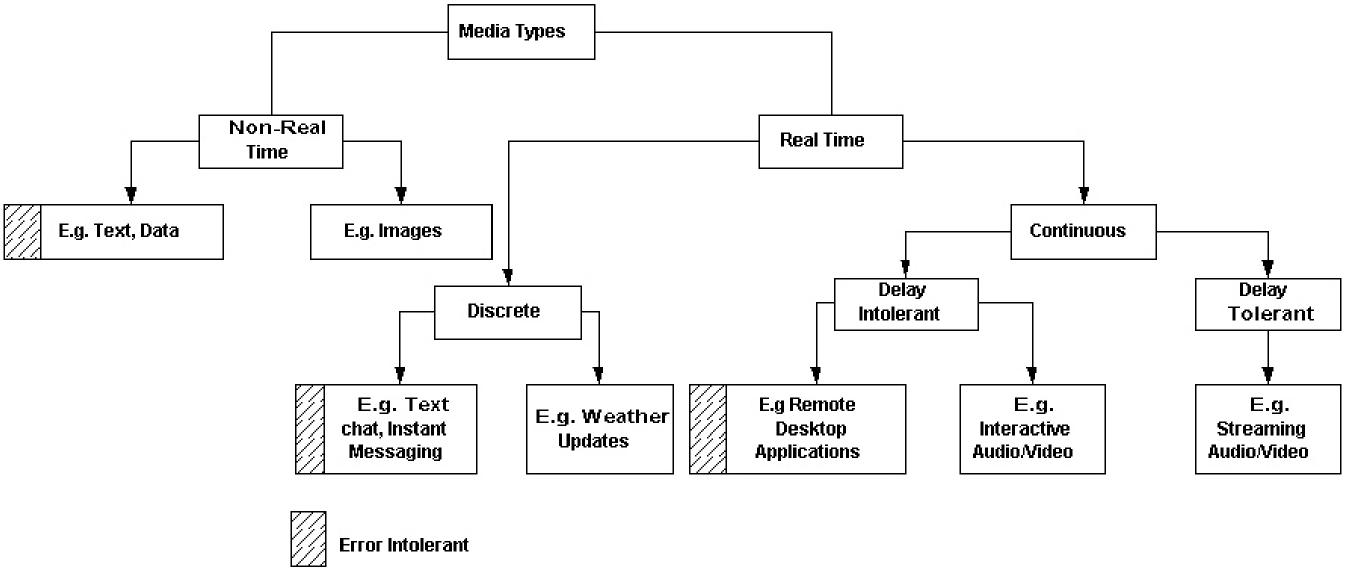


Fig. 2: Network oriented classification of Media Types.

We now discuss some common media types and their defining characteristics in terms of bandwidth usage, error requirements, and real-time nature.

2.2: Text

Text is the most popular of all the media types. It is distributed over the Internet in many forms including files or messages using different transfer protocols such as FTP (File Transfer Protocol: used to transfer binary and ASCII files over the Internet), HTTP (Hyper Text Transfer Protocol: used to transmit HTML pages) or SMTP (Simple Mail Transfer Protocol: Used for exchanging e-mails). Text is represented in binary as either 7-bit US-ASCII, 8-bit ISO-8859, 16-bit Unicode or 32-bit ISO 10646 character sets, depending on the language of choice and the country of origin. Bandwidth requirements of text media mainly depend on its size, which can be easily reduced using common compression schemes [32] as illustrated in Table 1.

The error characteristics of text media depend largely on the application under consideration. Some text applications, such as file transfer, require text communication to be completely loss/error free and therefore use TCP for transport. Other text applications such as instant messaging may tolerate some errors as well as losses and therefore can use UDP for transport.

Applications that use text as primary media, e.g., web browsing or e-mail do not have any real-time constraints, such as bounded delay or jitter. These applications are called *Elastic Applications*. However, applications like instant messaging do require some guarantees on the experienced delay.

Table 1: Text Compression schemes.

Compression Scheme	**Comments**
Shannon-Fano Coding	Uses variable length code words, i.e., symbols with higher probability of occurrence are represented by smaller codes-words.
Huffman Coding	Same as Shannon-Fano Coding.
LZW	LZW compression replaces strings of characters with single codes. It does not do any analysis of the incoming text. Instead, it just adds every new string of characters it sees to a table of strings. Compression occurs when a single code is output instead of a string of characters.
Unix Compress	Uses LZW with growing dictionary. Initially the dictionary contains 512 entries, and is subsequently doubled till it reaches the maximum value set by the user.

Overall, the text media has been around since the birth of the Internet and can be considered as the primary means of information exchange.

2.3: Audio

Audio media is sound/speech converted into digital form using sampling and quantization. Digitized audio media is transmitted as a stream of discrete packets over the network. The bandwidth requirements of digitized audio depend on its dynamic range and/or spectrum. For example, telephone-grade voice uses dynamic range reduction, using the logarithmic A-law (Europe) or μ-law (North America) capable of reducing the linear range of 12 bits to nonlinear range of 8 bits only. This reduces the throughput from 96 kbps to 64 kbps. A number of compression schemes [13] along with their bit rates, as illustrated in Table 2, are commonly used for audio media types:

Table 2: Audio Compression schemes.

Voice/Audio Codec	Used for	Bit Rate (Kbps)
Pulse code Modulation (G.711)	Narrowband speech (300 – 3300 Hz)	64
GSM	Narrowband speech (300 – 3300 Hz)	13
CS-ACELP (G.729)	Narrowband speech (300 – 3300 Hz)	8
G.723.3	Narrowband speech (300 – 3300 Hz)	6.4 and 5.3
Adaptive differential PCM (G.726)	Narrowband speech (300 – 3300 Hz)	32
SBC (G.722)	Wideband speech (50 – 7000 Hz)	48/56/64
MPEG layer III (MP3)	CD-quality music Wideband Audio (10 – 22Khz)	128 – 112 Kbps

The audio media type has loose requirements on packet loss/errors (or loss/error-tolerant), in the sense that it can tolerate up to 1 to 2 % packet loss/error without much degradation. Today, most multimedia applications that use audio, have inbuilt mechanisms to deal with the lost packets using advanced interpolation techniques.

The real-time requirements of audio strictly depend on the expected interactivity between the involved parties. Some applications like Internet-Telephony, which involves two-way communication, are highly interactive and require shorter response times. The audio media, in this case, requires strong bounds on end-to-end packet delay/jitter to be of acceptable/decipherable quality. Applications that use this media type are called *Real-Time Intolerant (RTI)* applications. In most RTI applications the end-to-end delay must be limited to ~200 msec to get an acceptable performance. Other applications like Internet webcast, which involves one-way communication, have relatively low interactivity. Interactivity, in this case, is limited to commands that allow the user to change radio channels (say), which can tolerate higher response times. Consequently, it requires weaker bounds on delay/jitter and the applications that use such kind of media are termed as *Real-Time Tolerant (RTT)* applications. "Streaming Audio" is also used to refer to this media type.

2.4: Graphics and Animation

This includes static media types like digital images and dynamic media types like flash presentations. An uncompressed, digitally encoded image consists of an array of pixels, with each pixel encoded in a number of bits to represent luminance and color. Compared to text or digital audio, digital images tend to be large in size. For example, a typical 4" x 6" digital image, with a spatial resolution of 480 x 640 pixels and color resolution of 24 bits, requires ~1MBytes. To transmit this image on a 56.6 Kbps line will take at least 2 minutes. If the image is compressed at the modest 10:1 compression ratio, the storage is reduced to ~100KB and transmission time drops to ~14 secs. Thus some form of compression schemes are always used that cash on the property of high spatial redundancy in digital images. Some popular compression schemes [32] are illustrated in Table 3. Most modern image compression schemes are progressive, which have important implications to transmission over the communication networks [20]. When such an image is received and decompressed, the receiver can display the image in a low-quality format and then improve the display as subsequent image information is received and decompressed. A user watching the image display on the screen can recognize most of the image features after only 5-10% of the information has been decompressed. Progressive compression can be achieved by: (i) encoding spatial frequency data progressively, (ii) using vector quantization that starts with a gray image and later adds colors to it, and (iii) using 'pyramid coding' which encodes images into layers, where early layers are of low resolution and the later layers progressively increase the resolution.

Images are error-tolerant and can sustain packet loss, provided the application used to render them knows how to handle lost packets. Moreover images, like text files, do not have any real-time constraints.

Table 3: Image Compression Schemes.

Compression scheme	**Comments**
Graphics Interchange Format (GIF)	Supports a maximum of 256 colors and is best used on images with sharply defined edges and large, flat areas of color like Text and line based drawings. GIF uses LZW (Lempel-Ziv-Welch) compression to make files small. This is a lossless compression scheme.
Portable Network Graphics (PNG)	Supports any number of colors and works best with almost any type of image. PNG uses the zlib compression scheme, compressing data in blocks dependant on the "filter" of choice (usually *adaptive*). This is a lossless compression scheme and does not support animation.
Joint Photographic Experts Group (JPEG)	Best suited for images with subtle and smooth color transitions such as photographs, grayscale/colored images. This compression standard is based on the Huffman and Run-Length encoding of the quantized Discrete Cosine Transform (DCT) coefficients of image blocks. JPEG is a "lossy" compression. Standard JPEG encoding does not allow interlacing, but the Progressive JPEG format does. Progressive JPEGs start out with large blocks of color that gradually become more detailed.
JPEG2000	Suitable for a wide range of images ranging from those produced by portable digital cameras through to advanced pre-press, medical imaging. JPEG 2000 is a new image coding system that uses state-of-the-art compression techniques based on wavelet technology that stores its information in a data stream, instead of blocks as in JPEG. This is a scalable lossy compression scheme.
JPEG-LS	Suitable for continuous-tone images. The standard is based on the LOCO-I algorithm (Low COmplexity LOssless COmpression for Images) developed by HP. This is a lossless/near-lossless compression standard.
Joint Bi-level Image Experts Group (JBIG)	Suitable for compressing black and white monochromatic images. Uses multiple arithmetic coding schemes to compress the image. This is a lossless type of compression.

2.5 Video

Video is a sequence of images/frames displayed at a certain rate, e.g., 24 or 30 frames/second. Digitized video, like digitized audio, is also transmitted as a stream of discrete packets over the network. The bandwidth requirements for digitized video depend on the spatial redundancy present within every frame, as well as the temporal redundancy present in consecutive frames. Both these redundancies can be exploited to achieve efficient compression of video data. Table 4, illustrates some common compression schemes [45] that are used in video.

The error- and real-time requirements of video media are similar to the audio media type. Hence, for the sake of brevity, we do not discuss them here.

Table 4: Video Compression Schemes.

Compression scheme	**Comments**
MPEG-I	Used to produce VCR NTSC (352 x 240) quality video compression to be stored on CD-ROM (CD-I and CD-Video format) using a data rate of 1.2 Mbps. Uses heavy down-sampling of images as well as limits image rate to 24-30 Hz to achieve this goal.
MPEG-II	More generic standard for a variety of audio-visual coding applications and supports error-resilience for broadcasting. Supports broadcast-quality video compression (DVB) and High Definition Television (HDTV). MPEG-2 supports four resolution levels: low (352 x 240), main (720 x 480), high-1440 (1440 x 1152), and high (1920 x 1080). The MPEG-2 compressed video data rates are in the range of 3-100 Mbps.
MPEG-IV	Supports low bandwidth video compression at data rate of 64 Kbps that can be transmitted over a single N-ISDN B channel. MPEG-4 is a genuine multimedia compression standard that supports audio and video as well as synthetic and animated images, text, graphics, texture, and speech synthesis.
H.261	Supports video communications over ISDN at data rates of px64 Kbps. It relies on intra and inter-frame coding where integer-pixel accuracy motion estimation is required for inter mode coding
H.263	The H.263 standard is aimed at video communications over POTS and wireless networks at very low data rates (as low as 18-64 Kbps). Improvements in this standard are due to the incorporation of several features such as half-pixel motion estimation, overlapping and variable blocks sizes, bi-directional temporal prediction, and improved variable-length coding options.

2.6: Multimedia Expectations from a Communication Network

In this section, we identify and analyze the requirements that a distributed multimedia application may enforce on the communication network. Due to the vastness of this field, we do not claim that this list is exhaustive, but we have tried to include all the important aspects (from our view point) that have significantly impacted the enhancements to the basic Internet architecture and its associated protocols. In

Sections III and IV, we further explore these aspects and give readers a sense of understanding of the efforts made to bring the Internet up to the challenges posed by such applications.

We divide these requirements into two categories [48], *traffic requirements* and *functional requirements.* The traffic requirements include limits on real-time parameters – such as delay and jitter –, bandwidth and reliability, and functional requirements include support for multimedia services such as multicasting, security, mobility and session management. The traffic requirements can be met only by enhancements to the basic Internet Architecture, while the functional requirements can be met by introducing newer protocols over the TCP/IP networking stack. The functional requirements are not an absolute necessity, in the sense that a distributed multimedia applications can still operate with high performance by incorporating the necessary functions into the application itself. However, they represent the most common functionality required amongst distributed multimedia applications, and it would only help to have standardized protocols operating over the networking protocol stack to satisfy them.

2.6.1: Real-time Characteristics (Limits on Delay and Jitter)

As discussed in Sections 2.1- 2.5, media types such as audio and video have real-time traffic requirements and the communication network must honor these requirements. For example, audio and video data must be played back continuously at the rate at which they are sampled. If the data does not arrive in time, the play back process will stop and human ears and eyes can easily pick up the artifact. In Internet telephony, human beings can tolerate a latency of ~200 msec. If the latency exceeds this limit, the voice will sound like a call routed over a long satellite link, which amounts to degradation in quality of the call. Thus real-time traffic enforces strict bounds on end-to-end packet delay – time taken by the packet to travel from the source to the destination – and jitter – variability in the inter-packet delay at the receiver. The performance of distributed multimedia applications improves with decrease in both these quantities.

2.6.2: Need for Higher Bandwidth

Multimedia applications require significantly higher bandwidths than conventional textual applications of the past. Moreover, media streams are transmitted using UDP that does not have any mechanism to control congestion. The communication network must be able to handle such high bandwidth requirements without being unfair to other conventional flows. Table 5 summarizes the bandwidth requirements of some common audio/video media types. We discussed several compression schemes that take advantage of spatial/temporal redundancy present in audio/video media, but the compressed media, still requires significantly higher bandwidth than what is typically required for text oriented services. Moreover, compression schemes cannot be expected to be used for all multimedia transmissions. There are two types of compression techniques: lossy and lossless. The lossy compression techniques eliminate redundant information from data and subsequently introduce distortion or noise in the original data. The lossless compression techniques do not loose any information and data received by the user is exactly identical to the original data. Lossy compression usually yields significantly higher compression ratios than lossless compression. However lossy compression might not be acceptable for all media types or applications (viz. medical images such as X-Ray images, Telemedicine, etc.) and it may be necessary to use either lossless compression or no compression at all.

Table 5: Sources of multimedia and their effective bandwidth requirements.

Audio Source	**Sampling Rate**	**Bits/Sample**	**Bit Rate**
Telephone Grade Voice (up to 3.4 KHz)	8000 samples/sec	12	96 Kbps
Wideband Speech (up to 7KHz)	1600 samples/sec	14	224 Kbps
Wideband Audio Two Channels (up to 20 KHz)	44.1 Ksamples/sec	16 per channel	1.412 Mbps for both channels
Image Source	Pixels	Bits/Pixel	Bit rate
Color Image	512 x 512	24	6.3 Mbps
CCIR TV	720 x 576 x 30	24	300 Mbps
HDTV	1280 x 720 x 60	24	1.327 Gbps

2.6.3: Error Requirements

As discussed in earlier sections, different media types have vastly different error requirements ranging from being completely error-intolerant to being somewhat error-tolerant depending on the application. An error is said to have occurred when a packet is either lost or damaged. Most error-tolerant multimedia applications use error concealment techniques to deal with lost or damaged packets, which predict lost information from correctly received packets.   Errors are handled using various FEC codes that can be used to detect and correct single or multiple errors.

The use of FEC implies that extra information has to be added to the packet stream in order to handle errors. However, if the communication path over which the packets are transmitted introduces additional errors beyond the level of degradation for which the FEC was designed, then some errors will remain undetected or may not be corrected and the performance will surely degrade. Thus, it is essential for a multimedia application to know the error characteristics of the communication network so that an adequate level of FEC is introduced in order to supplement the packet stream and protect against data loss or damage. As an example, wireless networks usually rely much more heavily on FEC than wired networks since its probability of packet loss is much higher. The minimization of packet retransmission achieved by using FEC can be too costly in wired networks that are characterized by very low probability of packet loss. The cost incurred is attributed to the additional bandwidth required for the representation of FEC information. The use of FEC is also critically dependent on the application. For instance, in real-time applications, some level of FEC is introduced for both wired and wireless communication networks since retransmissions are generally prohibited due to delay constraints.

2.6.4: Multicasting Support

Multicasting refers to single source of communication with simultaneous multiple receivers. Most popular distributed multimedia applications require multicasting. For example, multi-party audio/video conferencing is one of the most widely used services in Internet telephony. If multicasting is not naturally supported by the communication network (as was the case in some circuit-switched networks) then significant efforts need to be invested in building multimedia applications that support this functionality in an overlaid fashion, which often leads to in-efficient bandwidth utilization.

Multicasting is relatively easier to achieve for one-way communication than for two-way communication. For example, in the case of Internet Radio, multicasting can be achieved by creating a spanning tree consisting of the sender at the root and the receiver at the leaves and replicating packets over all links that reach the receivers. However, in the case of two-way communication like Internet telephony among multiple parties, there would be a need to have some form of audio mixing functionality that will mix the audios from all participants and only relay the correct information. Without this audio mixer, a two-way communication channel will need to be established between each participant in an all-to-all mesh fashion, which may amount to waste of bandwidth.

2.6.5: Session Management

The session management functionality includes:

- *Media Description:* This enables a distributed multimedia application to distribute session information such as media type (audio, video or data) used in the session, media encoding schemes (PCM, MPEG II), session start time, session stop time, IP addresses of the involved hosts, etc. It is often essential to describe the session before establishment, as most participants involved in the session will have different multimedia capabilities.
- *Session Announcement:* This allows participants to announce future sessions. For example, there are hundreds of Internet radio stations over the Internet, each web-casting different channels. Session announcement allows such radio stations to distribute information regarding their scheduled shows, so that a user finds it easier to tune-in to the preferred show.
- *Session Identification:* A multimedia session often consists of multiple media streams (including continuous media (audio, video) and discrete media (text, images)) that need to be separately identified. For example, the sender might choose to send the audio and video as two separate streams over the same network connection, which the receiver needs to decode synchronously. Another example is that the sender might put the audio and video streams together, but divide quality into a base layer and some enhancement layers, so that low-bandwidth receivers might be able to receive only the base layer, while high-bandwidth receivers might also receive the enhancement layers.
- *Session Control:* As said above, a multimedia session involves multiple media streams. The information contained in these data streams is often inter-related, and the multimedia communication network must guarantee to maintain such relationships as the streams are transmitted and presented to the user. This is called *Multimedia Synchronization* and can be achieved by putting timestamps in every media packet. Moreover many Internet multimedia users may want to control the playback of continuous media by pausing, playing-back, repositioning playback to a future or past point in time, visual fast-forwarding, visual rewinding playback etc [22]. This functionality is similar to what we have in a VCR, while watching a video or a CD player when listening to a CD.

2.6.6: Security

The security issue has been neglected in almost all discussions of multimedia communication. However, with the increasing use of online services and issues related to digital asset management, it is now apparent that security issues are quite significant. Security provides the following three aspects to multimedia data: Integrity (data cannot be changed in mid-flight), Authenticity (Data comes from the right source) and Encryption (Data cannot be deciphered by any third party). For example, public broadcasts require data integrity and data authenticity, while private communication requires data encryption. All the above aspects can be provided using different cryptographic techniques like secret key cryptography, public key cryptography and hash functions [19].

Another issue is that of protecting the intellectual copyrights for different media components. For example, consider digital movies that are distributed over the Internet using a pay-per-view service. It is possible for any entrepreneur to download such movies, and sell it illegally. Digital watermarking techniques [40], which embed extra information into multimedia data (such information is imperceptible to the normal user, as well as un-removable), can help to protect from such copyright violations.

2.6.7: Mobility Support

The advent of wireless and cellular networks has also enhanced multimedia applications with mobility. Cellular systems have a large coverage area and hence permit high mobility. Another emerging network is IEEE 802.11x wireless LAN [9], which can operate at speeds exceeding 54 Mbps. Wireless LAN's (WLAN) typically cover a smaller area and have limited mobility. However their main advantage is that they work in the ISM band (no licensing required, thus eliminating significant investments into license purchase), are relatively easy to set up and there is a vast availability of cheap WLAN products in the market that cost much less.

Mobility aspect has added another dimension of complexity to multimedia networks. It opens up questions on a host of complex issues like: routing to mobile terminals, maintaining the QoS when the host is in motion, inter-networking between wireless and wired networks etc.

III: Best-effort Internet Support for Distributed Multimedia *Traffic Requirements*

In this section, we further analyze why the current Internet, having the best-effort delivery model, is inadequate in supporting *traffic requirements* of multimedia traffic streams (as discussed in § 2.6.1-2.6.3) and justify the need for enhancements to this basic model. We also point out, the research approaches that have been adopted to make best effort Internet more accommodating to Real-time Multimedia traffic. To preserve the brevity of this chapter, we do not discuss every such approach at length but provide appropriate references for interested readers.

3.1: Best-effort Internet Support for Real-time Traffic

Real-time traffic requires strong bounds on packet delay and jitter and the current Internet cannot provide any such bounds. Packet delay and jitter effects are contributed at different stages of packet transmission and different techniques are used to reduce them. An analysis of these different components is essential in understanding the overall cause, since almost all enhancements to the current Internet architecture aim to reduce one of these components. Below we explain each of these components in more detail [4].

3.1.1: Packet Processing Delay

This is a constant amount of delay faced at both the source and the destination. At the source this delay might include the time taken to convert analog data to digital form and packetize it through different layers of protocols, till it is handed over to the physical layer for transmission. We can define similar packet processing delay at the receiver in the reverse direction. Usually this delay is the characteristic of the Operating System (OS) and the multimedia application under consideration. For a lightly loaded system this delay can be considered as negligible, however with increasing load this delay can become significant. However, it is independent of the Internet model (whether best-effort or any other enhanced version) and any reductions in this delay would imply software enhancements to Operating System (OS) kernel – such OS's are called Multimedia Operating system's [38] that provide enhanced process-, resource-, file- and memory-management techniques, with real-time scheduling – and the application.

3.1.2: Packet Transmission Delay

This is the time taken by the physical layer at the source to transmit the packets over the link. This delay depends on multiple factors including the following:

- *Number of active sessions*: The physical layer processes the packets in the FIFO order. Hence if there are multiple active sessions, this delay becomes quite significant especially if the OS does not support real-time scheduling algorithms to support multimedia traffic.
- *Transmission capacity of the link*: Increasing the transmission capacity reduces the transmission delay. For example, upgrading from the 10 Mbps Ethernet to 100 Mbps fast Ethernet will ideally reduce the transmission delay by a factor of 10.
- *Medium Access Control (MAC) access delay:* If the transmission link is shared, a suitable MAC protocol must be used for accessing the link [49]. The choice of MAC protocol largely influences this delay. For example, if the transmission capacity is C bps, and the packet length is L bits, time taken to transmit is L/C, assuming a dedicated link. However if the MAC protocol uses Time Division Multiple Access (TDMA), with m slots (say), this delay becomes mL/C, which is 'm' times larger than the earlier case. The widespread Ethernet networks cannot provide any firm guarantees on this access delay (and hence the overall QoS), due to the indeterminism of the carrier sense multiple access/collision detection (CSMA/CD) approach towards sharing of network capacity [48]. The reason for this is that the collisions, which occur in the bus-based Ethernet if two stations start sending data on the shared line at the same time, lead to delayed service time. Fast Ethernet exploits the same configuration as 10 Mbps Ethernet and increases the bandwidth with the use of new hardware in hubs and end stations to 100 Mbps, but provides no QoS guarantees. Isochronous Ethernet (integrated voice data LAN, IEEE 802.9) and demand priority Ethernet (100Base-VG, AnyLAN, IEEE 802.12) can provide QoS, yet their market potential remains questionable.
- *Context switch in the OS:* Sending or receiving a packet involves context switch in the OS, which takes a finite time. Hence there exists a theoretical maximum at which computer can transmit packets. For a 10 Mbps LAN, this delay might seem insignificant, however, for gigabit networks, this delay becomes quite significant. Again reduction in this delay will require enhancements to the device drivers and increasing the operating speed of the computer.

3.1.3: Propagation Delay

It is defined as the flight time of packets over the transmission link and is limited by the speed of light. For example, if the source and destination are in the same building at the distance of 200m, the propagation delay will be ~1 microsecond. However, if they are located in different countries at a distance of 20,000 Km, the delay is in order of 0.1 second. The above values represent the physical limits and cannot be reduced. This has major implications for interactive multimedia applications that require the response time to be less than ~200 msec. Thus if the one-way propagation delay is greater than this value, then no enhancements can improve the quality of the interactive sessions, and the user will have to settle for a less responsive system.

3.1.4 Routing and Queuing delay

This is the only delay component that we can reduce (or control) by introducing newer enhanced Internet architecture models. In the best-effort Internet, every packet is treated equally, regardless of whether it is a real-time packet or a non-real-time packet. All intermediate routers make independent routing decisions for every incoming packet. Thus a router can be ideally considered as an M/M/1 queuing system. When packets arrive at a queue, they have to wait for a random amount of time before they can be serviced, which depends on the current load on the router. This adds up to the queuing delay. The routing and queuing delay is random and hence the major contributor to jitter in the traffic streams. Sometimes when the queuing delay becomes large, the sender application times out and resends the packet. This can lead to an avalanche effect that leads to congestion and thus increase in queuing delays. Different techniques have been adopted to reduce precisely this delay component and thus have given rise to newer Internet Service models. For example, in the simplest case, if there is a dedicated virtual circuit connection (with dedicated resources in the form of buffers and bandwidth) from the source to the destination, then this delay will be negligible. The Integrated Services model (Intserv) and Multi-Protocol Label Switching (MPLS) follow this approach. Another option is to use a combination of traffic policing, admission control and sophisticated queuing techniques like priority queuing, weighted fair queuing etc. to provide a firm upper bound on delay and jitter. The Differentiated Services model (Diffserv) follows this approach. Later we will discuss, in some more detail, the principles that need to be followed to reduce this delay component.

3.2: High Bandwidth Requirements

Multimedia traffic streams have high bandwidth requirements (refer Table 5). The best effort Internet model neither provides any mechanism for applications to reserve network resources to meet such high bandwidth requirements nor prevents anyone from sending data at such high rates. Uncontrolled transmissions at such high rates can cause heavy congestion in the network leading to a congestion collapse that can completely halt the Internet. There is no mechanism in the best-effort Internet to prevent this from happening (except using a brute force technique of disconnecting the source of such congestion). It is left to the discretion of the application to dynamically adapt to network congestions. Elastic applications, that use TCP, utilize a closed-loop feedback mechanism (built into TCP) to prevent congestion (This method of congestion control is called re-active congestion control). However, most multimedia applications use UDP for transmitting media streams, which does not have any mechanism to control congestion and have the capability to create a congestion collapse.

To remove the above shortcomings, the enhanced Internet service models, use admission control, bandwidth reservations and traffic policing mechanisms. The application must first get permission from some authority to send traffic at a given rate and with some given traffic characteristics. If the authority accepts admission, it will reserve appropriate resources (bandwidth and buffers) along the path for the application to send data at the requested rate. Traffic policing mechanisms is used to ensure that applications do not send at a rate higher than what was initially negotiated.

3.3: Error Characteristics

Multimedia streams require some guarantees on the error characteristics of the communication network and the best-effort Internet cannot provide such guarantees because the path that a packet follows from the source to the destination is not fixed and hence the network has no idea about the error characteristics of each individual segment. Thus the sender application has no knowledge of the error characteristics of the network and may end up using error correction/detection mechanism that may not be optimum.

For the newer Internet Service models, the sender application has to go through admission control. At this time, the sender can specify the maximum error that it can tolerate. If the network uses a QoS based routing algorithm, explained later in Section 3.4.7, and is unable to find a path that can satisfy this requirement, it will just reject the connection or make a counter offer to the sender specifying the error rate at which it is willing to accept the connection. In other words, the sender is made aware of the error characteristics of the network.

3.4: Proposed Service Models for the Internet

We now discuss several new architecture models – Integrated Services (Intserv), Differentiated Services (Diffserv) and Multi-protocol Label Switching (MPLS) – that have been proposed for the best-effort Internet to satisfy the traffic requirements of distributed multimedia applications. But, before delving into the discussion of these QoS service models proposed for the Internet, we would like to summarize some of the principles that are common to all of them and can also be expected to be seen in any future proposals.

3.4.1: Clearly defined Service Expectations and Traffic Descriptions

To enhance the current Internet to support service guarantees, it is necessary to define such service guarantees in clear mathematical terms. QoS quantifies the level of service that a distributed multimedia application expects from the communication network. In general, three QoS parameters are of prime interest: bandwidth, delay, and reliability.

Bandwidth, as the most prominent QoS parameter, specifies how much data (maximum or average) are to be transferred within the networked system [48]. In general, it is not sufficient to specify the rate only in terms of bits, as the QoS scheme shall be applicable to various networks as well as to general-purpose end systems. For example, in the context of protocol processing, issues such as buffer management, timer management, and the retrieval of control information play an important role. The costs of these operations are all related to the number of packets processed (and are mostly independent of the packet size), emphasizing the importance of a packet-oriented specification of bandwidth. Information about the packetization can be given by specifying the maximum and the average packet size and the packet rate.

Delay, as the second parameter, specifies the maximum delay observed by a data unit on an end-to-end transmission [12]. The delay encountered in transmitting the elements of a multimedia object stream can vary from one element to the next. This delay variance can take two forms, viz. delay jitter and delay skew. Jitter implies that in an object stream the actual presentation times of various objects shift with respect to their desired presentation times. The effect of jitter on an object stream is shown in Fig. 3.

In Fig. 3(a) each arrow represents the position of an object, which is equally spaced in time. In Fig. 3 (b) the dotted arrows represent the desired positions of the objects and the solid arrows represent their actual positions. It can be seen in Fig. 3 (b) that these objects are randomly displaced from their original positions. This effect is called jitter in the timing of the object stream. The effect of jitter on a video clip is a shaky picture.

Skew implies constantly increasing difference between the desired presentation times and the actual presentation times of streamed multimedia objects. This effect is shown in Fig. 3 (c). The effect of skew in the presentation times of consecutive frames in a video will be a slow (or fast) moving picture. Jitter can be removed only by buffering at the receiver side.

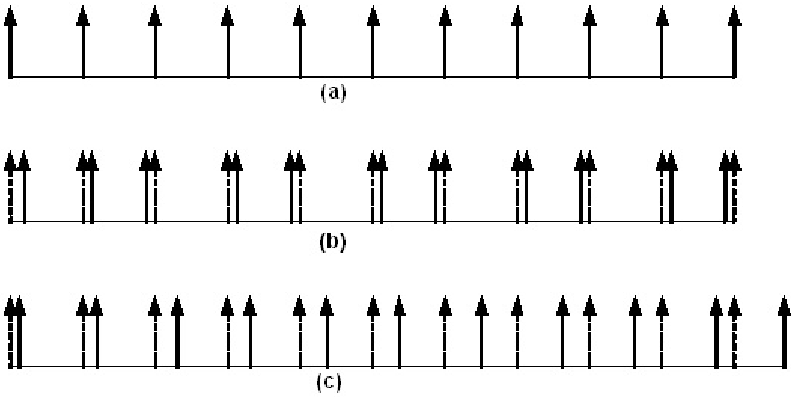


Fig. 3: (a) Original Multimedia object stream at regular intervals. (b) Effect of jitter. (c) Effect of delay skew [12].

Reliability pertains to the loss and corruption of data. Loss probability and the method for dealing with erroneous data can also be specified.

Also it becomes necessary for every source to mathematically describe the traffic characteristics of the traffic it will be sending. For example, every source can describe its traffic flow characteristics using a *traffic descriptor* that contains the peak rate, average rate and maximum burst size [22]. This can be specified in terms of leaky bucket parameters, like the bucket size *b*, and the token rate *r*. In this case, the maximum burst size will be equal to the size of the bucket i.e. *b*, peak rate will be '*r*T + *b'*, where 'T' is the time taken to empty the whole bucket, and average rate over time *t* is *rt* + *b*.

3.4.2: Admission Control

This is a *pro-active* form of congestion control (as opposed to reactive congestion control used in protocols like TCP) that ensures that demand for network resources never exceeds the supply. Preventing congestions from occurring reduces packet delay and loss, which improves real-time performance.

An admission control module (refer Fig. 4) takes as input the traffic descriptor and the QoS requirements of the flow, and outputs its decision of either accepting the flow at the requested QoS or rejecting it, if that QoS is not met [43]. For this it consults *Admission criteria* module, which are the rules by which an admission control scheme accepts or rejects a flow. Since the network resources are shared by all admitted flows, the decision to accept a new flow may affect the QoS commitments made to the admitted flows. Therefore, an admission control decision is usually made based on an estimation of the effect the new flow will have on other flows and the utilization target of the network.

Another useful component of Admission control is the *Measurement process* module. If we assume sources can characterize their traffic accurately using traffic descriptors, the admission control unit can simply use parameters in the traffic descriptors. However, it is observed that real-time traffic sources are very difficult to characterize and the leaky bucket parameters may only provide a very loose upper bound of the traffic rate. When the real traffic becomes bursty, the network utilization can get very low if admission control is solely based on the parameters provided at call setup time. Therefore, the admission control unit should monitor the network dynamics and use measurements such as instantaneous network load and packet delay to make its admission decisions.

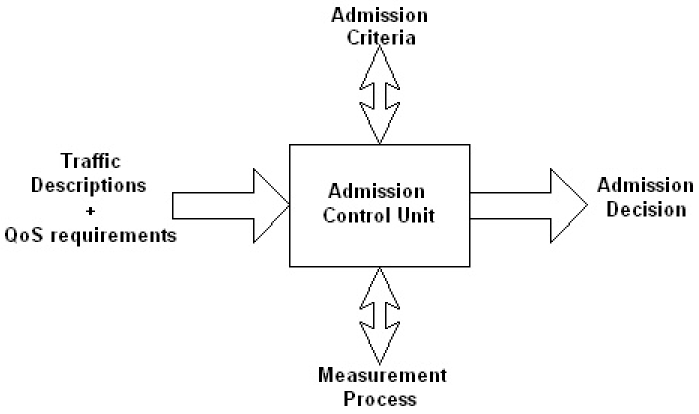


Fig. 4: Admission Control Components [43].

3.4.3: Traffic Shaping/Policing

After a traffic stream gets admitted with a given QoS requirement and a given traffic descriptor, it becomes binding on the source to stick to that profile. If a rogue source breaks its contract and sends more than what it had bargained for, then there will be breakdown in the service model. To prevent this possibility, traffic shaping and policing becomes essential.

Token bucket algorithm [42] is almost always used for traffic shaping. Token bucket is synonymous to a bucket with depth 'b', in which tokens are collected at a rate 'r'. When the bucket becomes full, extra tokens are lost. A source can send data only if it can grab and destroy sufficient tokens from the bucket.

Leaky bucket algorithm [42] is used for traffic policing, in which excessive traffic is dropped. Leaky bucket is synonymous to a bucket of dept 'b' with a hole at the bottom that allows traffic to flow at a fixed rate 'r'. If the bucket is full, the extra packets are just dropped.

3.4.4: Packet classification

Every packet, regardless of whether it is a real-time packet or a non-real time packet, is treated equally at all routers in the best-effort Internet. However, real-time multimedia traffic demands differential treatment in the network. Thus the newer Service models will have to use some mechanism to distinguish between real-time and non real-time packets. In practice, this is usually done by *packet marking*. The Type of Service (ToS) field in the IP header can be used for this purpose. Some newer Internet architectures like MPLS make use of short labels that are attached to the front of the IP packets for this purpose.

3.4.5: Packet Scheduling

If differential treatment is to be provided in the network, then FIFO scheduling, traditionally used in routers must be replaced by sophisticated queuing disciplines like Priority Queuing, Weighted Fair Queuing etc. Priority Queuing provides different queues for different traffic types. Every queue has an associated priority in which it is served. Queues with lower priority are served only when there are no packets in all the higher priority queues. One disadvantage of priority queuing is that it might lead to starvation of some low priority flows.

Weighted Fair Queuing also has different queues for different traffic classes. However, every queue is assigned a certain weight ‘w’ and the packets in that queue always get a fraction w/C of the bandwidth, where C is the total link capacity.

3.4.6: Packet Dropping

Under congestion, some packets need to be dropped by the routers. In the past this was done at random, leading to inefficient performance for multimedia traffic. For example, an MPEG encoded packet stream contains I, P and B frames. The I frames are compressed without using any temporal redundancy between frames, while the P and B frames are constructed using motion vectors from I (or P) frames. Thus the packets containing I frames are more important than those containing P or B frames. When it comes to packet dropping the network should give higher dropping priority to the P and B frames as compared to I frame packets. For a survey on the different packet dropping schemes we refer to [23].

3.4.7: QoS Based Routing

The best-effort Internet uses routing protocols such as Open Shortest Path First (OSPF), Routing Information Protocol (RIP), and Border Gateway Protocol (BGP) [41]. These protocols are called best effort routing protocols, and they normally use single objective optimization algorithms, which consider only one metric (either hop count or line cost) and minimize it to find the shortest path from the source to the destination. Thus, all traffic is routed along the shortest path leading to congestion on some links while other links might remain under-utilized. Furthermore, if link congestion is used to derive the line cost such that highly congested links have a higher cost, then such algorithms can cause oscillations in the network, where traffic load continuously shifts from heavily congested links to lightly congested links and this will increase the delay and jitter experienced by the end users.

In QoS-based routing, paths for different traffic flows are determined based on some knowledge of resource availability in the network as well as the QoS requirement of the flows. For example, in Fig. 5, suppose there is a traffic flow from host X to host Y, which requires 4Mbps bandwidth. Thus although path A-B-C is shorter (with just two hops), it will not be selected because it doesn't have enough bandwidth. Instead, path A-E-D-C is selected as it satisfies the bandwidth requirement.

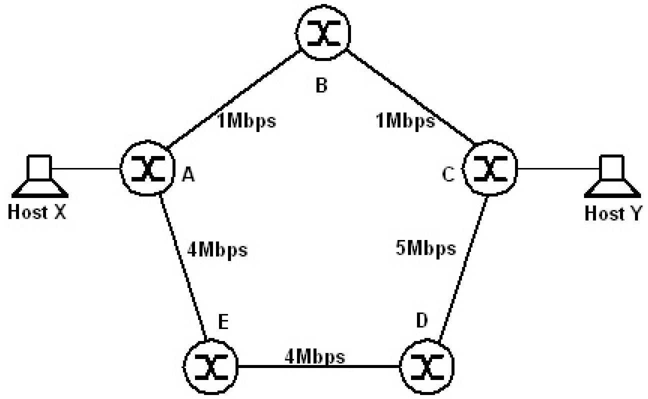


Fig. 5: QoS Based Routing.

Besides QoS based routing, there are two relevant concepts called *Policy-based Routing* and *Constraint-based Routing*. Policy-based Routing [1] commonly means the routing decision is not based on the knowledge of the network topology and metrics, but on some administrative policies. For example, a policy may prohibit a traffic flow from using a specific link for security reason, even if the link satisfies all the QoS constraints. Constraint-based Routing [21] is another new concept, which is derived from QoS-based routing but has a broader sense. In this routing algorithms routes are computed based on multiple constraints, including both QoS constraints and policy constraints. Both QoS-based routing and policy-based routing can be considered as special cases of constraint-based routing.

3.5: Integrated Services

To support multimedia traffic over the Internet, the Integrated Services working group in the Internet Engineering Task Force (IETF) has developed an enhanced Internet service model called Integrated Services (IntServ) [46]. This model is characterized by resource reservations. It requires applications to know their traffic characteristics and QoS requirements beforehand and signal the intermediate network routers to reserve resources, like bandwidth and buffers, to meet them. Accordingly, if the requested resources are available, the routers reserve them and send back a positive acknowledgment to the source, which can then start sending data. If, on the other hand, sufficient resources are not available at any router in the path, the request is turned down and the source has to try again after some time. This model also requires the use of *packet classifiers* to identify flows that are to receive a certain level of service as well as *packet schedulers* to handle the forwarding of different packets in manner to ensure that the QoS commitments are met.

The core of Intserv is almost exclusively concerned with controlling the queuing component of the end-to-end packet delay. Thus, per-packet delay is the central quantity about which the network makes service commitments.

Intserv introduces three service classes to support RTI, RTT and elastic multimedia applications. They are: Guaranteed service, Controlled Load service and the best-effort service. A *flow descriptor* is used to describe the traffic and QoS requirements of a flow [25]. The flow descriptor consists of two parts: a *filter specification* (filterspec) and a *flow specification* (flowspec). The filterspec provides the information required by the packet classifier to identify the packets that belong to that flow. The flowspec consists of a *traffic specification* (Tspec) and *service request specification* (Rspec). Tspec specifies the traffic behavior of the flow in terms of token bucket parameters (*b,r*), while the Rspec specifies the requested QoS requirements in terms of bandwidth, delay, jitter and packet loss.

Since all network nodes, along the path from source to destination, must be informed of the requested resources, a signaling protocol is needed. Resource Reservation Protocol (RSVP) is used for this purpose [6]. The signaling process is illustrated in Fig. 6. The sender sends a PATH message to the receiver, specifying the characteristics of the traffic. Every intermediate router along the path forwards the PATH message to the next hop determined by the routing protocol. The receiver, upon receiving the PATH message, responds with the RESV message to request resources for the flow. Every intermediate router along the path can reject or accept the request of the RESV message. If the request is rejected the router will send an error message to the receiver, and the signaling process terminates. If the request is accepted, link buffer and bandwidth are allocated to the flow, and related flow state information will be installed in the router.

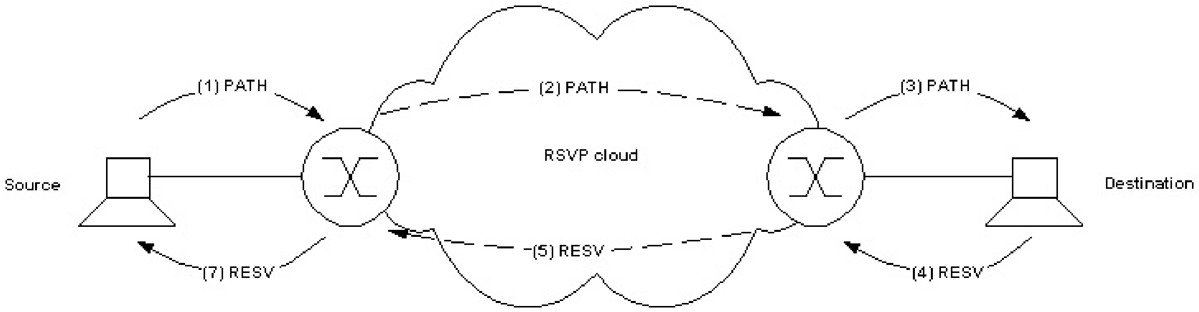


Fig. 6: RSVP signaling.

The design of RSVP lends itself to be used with a variety of QoS control services. RSVP specification does not define the internal format of the RSVP protocol fields, or objects and treats them as opaque and deals only with the setup mechanism. RSVP was designed to support both unicast and multicast applications. RSVP supports heterogeneous QoS, which means different receivers in the same multicast group can request different QoS. This heterogeneity allows some receivers to have reservations while there could be others receiving the same traffic using the best-effort service. We now discuss the service classes offered by IntServ.

3.5.1: Guaranteed Service Class

The Guaranteed service class provides firm end-to-end delay guarantees. Guaranteed service does not control the minimum or average delay of packets, merely the maximal queuing delay. This service guarantees that packets will arrive at the receiver within a requested delivery time and will not be discarded due to queue overflows, provided the flow's traffic stays within its specified traffic limits, which is controlled using traffic policing. This service is intended for applications, which need a firm guarantee that a packet will arrive no later than a certain delay bound.

Using traffic specification (TSpec), the network can compute various parameters describing how it will handle the flow, and by combining the parameters, it is possible to compute the maximum queuing and routing delay that a packet can experience. Using the fluid flow model, the queuing delay is approximately a function of two parameters: the token bucket size ‘b’, and the data rate ‘R’ that the application requests and gets when admitted.

3.5.2: Controlled Load Service

Controlled load Service is an enhanced quality of service intended to support RTT applications requiring better performance than that provided by the traditional best-effort service. It approximates the end-to-end behavior provided by best effort under unloaded conditions. The assumption here is that under unloaded conditions, a very high percentage of the transmitted packets are successfully delivered to the end-nodes, and the transmission delay experienced by a very high percentage of the delivered packets will not vary much from the minimum transit delay.

The network ensures that adequate bandwidth and packet processing resources are available to handle the requested level of traffic. The controlled load service does not make use of specific target values for delay or loss. Acceptance of a controlled-load request is merely a commitment to provide the flow with a service closely equivalent to that provided to uncontrolled traffic under lightly loaded conditions. Over all timescales significantly larger than the burst time, a controlled load service flow may experience little or no average packet queuing delay, and little or no congestion loss.

The controlled load service is described only using a TSpec. Since the network does not give any quantitative guarantees, RSpec is not required. The controlled load flows not experiencing excess traffic will get the contracted quality of service, and the network elements will prevent excess controlled load traffic from unfairly impacting the handling of arriving best-effort traffic. The excess traffic will be forwarded on a best-effort basis.

3.5.3: Best Effort Service

The Best Effort Service class does not have a TSpec or an RSpec. There are no guarantees by the network whatsoever. The network does not do any admission control for this class.

3.5.4: Disadvantages of the Intserv Service Model for the Internet

Intserv uses RSVP to make per-flow reservations at routers along a network path. Although this allows the network to provide service guarantees at the flow level, it suffers from scalability problems. The routers have to maintain per-flow state for every flow that is passing through the router, which can lead to huge overhead. Moreover RSVP is a soft-state protocol, which means that the router state has to be refreshed at regular intervals. This increases traffic overhead.

3.6: Differentiated Services

Diffserv working group in the IETF, have proposed the Differentiated Services (Diffserv) service model for the Internet, which removes some of the shortcomings of the Intserv architecture [5]. DiffServ divides the network into distinct regions, called DS domains and each DS domain can be controlled by a single entity. For example, an organization's intranet or an ISP can form its own DS domain. One important implication of this is that in order to provide any service guarantees the entire path between the source and destination must be under some DS domain (possibly multiple). Even if a single hop is not under some DS domain, then service cannot be guaranteed. DiffServ architecture can be extended across multiple domains using Service Level Agreement's (SLA) between them. A SLA specifies rules for traffic remarking, actions to be taken for out-of-profile traffic etc.

Every node in a DS domain can be either a:

Boundary node: Boundary Nodes are the gatekeepers of the DS domain. A boundary node is the first (or last) node that a packet can encounter when entering (or exiting) a DS domain. It performs certain edge functions like admission control, packet classification and traffic conditioning. The admission control algorithm, limits the number of flows that are admitted into the diffserv domain and is distributed in nature. For example, in the simplest case, the admission control algorithm may maintain a central data structure that contains the current status of all links in the DS domain. When a flow is considered for admission, the corresponding Boundary node might just check this data structure to verify if all the links of the flow path can satisfy the requested QoS. Every packet, belonging to an admitted flow, arriving into the DS domain is classified and marked as belonging to one of the service classes, called “Behavior Aggregates” in Diffserv terminology. Each such behavior aggregate is assigned a distinct 8-bit codeword, called the DS code-point. Packet marking is achieved by updating the TOS field in the packet’s IP header with appropriate DS code-point. Boundary nodes also enforce Traffic Conditioning Agreements (TCA) between its own DS domain and the other connected domains, if any.

Interior node: An Interior node is completely inside a DS domain and is connected to other interior nodes or boundary nodes within the same DS domain. The Interior nodes only perform packet forwarding. When a packet with a particular DS code point arrives at this node, it is forwarded to the next hop, according to some pre-defined rule associated with the packet class. Such pre-defined rules are called Per-Hop Behaviors (PHB’s), discussed next.

Thus unlike Intserv, only the edge routers have to maintain per-flow states, which makes Diffserv relatively more scalable.

3.6.1: Per Hop Behavior's

A PHB is a pre-defined rule that influences how the router buffers and link bandwidth are shared among competing behavior aggregates. PHB's can be defined either in terms of router resources (viz. buffer and bandwidth), or in terms of their priority relative to other PHB's or in terms of their relative traffic properties (e.g. delay and loss). Multiple PHB's can be lumped together to form a PHB Group. A particular PHB Group can by implemented in a variety of ways because PHB's are defined in terms of behavior characteristics and are not implementation dependent. Thus PHB's can be considered as basic building blocks for creating services. A PHB for a packet is selected at the first node on the basis of its DS codepoint. The mapping from DS codepoint to PHB maybe 1 to 1 or N to 1. Examples of the parameters of the forwarding behavior that each traffic should receive are bandwidth partition and the drop priority. Examples of implementations of these are Weighted Fair Queuing (WFQ) for bandwidth partition and Random Early Detect (RED) for drop priority. Two commonly used PHB's defined by IETF are:

- **Assured Forwarding (AF) PHB:** AF PHB divides incoming traffic into four classes, where each AF class is guaranteed some minimum bandwidth and buffer space. Within each AF class, packets are further assigned one of three drop priorities. By varying the amount of resources allocated to each AF class, different levels of performance can be offered.

- **Expedited Forwarding (EF) PHB:** EF PHB dictates that the departure rate of a traffic class from any router must equal or exceed the configured rate. Thus for a traffic class belonging to EF PHB, during any interval of time, it can be confidently said that departure rate from any router will equal or exceed the aggregate arrival rate at that router. This has strong implication on the queuing delay that is experience by the packet. In this case, the queuing delay can be guaranteed to be bounded and is negligible (limited by the link bandwidth). EF PHB is used to provide Premium Service (having low delay and jitter) to the customer. However EF PHB requires very strong admission control mechanism. The admission control algorithm will basically ensure that the arrival rate of traffic belonging to EF PHB is less than the departure rate configured at any router in its path. Also proper functioning of EF PHB demands strict policing. This job can be carried out by the Ingress routers. If packets are found to be in violation of the contract, they can be either dropped or demoted to a lower traffic class.

3.7: Multi Protocol Label Switching

[30] When an IP packet arrives at a router, the next hop for this packet is determined by the routing algorithm in operation, which uses the "longest prefix match" (i.e. matching the longest prefix of an IP destination address to the entries in a routing table) to determine the appropriate outgoing link. This process introduces some latency, as the routing tables are very large and table lookups take time. Also the same process needs to be repeated independently for every incoming packet, even though all packets may belong to the same flow and going towards the same destination. This shortcoming of IP routing can be removed by using IP switching, in which a short label is attached to a packet and updated at every hop. When this modified packet arrives at a switch (router in our previous description), this label is used to index into a short switching table (an O(1) operation) to determine the outgoing link and new label for the next hop. The old label is then replaced by new label and the packet is forwarded to the next hop. All this can be easily done in hardware, resulting in very high speeds. This concept was earlier used in ATM for cell switching, which used the VPI/VCI field in the packet as the label. MPLS introduces the same switching concept in IP networks. It is called 'Multi Protocol' since this technique can also be used with any network layer protocol other than IP.

Label switching provides a low-cost hardware implementation, scalability to very high speeds, and flexibility in the management of traffic flows.

Similar to Diffserv, an MPLS network is also divided into domains with boundary nodes, called 'Label Edge Routers' (LER), and interior nodes, called 'Label Switching Routers' (LSR). Packets, entering an MPLS domain, are assigned a label at the ingress LER and are switched inside the domain by a simple label lookup. The labels determine the quality of service that the flow receives in the network. The packets are stripped off the labels at the egress LER and might be routed in the conventional fashion thereafter, before they reach their final destination. A sequence of LSR's that is to be followed by a packet within an MPLS domain is called 'Label Switched Path' (LSP). Again, similar to Diffserv, in order to guarantee certain quality of service to the packet, both the source and destination have to be attached to the same MPLS domain, or if they are attached to different domains, then there should be some service agreement between the two.

MPLS uses the concept of 'Forward Equivalence Class' (FEC) to provide differential treatment to different media types. A group of packets that are forwarded in the same manner are said to belong to the same FEC. There is no limit to the number and granularity of FEC's that can exists. Thus it is possible to define a separate FEC for every flow (which is not advisable, due to large overheads) or for every media type, each tailored for that media type. One important thing to note here is that labels have only local significance in the sense that two LSR's agree upon using a particular label to signify a particular FEC, among themselves. The same label can be used to distinguish a different FEC by another pair of LSR's. Thus it becomes necessary to do label assignments, which includes label allocation and label-to-FEC bindings on every hop of the LSP before the traffic flow can use the LSP. Label assignments can be initiated in one of the following three ways:

- Topology-driven label assignment: In this, LSP's (for every possible FEC) are automatically set up between every pair of LSR (full mesh). Thus this scheme can place a heavy demand on the usage of labels at each LSR. However, the main advantage is that there is no latency involved in setting up an LSP before the traffic flow can use it, i.e. Zero call set up delay.
- Request driven label assignment: Here the LSP's are set up based on explicit requests. RSVP can be used to make the request. The advantage of this scheme is that LSP's will be set up only when required and a full mesh is avoided. However, the disadvantage is the added setup latency, which can dominate short-lived flows.
- Traffic driven label assignment: This combines the advantages of the above two methods. In this, LSP's are set up only when the LSR identifies traffic patterns that justify the setup of an LSP. Those that are identified as not needing an established LSP are routed using the normal routing method.

MPLS also supports label stacking, which can be very useful for performing tunneling operations. In label stacking, labels are stacked in a FILO order. In any particular domain only the topmost label can be used to make forwarding decisions. This functionality can be very useful for providing mobility, where a home agent can push another label on incoming packets and forward the packet to a foreign agent, which pops it off and finally forwards the packet to the destination mobile host.

IV: Enhancing the TCP/IP Protocol Stack to Support *Functional Requirements* of Distributed Multimedia Applications.

In this section, we illustrate standards/protocols that have been introduced to operate over the basic TCP/IP protocol stack to satisfy the *functional requirements* of multimedia traffic streams. Later we describe two protocol architectures, H.323 and Session Initiation Protocol (SIP) that have been standardized to support these functional requirements. Again, to preserve the brevity of this paper, we do not discuss every such approach at length but provide appropriate references for interested readers.

4.1: Supporting Multicasting

The easiest way to achieve multicasting over the Internet is by sending packets to a multicast IP address (Class D IP addresses are multicast IP addresses) [3]. Hosts willing to receive multicast messages for particular multi-cast groups inform their immediate-neighboring routers using Internet Group Management Protocol (IGMP). Multicasting is trivial on a single Ethernet segment (where packets can be multicast using the multicast MAC address). However, for delivering a multicast packet from the source to the destination nodes on other networks, multicast routers need to exchange the information they have gathered from the group membership of the hosts directly connected to them. There are many different algorithms such as "flooding", "spanning tree", "reverse path broadcasting", and "reverse path multicasting" for exchanging the routing information among the routers. Some of these algorithms have been used in dynamic multicast routing protocols such as Distance Vector Multicast Routing Protocol (DVMRP), Multicast extension to Open Shortest Path First (MOSPF), and Protocol Independent Multicast (PIM) [31]. Based on the routing information obtained through one of these protocols, whenever a multicast packet is sent out to a multicast group, multicast routers will decide whether to forward that packet to their network(s) or not.

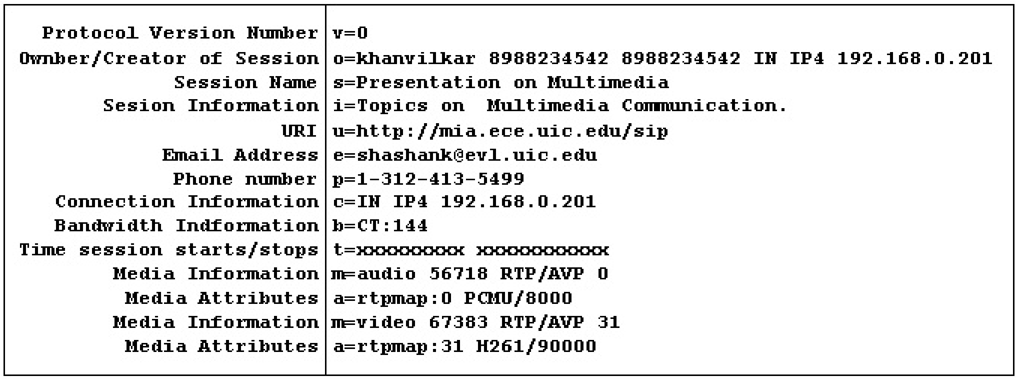
Another approach is MBone or Multicast Backbone. Mbone is essentially a virtual network implemented on top of some portions of the Internet. In the MBone, islands of multicast-capable networks are connected to each other by virtual links called "tunnels". It is through these tunnels that multicast messages are forwarded through non-multicast-capable portions of the Internet. For forwarding multicast packets through these tunnels, they are encapsulated as IP-over-IP (with protocol number set to 4) such that they look like normal unicast packets to intervening routers.

ITU-T H.323 and IETF Session Initiation protocol (SIP), as discussed in detail later, support multicasting through the presence of a Multi-point control unit, that provides both mixing and conferencing functionality needed for audio/video conferencing. [39] gives a survey of QoS Multicasting issues.

4.2: Session Management

We now discuss the different protocols that have been standardized to meet different aspects of session management.

Session Description: Session Description Protocol [15][17], developed by IETF, can be used for providing the session description functionality (to describe media type, media encoding used for that session). It is more of a description syntax than a protocol as it does not provide a full-range media negotiation capability (This is provided by SIP, as discussed in later sections). SDP encodes media descriptions in simple text format. An SDP message is composed of a series of lines, called *fields*, whose names are abbreviated by a single lower-case letter, and are in a required order to facilitate parsing. The fields are in the form *attribute_type=value*. A sample SDP message is illustrated in Fig. 7. The meaning of all attributes is illustrated on the left, while the actual message is illustrated on the right.



Protocol Version Number	v=0
Ownber/Creator of Session	o=khanvilkar 8988234542 8988234542 IN IP4 192.168.0.201
Session Name	s=Presentation on Multimedia
Sesion Information	i=Topics on Multimedia Communication.
URI	u=http://mia.ece.uic.edu/sip
Email Address	e=shashank@evl.uic.edu
Phone number	p=1-312-413-5499
Connection Information	c=IN IP4 192.168.0.201
Bandwidth Indformation	b=CT:144
Time session starts/stops	t=xxxxxxxxx xxxxxxxxxxx
Media Information	m=audio 56718 RTP/AVP 0
Media Attributes	a=rtpmap:0 PCMU/8000
Media Information	m=video 67383 RTP/AVP 31
Media Attributes	a=rtpmap:31 H261/90000

Fig. 7: Sample SDP message.

Session Announcement: Session Announcement Protocol (SAP) [16][2] can be used for this purpose. This protocol is used for advertising the multicast conferences and other multicast sessions. An SAP announcer periodically multicasts announcement packets to a well-known multicast address and port (port number 9875) with the same scope as the session it is announcing, ensuring that the recipients of the announcement can also be potential recipients of the session being advertised. Multiple announcers may also announce a single session, to increase robustness against packet loss or failure of one or more announcers. The time period between repetitions of an announcement is chosen such that the total bandwidth used by all announcements on a single SAP group remains below a pre-configured limit. Each announcer is expected to listen to other announcements in order to determine the total number of sessions being announced on a particular group. SAP is intended to announce the existence of a long-lived wide area multicast sessions and involves a large startup delay before a complete set of announcements is heard by a listener. SAP also contains mechanisms for ensuring integrity of session announcements, for authenticating the origin of an announcement and for encrypting such announcements.

Session Identification: In the best-effort Internet, every flow (or session) can be identified using the tuple <*Src Ip, Src Port, Dst IP, Dst Port, Protocol*>. Thus individual transport layer sockets have to be established for every session. However, if there ever arises a need to bunch sessions together (for cutting costs) there is no available mechanism. Hence there is clearly a need to multiplex different streams into the same transport layer socket. This functionality is similar to that of the "Session Layer" in the 7-layer OSI model, which has been notably absent in TCP/IP protocol stack used in the Internet. Session identification can be done using RTP, and is described in more detail in the next point.

Session Control: All the session control functionalities can be satisfied using a combination of RTP, RTCP and RTSP. Below we briefly describe each of them.

Real-time Protocol (RTP) [35] typically runs on top of UDP. Specifically, chunks of audio/video data that are generated by the sending side of the multimedia application are encapsulated in RTP packets, which in turn are encapsulated in UDP. The RTP functionality needs to be integrated into the application. The functions provided by RTP include: (i) Sequencing: A *sequence number* field is included in the RTP header to detect lost packets. (ii) Payload Identification: To dynamically change media encoding schemes to adjust to changing bandwidth. To provide this functionality, a *payload identifier* is included in each RTP packet to describe the encoding of the media. (iii) Frame Indication: Video and audio are sent in logical units called frames. To indicate the beginning and end of the frame, a *frame marker* bit has been provided (iv) Source Identification: In a multicast session, we have many participants. So an identifier is required to determine the originator of the frame. For this *Synchronization Source* (SSRC) identifier has been provided. (v) Intramedia Synchronization: To compensate for the different delay jitter for packets within the same stream, RTP provides *timestamps*, which are needed by the play-out buffers.

Additional information pertaining to particular media types and compression standards can also be inserted in the RTP packets by use of profile headers and extensions. Cavusoglu et. al. [7] have relied on the information contained in the RTP header extensions to provide an adaptive forward error correction (FEC) scheme for MPEG-2 video communications over RTP networks. This approach relies on the group of pictures (GOP) sequence and motion information in order to assign a dynamic weight to the video sequence. The fluctuations in the weight of the video stream are used to modulate the level of FEC assigned to the GOP. Much higher performance is achieved by the use of the adaptive FEC scheme in comparison to other methods that assign an uneven level of protection to video stream. The basic notion presented by the adaptive weight assignment procedure presented in [7] can be employed in other applications such as diffserv networks and selective retransmissions (see discussion on LSP networks a few paragraphs later).

RTCP [35] is a control protocol that works in conjunction with RTP and provides participants with useful statistics about the number of packets sent, number of packets lost, inter-arrival jitter and round trip time. This information can be used by the sources to adjust their data rate. Other information such as email address, name and phone number are included in the RTCP packets, which allows all users to know the identities of the other users for that session.

Real-Time Streaming Protocol (RTSP) [36] is an out-of-band control protocol that allows the media player to control the transmission of the media stream including functions like *pause/resume*, *repositioning playback* etc.

The use of the RTP protocol above UDP provides for additional functionality required for reordering and time stamping the UDP packets. The inherent limitations of the UDP protocol, however, are not completely overcome by the use of RTP. Specifically, the unreliability of UDP persists and consequently there is not guarantee of delivery of the transmitted packets at the receiver. On the other hand, the error-free transmission guarantees provided by the TCP protocol pose severe time delays that render it useless for real-time applications. Mulabegovic et. Al. [27] have proposed an alternative to RTP provided by the Lightweight Streaming Protocol (LSP). This protocol also resides on top of the transport layer and relies on the UDP protocol. LSP provides sequence numbers and timestamps—as facilitated by RTP—in order to reorder the packets and manage the buffer at the receiver. However, unlike UDP and RTP, the LSP protocol allows for limited use of retransmissions in order to minimize the effects of error over the communication network. This is accomplished by use of negative acknowledgements in the event of lost packets and satisfaction of timing delay constraints required to maintain real-time communication capability.

4.3: Security

IpSec [18] [28], provides a suite of protocols that can be used to carry out secure transactions. IpSec adds security protection to IP packets. This approach allows applications to communicate securely without having to be modified. For e.g. before IpSec came into picture, applications often uses SSH or SSL for having a secure peer-to-peer communication, but this required modifying certain API's in the application source code and subsequent recompilation. IpSec cleanly separates policy from enforcement. The policy (which traffic flow is secured and how it is secured) is provided by the system administrator, while the enforcement of this policy is carried out by a set of protocols; viz. Authentication Header (AH) and Encapsulated Security Payload (ESP). The policies are placed as rules in a Secure Policy Database (SPD), consulted for every inbound/outbound packet and tell IpSec how to deal with a particular packet: whether IpSec mechanism needs to be applied to the packet, or the packet should be dropped, or should the packet be forwarded without placing any security mechanism. If the administrator has configured the SPD to use some security for a particular traffic flow, then IpSec first negotiates the parameters involved in securing the traffic flow with the peer host. These negotiations results in the so-called "Security Association (SA)" between the two peers. The SA contain the type of IpSec mechanism to be applied to the traffic flow (AH or ESP) the encryption algorithms to be used, the security keys etc. SA's are negotiated using the Internet key Exchange (IKE) protocol.

As said earlier, IpSec provides two protocols to secure traffic flow. (a) *Authenticated Header (AH):* The main function of the AH is to establish the authenticity of the sender to the receiver. It does provide any data-confidentiality. In other words, AH does not encrypt the payload. It may seem, at first, that authentication without confidentiality might not be useful in the industry. However there are many applications where it does provide a great help. For e.g. there may be many situations, such as news reports, where the data may not be encrypted but it may be necessary to establish the authenticity of the sender. AH provides significantly less overhead as compared to ESP. (b) *Encapsulated Security Payload (ESP):* ESP provides both authentication and encryption services to the IP packets. Since every packet is encrypted, ESP puts higher processing load on the processor.

Currently IpSec can operate in *transport* mode or *tunnel* mode. In transport mode, IpSec takes a packet to be protected, preserves the packet's IP header, and modifies only the upper layer portions by adding IpSec headers and the requested kind of protection between the upper layers and the original IP header. In the tunnel mode, IpSec treats the entire packet as a block of data, adds a new packet header, and protects the data by making it part of the encrypted payload of the new packet.

IpSec can be easily integrated into the current operating system environment, by either changing the native implementation of IP protocol (and subsequent kernel compilation), or inserting an additional layer below the IP layer of the TCP/IP protocol stack (also known as Bump-In-The-Stack (BITS)), or using some external hardware (also known as Bump-In-The-Wire (BITW)).

4.4: Mobility

Mobile IP [29] is an Internet protocol used to support mobility. Its goal is to provide the ability of a host to stay connected to the Internet regardless of its location. Every site that wants to allow its users to roam has to create a *Home Agent* (HA) entity and every site that wants to allow visitors, has to create a *Foreign Agent* (FA) entity. When a Mobile Host (MH) roams into a foreign site (which allows roaming), it contacts and registers with the FA for that site. The FA in turn, contacts the HA of that mobile host and gives it a *care-of-address*, normally the FA's own IP address. All packets destined for the MH will eventually reach the HA, which then encapsulates the packets inside another IP packet and forwards it to the FA, which then de-capsulates it and ultimately forwards the original packet to the MH. The HA may also inform the source of the new IP address where the packets must be directly forwarded.

4.5: H.323

ITU-T recommendation H.323 [44][26][2] is an umbrella recommendation that specifies the components, protocols, and procedures used to enable voice, video and data conferencing over a packet-based network like the IP-based Internet or IPX–based local-area networks. H.323 is a very flexible recommendation that can be applied in a variety of ways—audio only (IP telephony); audio and video (video-telephony); audio and data; and audio, video and data; over point-to-point and point-to-multipoint multimedia conferencing. It is not necessary for two different clients to support the same mechanisms in order to communicate, as individual capabilities are exchanged at the beginning of any session, and communication is set up based on the lowest common denominator. Also point-to-multipoint conferencing can be supported without the presence of any specialized hardware/software.

H.323 is a part of a family of ITU—T recommendations, illustrated in **Table 6**, called H.32x that provide multimedia communication services over a wide variety of networks. Inter-operation between these different standards is effected by the presence of a 'Gateway' that provides data format translation, control signaling translation, audio and video codec translation, and call setup/termination functionality.

Table 6: ITU-T recommendations for Audio/Video/Data conferencing standards.

ITU-T recommendation	Underlying Network over which audio, video and data conferencing is provided.
H.320	ISDN
H.321 and H.310	ATM
H.322	LAN's that provide a guaranteed QoS
H.323	LAN's and Internet
H.324	PSTN/Wireless

The H.323 standard defines four components, which, when networked together, provide the point-to-point and point-to-multipoint multimedia-communication services: (a) Terminals: These are the end-points of the H.323 conference. A multimedia PC with a H.323 compliant stack can act as a terminal. (b) Gateway: As discussed earlier, gateway is only needed whenever conferencing needs to be done between different H.32X-based clients. (c) Gatekeeper: This provides many functions including admission control and bandwidth management. Terminals must get permission from the gatekeeper to place any call. (d) Multi-point Control Unit (MCU): This is an optional component that provides point-to-multipoint conferencing capability to an H.323 enabled network. The MCU consists of a mandatory Multipoint Controller (MC) and optional Multipoint Processors (MP). The MC determines the common capabilities of the terminals by using H.245 but it does not perform the multiplexing of audio, video and data. The multiplexing of media streams is handled by the MP under the control of the MC. Fig. 8, illustrates an H.323 enabled network with all these different components.

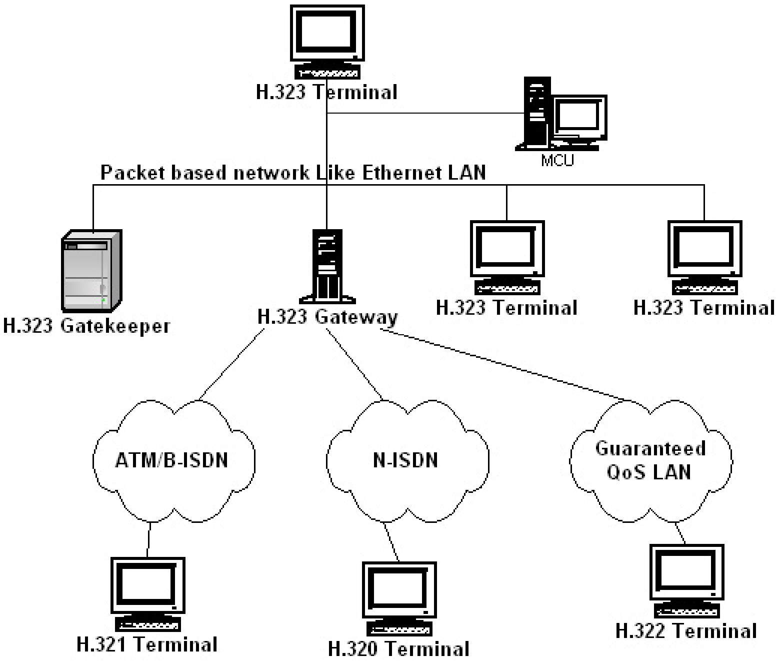


Fig. 8: An H.323 enabled network with different components.

Fig. 9: illustrates the protocol stack for H.323. RTP and its associated control protocol, RTCP, are employed for timely and orderly delivery of packetized audio/video streams. The operation of RTP and RTCP has been discussed in earlier sections.

The H.225 RAS (Registration, Admission and Status) is mainly used by H.323 end-points (terminals and gateways) to discover a gatekeeper, register/un-register with the gatekeeper, requesting call admission and bandwidth allocation and clearing a call. The gatekeeper can also use this protocol for inquiring on an end-point and for communicating with other peer gateways.

The Q.931 signaling protocol is used for call setup and teardown between two H.323 end-points and is a lightweight version of the Q.931 protocol defined for PSTN/ISDN. The H.245 media control protocol is used for negotiating media processing capabilities such as audio/video codec to be used for each media type between two terminals and determining Master-Slave relationships.

Real-time data conferencing capability is required for activities such as application sharing, whiteboard sharing, file transfer, fax transmission, and instant messaging. Recommendation T.120

provides this optional capability to H.323. T.120 is a real-time data communication protocol designed specifically for conferencing needs. Like H.323, Recommendation T.120 is an umbrella for a set of standards that enable the real-time sharing of specific applications data among several clients across different networks.

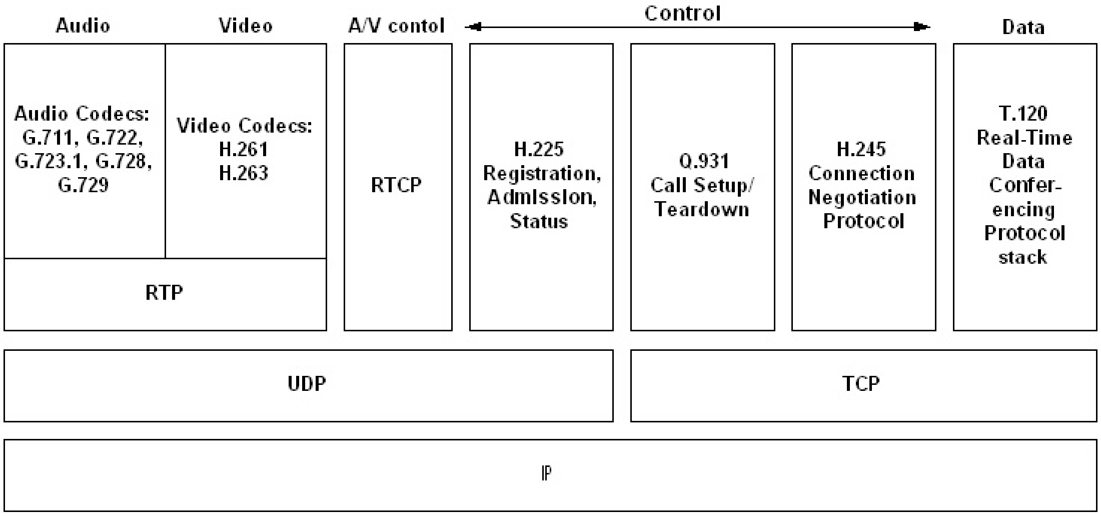

Audio
Video
A/V contol
Control
Data
Audio Codecs: G.711, G.722, G.723.1, G.728, G.729
Video Codecs: H.261 H.263
RTP
RTCP
H.225 Registration, Admlsslon, Status
Q.931 Call Setup/ Teardown
H.245 Connection Negotiation Protocol
T.120 Real-Time Data Confer-encing Protocol stack
UDP
TCP
IP

Fig. 9: H.323 Protocol Stack.

Table 7 illustrates the different phases involved in setting up a point-to-point H.323 conference, when a gatekeeper is present in an H.323 network. The first three phases correspond to call setup while the last three correspond to call teardown. When no gatekeeper is involved, phases 1 and 7 are omitted.

Accordingly two call control models are supported in H.323: direct call and gatekeeper-routed call. In the direct call model, all Q.931 and H.245 signaling messages are exchanged directly between the two H.323 endpoints; so are RTP media streams. As long as the calling endpoint knows the transport address of the called endpoint, it can set up a direct call with the other party. This model is unattractive for large-scale carrier deployments because carriers may be unaware of which calls are being set up and this may prevent them from providing sufficient resources for the call and charging for it. In the gatekeeper-routed call model, all signaling messages are routed through the gatekeeper. In this case, use of RAS is necessary. This model allows endpoints to access services provided by the gatekeeper, such as address resolution and call routing. It also allows the gatekeepers to enforce admission control and bandwidth allocation over their respective zones. This model is more suitable for IP telephony service providers since they can control the network and exercise accounting and billing functions.

Table 7: Phases in an H.323 call.

Phase	Protocol	Intended functions
1. Call admission	RAS	Request permission from gatekeeper to make/receive a call. At the end of this phase, the calling endpoint receives the Q.931 transport address of the called endpoint.
2. Call setup	Q.931	Set up a call between the two endpoints. At the end of this phase, the calling endpoint receives the H.245 transport address of the called endpoint.
3. Endpoint capability	H.245	Negotiate capabilities between two endpoints. Determine master-slave relationship. Open logical channels between two endpoints. At the end of this phase, both endpoints know the RTP/RTCP addresses of each other.
4. Stable call	RTP	Two parties in conversation.
5. Channel closing	H.245	Close down the logical channels.
6. Call teardown	Q.931	Tear down the call.
7. Call disengage	RAS	Release the resources used for this call

4.6: Session Initiation Protocol.

SIP is an application-layer signaling protocol for creating, modifying, and terminating multimedia sessions (voice, video or data) with either one or more participants [17][34][2]. SIP does not define what a "session" is; this is defined by the content carried opaquely in SIP messages. To establish a multimedia session, SIP has to go through:

- *Session Initiation:* Initiating a session, is perhaps the hardest part, as it requires determining where the user to be contacted is residing at the current moment: the user may be at home working on his Home PC or he may be at work working on his office PC. Thus SIP allows user's to be located and addressed by a single global address (usually his e-mail address) irrespective of his physical location.
- *Delivering session description:* Once the user is located, SIP performs the second function of delivering a description of the session that the user is invited to. SIP, itself is opaque to the session description, in the sense that it does not know anything about the session. It merely indicates the user about the protocol to be used to understand the session description. Session Description Protocol (SDP) is the most common protocol used for this purpose. SIP can also be used to decide a common format to describe a session, so that protocols other than SDP can also be used.
- *Active Session management:* Once the session description is delivered, SIP conveys the response (accept or reject) to the session initiation point (the caller). If the response is 'accept' the session becomes active. If the session involves multimedia, media streams can now be exchanged between the two users. RTP and RTCP are some common protocols for transporting real-time data. SIP can also be used to change the parameters of an active session; like removing some video media stream or reducing the quality of the audio stream etc.
- *Session Termination:* Finally, SIP is used to terminate the session.

Thus, SIP is only a signaling protocol and must be used in conjunction with other protocols like SDP, RTP, RTCP etc. in order to provide a complete multimedia service architecture as the one provided in H.323. Note that the basic functionality and operation of SIP does not depend on any of these protocols.

The SIP signaling system consists of the following components:

- User Agents: End system acting on behalf of a user. If the user-agent initiates SIP requests, it is called user agent client (UAC), while a user-agent server (UAS) receives such requests and return responses.
- Network Servers: There are 3 types of servers within a network:

- Registration Server (or Registrars): This keeps track of the user location: i.e. the current PC or terminal on which the user resides. The User Agent sends a registration message to the SIP Registrar and the Registrar stores the registration information in a location service via a non-SIP protocol (e.g. LDAP). Once the information is stored, the Registrar sends the appropriate response back to the user agent.
- Proxy server: Proxy servers are application-layer routers that receive SIP requests and forwards them to the next-hop server that may have more information about the location of the called party.
- Redirect Server: Redirect servers receive requests and then return the location of another SIP user agent or server where the user might be found.

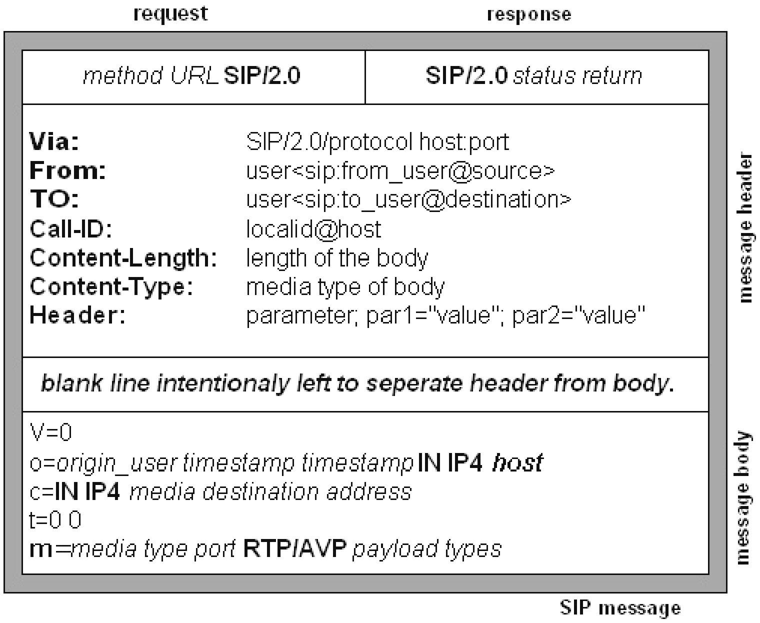
It is quite common to find proxy, redirect, and registrar servers implemented within the same program.

SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method, or function, on the server and at least one response. Like HTTP the all requests and responses use textual encoding. Some commands and responses of SIP and their use are illustrated in Table 8.

Table 8: Commands and responses used in SIP.

Method	Used
INVITE	for inviting a user to a call
ACK:	for reliable exchange of invitation messages
BYE:	for terminating a connection between the two end points
CANCEL:	for terminating the search for a user
OPTIONS:	for getting information about the capabilities of a call
REGISTER:	Gives information about the location of a user to the SIP registration server.

Message format for SIP is shown Fig. 10. The message body is separated from the header by a blank line. The 'Via' indicates the host and port at which the caller is expecting a response. When an SIP message goes through a number of proxies, each such proxy appends to this field with its own address and port. This enables the receiver to send back an acknowledgement through the same set of proxies. The 'From' (and 'To') fields specifies the SIP URI of the sender (or receiver) of the invitation, which is usually the email address assigned to the user. The 'call-Id' contains a globally unique identifier of this call generated by a combination of a random string and IP address. The ‘Content-Length’ and ‘Content-Type’ field describe the body of the SIP message.



method URL **SIP/2.0**	**SIP/2.0** *status return*

```
Via:              SIP/2.0/protocol host:port
From:             user<sip:from_user@source>
TO:               user<sip:to_user@destination>
Call-ID:          localid@host
Content-Length:   length of the body
Content-Type:     media type of body
Header:           parameter; par1="value"; par2="value"

blank line intentionaly left to seperate header from body.

V=0
o=origin_user timestamp timestamp IN IP4 host
c=IN IP4 media destination address
t=0 0
m=media type port RTP/AVP payload types
```

Fig. 10: SIP message format.

We now illustrate a simple example (refer Fig. 11) that captures the essence of SIP operations. Here a client (caller) is inviting a participant (callee) for a call. The SIP client creates an INVITE message for callee@uic.edu, which is normally sent to a proxy server (Step:1). This proxy server tries to obtain the IP address of the SIP server that handles requests for the requested domain. The proxy server consults a Location Server to determine this next hop server (Step: 2). The Location server is a non-SIP server that stores information about the next hop servers for different users and returns the IP address of the machine where callee can be found (Step: 3). On getting this IP address, the proxy server forwards the INVITE message (Step: 5) to the host machine. After the User Agent Server (UAS) has been reached, it sends an OK response back to the proxy server (Step: 8), assuming that the callee wants to accept the call . The proxy server in-turn sends back an OK response to the client (Step: 9). The client then confirms that it has received the response by sending an ACK (Step: 10). After this a full-fledged multimedia session is initiated between the two participants. At the end of this session, the callee sends a BYE message to the caller (Step: 11), which in turn ends the session with another BYE message (Step: 12). Note that we have skipped the TRYING and RINGING message exchanges in the above explanation.

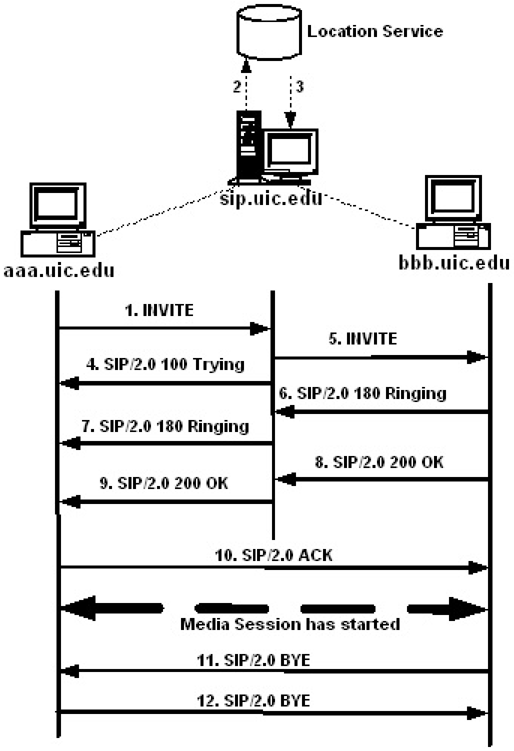


Fig. 11: Timeline for a typical SIP session.

V: Quality of Service Architecture for 3rd Generation Cellular Systems

Over the past decade there has been a phenomenal growth in the development of cellular networks around the world. Wireless communications technology has evolved over these years, from a simple 1st Generation (1G) analog system supporting only voice (AMPS, NMT, TACS etc.), to the current 2nd Generation (2G) digital systems (GSM, IS-95, IS-136, etc.) supporting voice and low rate data and still evolving towards the 3rd Generation (3G) digital system (IMT-2000, cdma2000 etc.) supporting multimedia [13].

IMT-2000/UMTS is the 3G specification under development by the ITU that will provide enhanced voice, data, and multimedia services over wireless networks. In its current state, the plan is for IMT-2000 to specify a "family of standards" that will provide at least 384 kbps data rate at pedestrian speeds, 144 kbps at mobile speeds, and up to 2 Mbps in an indoor environment. Numerous standards bodies throughout the world have submitted proposals to the ITU on UMTS/IMT-2000.

In 1997 Japan's major standards body, the Association for Radio Industry and Business (ARIB), became the driving force behind a third-generation radio transmission technology known as wideband CDMA (WCDMA) [37]. In Europe, the European Telecommunications Standards Institute (ETSI) Special Mobile Group (SMG) technical subcommittee has overall responsibility for UMTS standardization. ETSI and ARIB have managed to merge their technical proposal into one harmonized WCDMA standard air interface. In the United States, the Telecommunications Industry Association (TIA) has proposed two air interface standards for IMT-2000, one based on CDMA and the other based on TDMA. Technical committee TR45.5 within TIA proposed a CDMA-based air interface, referred to as cdma2000, that maintains backward compatibility with existing IS-95 networks. The second proposal comes from TR45.3, which adopted the Universal Wireless Communications Consortium's (UWCC) recommendation for a third-generation air interface that builds off on existing IS-136 networks. Last but not least, the South Korean Telecommunications Technology Association (TTA) supports two air interface proposals, one similar to WCDMA and the other to cdma2000.

In the discussion that follows, we describe the layered QoS architecture that has been adopted by the UMTS standard [14] [10]. In UMTS there is a clear separation between the radio access network (called UMTS Terrestrial Radio Access Network or UTRAN), which comprises of all the air-interface related functions (like medium access) and the Core Network (CN) that comprises of the switching and control related functions. Such separation allows both the CN and the radio access network to evolve independently of each other. UTRAN and CN exchange information over the Iu air interface.

Network services are end-to-end, i.e. from Terminal Equipment (TE) to another TE. An end-to-end service has certain QoS requirements, which is provided to the user by the network (called network bearer service in telecommunication terminology). A network bearer service describes how a given network provides QoS and is set up from the source to the destination. The UMTS bearer service layered architecture is illustrated in Fig. 12. Each bearer service at layer N offers its service by using the services provided to it by layer (N – 1).

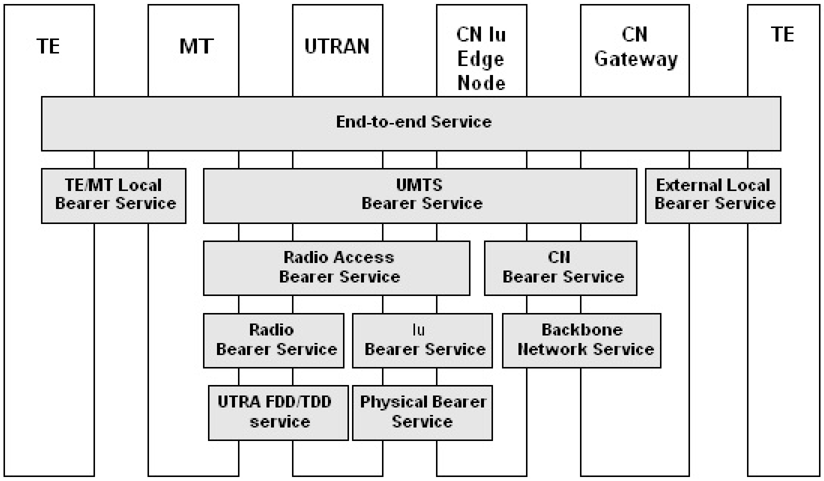


Fig. 12: UMTS QoS Architecture[9].

In the UMTS architecture, the end-to-end bearer service can be decomposed into three main components, which are the terminal equipment TE/MT (Mobile Terminal) local bearer service, the external local bearer service, and the UMTS bearer service.

The TE/MT local bearer service enables communication between the different components of a mobile station. These components are an MT, mainly responsible for the physical connection to the UTRAN through the air interface, and one or several attached end user devices, also known as TEs. Examples of such devices are communicators, laptops, or traditional mobile phones.

The external bearer service connects the UMTS core network and the destination node located in an external network. This service may use IP transport or other alternatives (like those provided by Intserv, Diffserv or MPLS).

The UMTS bearer service uses the radio access bearer service (RAB) and the core network bearer service (CN). Both the RAB and CN reflect the optimized way to realize UMTS bearer service over the respective cellular network topology taking into account aspects such as mobility and mobile subscriber profiles. The RAB provides confidential transport of signaling and user data between the MT and the *CN Iu edge node* with the QoS negotiated by the UMTS bearer service. This service is based on the characteristics of the radio interface and is maintained even for a mobile MT.

The CN service connects the UMTS *CN Iu edge node* with the *CN gateway* to the external network. The role of this service is to efficiently control and utilize the backbone network in order to provide the contracted UMTS bearer service. The UMTS packet CN shall support different backbone bearer services for a variety of QoS options.

The RAB is realized by a radio bearer service and an Iu-bearer service. The role of the radio bearer service is to cover all the aspects of the radio interface transport. This bearer service uses the UTRA frequency-/time-division duplex (FDD/TDD). The Iu bearer service provides the transport between the UTRAN and CN. Iu bearer services for packet traffic shall provide different bearer services for different levels of QoS.

The CN service uses a generic backbone network service. The backbone network service covers the layer 1 and layer 2 functionality and is selected according to the operator's choice in order to fulfill the QoS requirements of the CN bearer service. The backbone network service is not specific to UMTS but may reuse an existing standard.

REFERENCES

[1] Z. Avramovic, *“Policy based routing in the Defense Information System Network”*, IEEE Military Communications Conference, vol: 3, pp: 1210-1214.

[2] R. Arora, R. Jain, *"Voice over IP : Protocols and Standards"*, Class report avail. on-line at ftp://ftp.netlab.ohio-state.edu/pub/jain/courses/cis788-99/voip_protocols/index.html

[3] M. Banikazemi, R. Jain, *"IP Multicasting: Concepts, Algorithms, and Protocols"*, Class report avail. on-line at http://ftp.netlab.ohio-state.edu/pub/jain/courses/cis788-97/ip_multicast/index.htm

[4] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987.

[5] S. Blake, D. Black, M. Carlson, E. Davies, et. al., “*An Architecture for Differentiated Services”*, RFC2475, December 1998.

[6] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, *"Resource ReSerVation Protocol (RSVP): Version 1 Functional Specification"*, RFC2205, September 1997.

[7] B. Cavusoglu, D. Schonfeld, and R. Ansari, *“Real-time adaptive forward error correction for MPEG-2 video communications over RTP networks”*, Proceedings of the IEEE International Conference on Multimedia and Expo, Baltimore, Maryland, 2003, to appear.

[8] N. Chapman and J. Chapman, *Digital Multimedia*, John Wiley and Sons Ltd, 2000.

[9] B. P. Crow, I. Widjaja, J.G. Kim, P.T. Sakai, *"IEEE 802.11: Wireless Local Area Networks"*, IEEE Communications Magazine, September 1997.

[10] S. Dixit, Y. Guo, and Z. Antoniou, *“Resource Management and Quality of Service in Third-Generation Wireless Networks”*, IEEE Communications Magazine, February 2001.

[11] B. Furht, *Multimedia Tools and Applications*, Kluwer Academic Publishers, 1996.

[12] B. Furht, *“Real-time issues in distributed multimedia systems*”, Proceedings of the Second Workshop on Parallel and Distributed Real-Time Systems, April 1994, pp: 88-97.

[13] V.K. Garg, *IS-95 CDMA and cdma2000*, Prentice Hall, 1999.

[14] V.K. Garg and O.T.W Yu, *"Integrated QoS support in 3G UMTS networks"*, IEEE Wireless Communications and Networking Conference, vol: 3, pp: 1187 -1192.

[15] M. Handley and V. Jacobson, *“SDP: Session Description Protocol”*, RFC2327, April 1998.

[16] M. Handley, C. Perkins and E. Whelan, *“Session Announcement Protocol”*, RFC2974, October 2000.

[17] A.B. Johnston, *Understanding the Session Initiation Protocol,* Artech House, 2000.

[18] S. Kent and R. Atkinson, *"Security Architecture for the Internet Protocol"*, RFC2401, November 1998.

[19] G. Kessler, “*An overview of cryptography”*, online version avail. at http://www.garykessler.net/library/crypto.html

[20] W. Kinsner, *"Compression and its metrics for multimedia",* Proceedings of First IEEE International Conference on Cognitive Informatics, 2002, pp: 107-121.

[21] F. Kuipers, P.V. Mieghem, T. Korkmaz and M. Krunz, *"An Overview of Constraint-Based Path Selection Algorithms for QoS Routing"*, IEEE Communications Magazine, December 2002.

[22] J. Kurose and K. Ross, *Computer Networking: A top-down approach featuring the Internet*, Addision Wesley, 2001.

[23] M. Labrador and S. Banerjee, *“Packet Dropping policies for ATM and IP networks”*, IEEE communication surveys, 3rd Quarter 1999, vol: 2(3).

[24] J. Leigh, O. Yu, D. Schonfeld, R. Ansari, et. al., *“Adaptive networking for tele-immersion”*, Proc. Immersive Projection Technology/Eurographics Virtual Environments Workshop (IPT/EGVE), Stuttgart, Germany, 2001.

[25] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*, McGraw-Hill, 2000.

[26] H. Liu and P. Mouchtaris, *"Voice over IP Signaling: H.323 and Beyond"*, IEEE Communications Magazine, October 2000.

[27] E. Mulabegovic, D. Schonfeld, and R. Ansari, *“Lightweight Streaming Protocol (LSP)”*, ACM Multimedia Conference, Juan Les Pins, France, 2002.

[28] R. Oppliger, *“Security at the Internet Layer”*, Computer, vol: 31(9), September 1998.

[29] C.E. Perkins, *“Mobile networking through Mobile IP”*, IEEE Internet Computing, vol: 2(1) Jan/Feb 1998, pp: 58-69.

[30] E. Rosen, A. Viswanathan, R. Callon, *“Multiprotocol Label Switching Architecture”*, RFC3031, January 2001.

[31] L.H. Sahasrabuddhe and B.M. Mukherjee, *"Multicast Routing Algorithms and Protocols: A Tutorial"*, IEEE Network, January/February 2000.

[32] D. Salomon, *Data Compression: The Complete Reference,* Springer, 1998.

[33] D. Schonfeld, *“Image and Video Communication Networks”*, (Invited Chapter). Handbook of Image and Video Processing. A. Bovik (ed.), Academic Press: San Diego, California, Chapter 9.3, pp. 717-732, 2000.

[34] H. Schulzrinne and J. Rosenberg, *"The Session Initiation Protocol: Internet-Centric Signaling"*, IEEE Communications Magazine, October 2000.

[35] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, *“RTP: A Transport Protocol for Real-Time Applications”*, RFC1889, January 1996.

[36] H. Schulzrinne, A. Rao and R. Lanphier, *"Real Time Streaming Protocol (RTSP)"*, RFC2326, April 1998.

[37] J. J. Steinbugl and R. Jain, *“Evolution Toward 3G Wireless Networks”*, Class report avail. on-line at ftp://ftp.netlab.ohio-state.edu/pub/jain/courses/cis788-99/3g_wireless/index.html

[38] R. Steinmetz, *"Analyzing the multimedia operating system"*, IEEE Multimedia, vol: 2(1), Spring 1995, pp: 68-84.

[39] A. Striegel and G. Manimaran, *"A Survey of QoS Multicasting Issues"*, IEEE Communications Magazine, June 2002.

[40] J. Su, F. Hartung, and B. Girod, *"Digital Watermarking of Text, Image, and Video Documents"*, Computers & Graphics, vol: 22 (6), pp: 687 - 695, February 1999.

[41] W. Sun and R. Jain, *“QoS/policy/constraint based routing”*, Class report avail. on-line at ftp://ftp.netlab.ohio-state.edu/pub/jain/courses/cis788-99/qos_routing/index.html.

[42] A. Tanenbaum, *Computer Networks, 3e*, Prentice Hall, 1996.

[43] N. Tang, S. Tsui and L. Wang, *"A Survey of Admission Control Algorithms"*, on-line report available at http://www.cs.ucla.edu/~tang/

[44] G.A. Thom, *"H.323: the multimedia communications standard for local area networks"*, IEEE Communications Magazine , vol: 34(12), December 1996, pp: 52-56.

[45] J. Watkinson, *The MPEG Handbook: MPEG-I, MPEG-II, MPEGIV*, Focal Press, 2001.

[46] P. P. White, *“RSVP and integrated services in the Internet: a tutorial”*, IEEE Communications Magazine, vol: 35(5), May 1997, pp: 100-106.

[47] C. Wu and J. Irwin, *Multimedia Computer Communication Technologies*, Prentice Hall, 1998.

[48] L.C. Wolf, C. Griwodz and R. Steinmetz, *"Multimedia Communication"*, Proceedings of the IEEE, vol: 85(12), December 1997.

[49] Yu, O. and Khanvilkar, S., *“Dynamic Adaptive Guaranteed QoS Provisioning over GPRS Wireless Mobile Links”*, ICC2002, vol: 2, pp: 1100-1104.