Course compendium in network modelling

Ernst Nordström

Department of Computer Science, Dalarna University

SE-781 88 Borlänge, Sweden

eno@du.se

June 27, 2006

Contents

Part I

Human perception and multimedia coding/compression

Chapter 1

Human perception

Humans perceive sounds and images differently and therefore different priorities should be used regarding transport of sounds and images [29]. The ear can be modelled and described as a “differentiator”, i.e. the human can detect very small variations in sound signals both in terms of frequency and amplitude. On the other hand, the eye can be modelled and described as an “integrator” which means that the eye can not detect small variations between and within images in a video sequence. Since humans are more sensitive to sound changes than image changes, the transport of sound should have higher loss priority.

1.1 Hearing

Sounds are transmitted as variations in air pressure. The ear converts variations in air pressure to neural signals and sends then to the brain [10]. The conversion can be modelled with a number of band pass filters [29]. The filters has bandwidths of 50-100 Hz for frequencies around 500 Hz and bandwidths up to 5 kHz for high frequencies. The range of the filters overlaps each other and a reasonable model uses 25 filters which covers the audible frequency range (20 - 20,000 Hz). Figure 1.1 shows the sensitivity of the human ear as function of the frequency. Only sounds with air pressures above the sound threshold can be heard.

If a disturbance, a masker, occurs in the sound signal it will mask away sounds (make them inaudible) in the neighbor frequencies. Figure 1.2 shows an example of masking immediately after the occurrence of a sound at 200 Hz with pressure level of 50 dB. Masking means that the hearing threshold increases around the frequency of the disturbing sound.

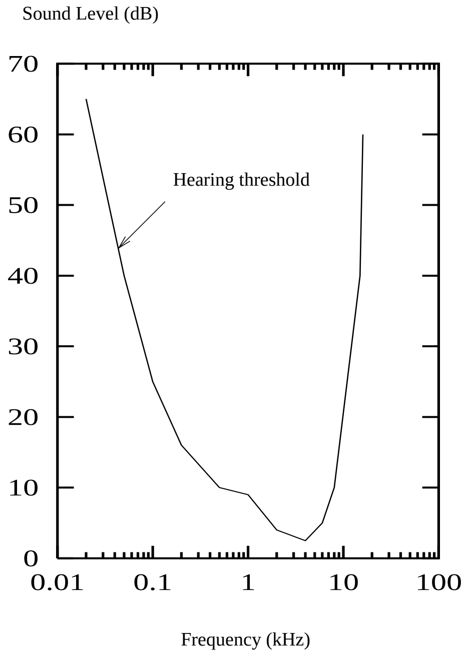


Figure 1.1: Sensitivity of the human ear as function of frequency.

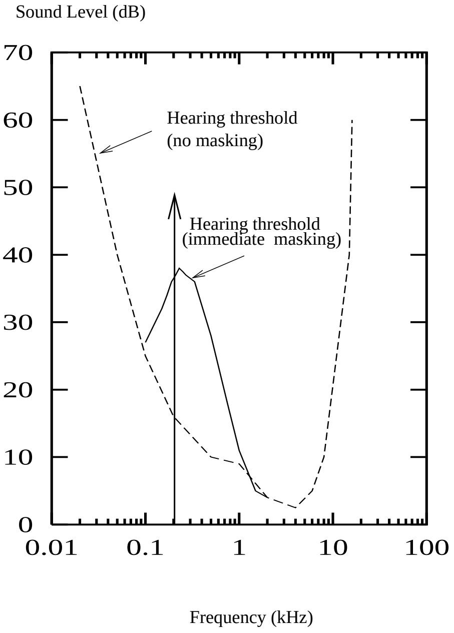


Figure 1.2: Immediate masking of sounds in the human ear.

The masking effect decays with the time. Figure 1.3 shows that lower and lower sound pressure levels can be detected as time passes.

1.2 Vision

Human vision is a complex mechanism [10]. First, the photons impinging on the retina are detected by cones and rods who operate as photo detectors. The *cones* function at high luminance levels. The *rods* are more sensitive and respond to lower luminance levels. Animals that are specialized for night vision have only rods in their retina. Cones are responding preferentially to different narrow bands of the spectrum; they provide a mechanism by which the visual system can respond to the *quality* (i.e. the color) of a source of light, as well as the quantity. The signal detected by the cones and rods is processed by the neural circuits in the eye. The processed signal is then send to the brain visual cortex where the image is interpreted. The human vision is optimized for our natural environment. One example of this is that the we have better resolution along the horizontal axis than along the vertical axis. Important events (dangers) are more often occurring in the horizontal plane rather than in the vertical plane.

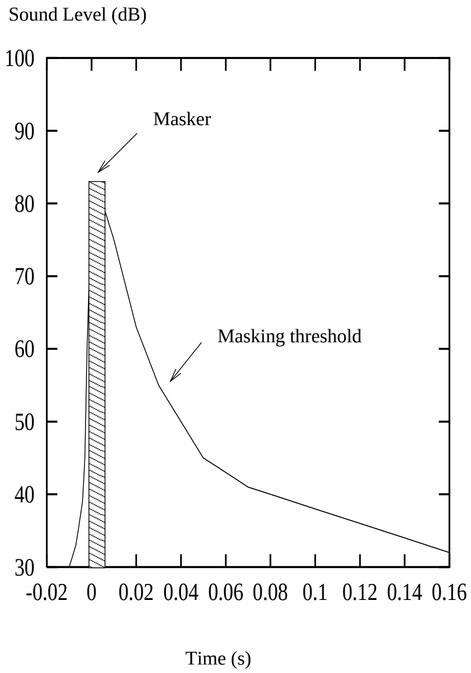


Figure 1.3: Temporal masking of sounds in the human ear.

The vision is not linear in intensity. That is, we will not see twice as well if we double the intensity. However, the contrast sensitivity is always approximately 1% of the intensity. The human is not able to detect too fast or too slow variations in successive images in a video sequence. An optimization again. Variations in successive images can not be detected at a faster rate than around 25 Hz. Hence, there is no point in presenting more than 25 images per second in a video.

The human vision can also be characterized in terms of the sensitivity of spatial intensity variations in the same image [29]. Such intensity variations occurs at a certain spatial frequency. Spatial frequency can be examplified by the stripes on a sweater. The wider the stripes the lower the frequency. The spatial frequency is given as cycles per degree and the maximum of the curve occurs at 3-4 cycles per degree.

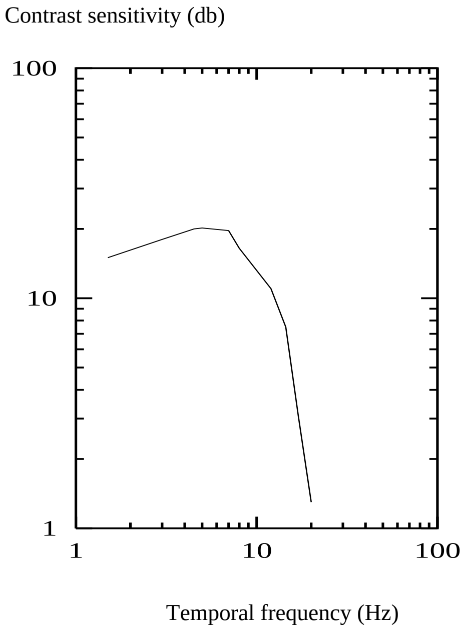


Figure 1.4: Sensitivity if the human eye as function of the temporal frequency.

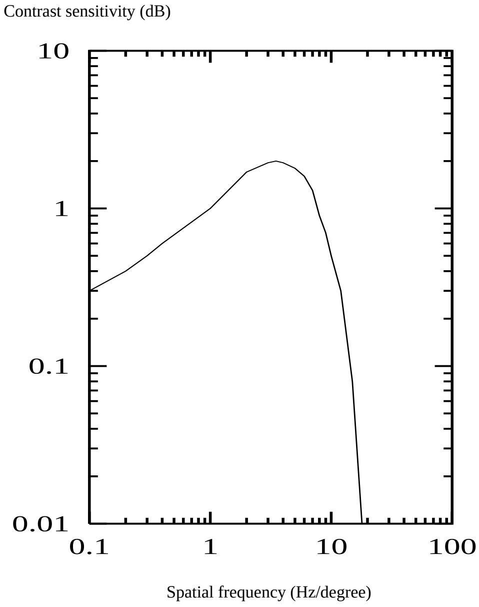


Figure 1.5: Sensitivity of the human eye as function of the spatial frequency.

Chapter 2

Multimedia coding and compression

Multimedia coding and compression is the art of representing raw image, video or audio data with a compact set of symbols or codes. The raw multimedia data can have a considerable volume. The goal of compression is to reduce the data size to facilitate efficient storage and low transmission rates for transport of multimedia data.

This section will discuss coding standards for image, video and audio data. Typical compression ratios are 20:1, 40:1 and 10:1 for image, video and audio, respectively. To achieve a high compression ratio the coding/compression algorithm tries to remove redundancy in the data. The characteristics of human vision and hearing systems sets the framework for the redundancy. Images normally have spatial redundancy, since the eye will not detect fast spatial intensity variations. Video has spatial and temporal redundancy. Fast intensity variations within and between images will not be detected by the eye. Audio also has spatial and temporal redundancy. The spatial redundancy is controlled by the frequency range of audible sounds. The temporal redundancy is controlled by the masking effects of the ear.

2.1 Image coding and compression

ISO has defined a digital image format known as JPEG, named after the Joint Photographic Experts Group that designed it. The image is divided into a set of 8x8 pixel blocks. One representation, called RGB, represents each pixel with three color components: red, green and blue. The pixel value is typically encoded by 24 bits and the each color is encoded by 8 bits. Another representation, called YUV, also has three components: one luminance (Y) and two chromonance (U and V). Just like RGB, YUV is a three-dimensional coordinate system. However, compared to RGB, its coordinate are rotated to better match the human visual system. The RGB components are transformed into YUV components according to:

$$Y = 0.299R - 0.587G + 0.114B \tag{2.1}$$

$$U = -0.1278R - 0.3313G + 0.5B \tag{2.2}$$

$$V = 0.5R - 0.4187G + 0.0812B \tag{2.3}$$

We can distinguish the luminance (brightness) of a pixel much better than its hue (color).

The JPEG compression takes place in four phases [55, 70], as illustrated in Figure 2.1. The first phase "downsamples" the U and V components in each block into an 4x4 block. That is, each 2x2 subblock in the original block is given by one U value and one V value – the average of the four pixel values. The subblock still has four Y values. The second phase applies the discrete cosine transform (DCT) to the YUV blocks. The YUV blocks can be seen as a signal in the spatial domain. The DCT transform the this signal into the *spatial frequency* domain. This is a lossless operation but a necessary precursor to the next, lossy step. After the DCT, the second phase applies a quantization to the resulting signal and, in doing so, loses the least significant information in the signal. The forth phase encodes the final result, but in so doing, adds an element of of lossless compression to the lossy compression achieved in the first two phases. Decompression follows the same three phases, but in reverse order. Real-time encoding/decoding of JPEG is possible in software.

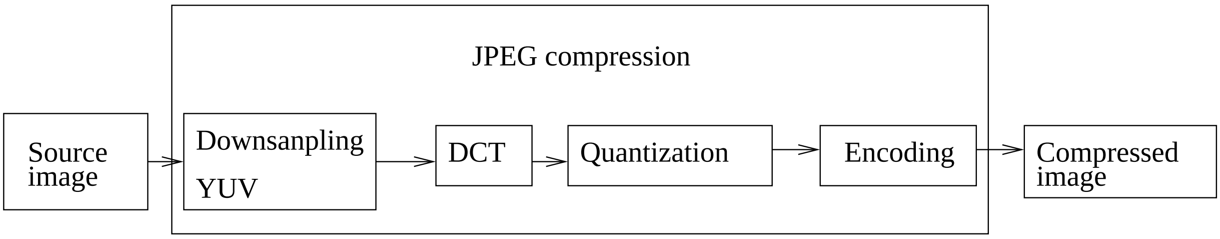


Figure 2.1: Block diagram of JPEG compression.

2.1.1 DCT phase

DCT is a transformation closely related to the fast Fourier transform (FFT). It takes 8x8 or 4x4 matrix of YUV pixel values as input and outputs an 8x8 or 4x4 matrix of frequency components. You can think of the input matrix as a 64-point (16-point) signal that is defined in two spatial dimensions (x and y); DCT breaks this signal into 64 (16) spatial frequencies. To get a intuitive feel for spatial frequency, image your self moving across a picture in, say the x direction. You would see the value of each pixel varying as some function of x. If this value changes slowly with increasing x, then it has low spatial frequency, and if it changes rapidly, it has high spatial frequency. So the low frequencies correspond to the gross features of the picture, while the high frequencies correspond to fine detail. The idea behind the DCT is to separate the gross features, which are essential to viewing the image, from the fine detail, which is less essential and, in some cases, might be barely perceived by the eye.

The DCT is defined by the formula:

$$DCT(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (2.4)$$

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \quad (2.5)$$

The inverse DCT is defined by the formula:

$$pixel(x,y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j)DCT(i,j) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (2.6)$$

The first frequency component, at location (0,0) in the output matrix, is called the *DC coefficient*. Intuitively, we can see that the DC coefficient is a measure of the average value of the 64 (16) input pixels. The other 63 (15) elements of the output matrix are called *AC coefficients*. They add the higher-spatial-frequency information to this average value. The higher frequency coefficients are increasingly unimportant to the perceived quality of the image.

2.1.2 Quantization phase

The third phase of JPEG is where the compression becomes lossy. DCT does not itself lose information; it just transforms the image into a form that makes it easier to know what information to remove. Quantization is easy to understand – it is simply a matter of dropping the insignificant bits of the frequency components.

To see how the quantization phase works, imagine that you want to compress some whole numbers less than 100, for example, 45, 98, 23, 66 and 7. If you decided that knowing these numbers truncated to the nearest multiple of 10 is sufficient for your purposes, then you could divide each number by the quantum 10 using integer arithmetic, yielding 4, 9, 3, 6, and 0. These numbers can be encoded in 4 bits rather than 7 bits needed to encode the original numbers.

Rather than using the same quantum for all 64 (16) coefficients, JPEG uses a quantization table that gives the quantum to use for each of the coefficients, as specified by the formula:

$$QuantizedValue(i,j) = IntegerRound(DCT(i,j)/Quantum(i,j)) \tag{2.7}$$

where

$$IntegerRound(x) = \begin{cases} \lfloor x + 0.5 \rfloor & \text{if } x \geq 0 \\ \lfloor x - 0.5 \rfloor & \text{if } x < 0 \end{cases} \tag{2.8}$$

Decompression is the simply defined as:

$$DCT(i,j) = QuantizedValue(i,j) \times Quantum(i,j) \tag{2.9}$$

2.1.3 Encoding phase

The final phase of JPEG encodes the quantized frequency coefficients in a compact form. This results in additional compression, but this compression is lossless. Starting with the DC coefficient in position (0,0), the coefficients are processed in the zigzag sequence shown in Figure 2.2. Along the zigzag, a form of run length encoding is used – RLE is applied to only the 0 coefficients, which is significant because many of the later coefficents are 0. The individual coefficients are then encoded using a Huffman code, which use fewer bits to represent common values than it use to represent uncommon values.

In addition, because the DC coefficient contains a large percentage of the information in the block from the source image, and images typically change slowly from block to block, each DC coefficient is encoded as the difference from the previous DC coefficient. This approach is called delta encoding.

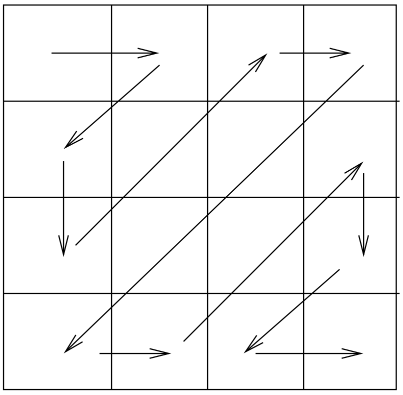


Figure 2.2: Zigzag traversal of quantized frequency coefficients for a U or V block.

2.2 Video coding and compression

2.2.1 Classification of video formats

Analog or digital video can be classified as interlaced or non-interlaced (progressive scan). Interlaced video displays the frames in two phases. The first phase displays all odd numbered lines in the frame. The second phase displays all even numbered lines in the frame. Progressive scan displays all lines in one pass from top to bottom before the next frame appears.

Video can be divided into five classes [29]:

- High Definition TV (HDTV) class
- Analog TV class
- Digital TV class
- VCR class
- Video conferencing class

The **HDTV class** offers a variety of resolutions. The highest resolution has 1920x1080 pixels per frame. The frame rate is 60 frames per second for interlaced scan and 30 frames per second for progressive scan. Uncompressed HDTV requires bandwidth exceeding 1 Gbps. MPEG compressed HDTV requires 12 to 20 Mbps.

The **digital TV class** offers a variety of resolutions. For example DVB-PAL has a resolution of 720x576 pixels per frame. DVB stands for Digital Video Broadcast which is a European standard for digital TV. PAL stands for Phase Alternating Line which is a European standard for analogue TV. The frame rate for digital TV is between 24-30 frames per second for interlaced scan and between 24-60 frames per second for progressive scan. Depending on program style and quality the bandwidth requirement after compression is between 2 to 10 Mbps.

The **analog TV class** offers a variety of resolutions, depending on the analogue TV standard. For example, the resolution is 720x576 for the PAL standard. The frame rate for PAL is 25 frames per second. Uncompressed sampled PAL requires 216 Mbps and MPEG compressed sampled PAL requires 4 to 6 Mbps. PAL uses interlaced scan.

The **VCR class** has a resolution of 352x288 and the bandwidth is about 1/4 of the PAL bandwidth.

The **video conferencing class** requires 64 kbps after compression. The resolution is about 240x360 pixels per frame and the frame rate is between 5 to 15 frames per second.

2.2.2 Introduction to MPEG

The MPEG is an ISO standard for coding/compression of digital video and audio named after the Motion Pictures Experts Group that defined it.

The first version of MPEG was called MPEG-1. Its goal was to produce VCR quality output using a bit rate of 1.2 Mbps.

The second version of MPEG was called MPEG-2. It was originally designed for compressing broadcast quality video into 4 to 6 Mbps for digital transmission. Later MPEG-2 was expanded to support higher resolutions, including HDTV.

The third version of MPEG was called MPEG-4. It was aimed at medium-resolution videoconferencing with low frame rates and at low bandwidths.

The basic principles of MPEG-1 and MPEG-2 are similar but the details are different.

To a first approximation, MPEG-2 is a superset of MPEG-1, with additional features, frame formats and encoding patterns. It is likely that in the long run MPEG-1 will dominate for CD-ROM movies and MPEG-2 will dominate for long-haul video transmission. The description below applies to both MPEG-1 and MPEG-2.

Real-time encoding and decoding of MPEG is possible with special hardware.

2.2.3 Frame types

To a first approximation, a moving picture (i.e. video) is simply a succession of still images – also called *frames* or *pictures* – displayed at some video rate. Each of these frames can be compressed using the same DCT-based technique used in JPEG. However, there is also interframe redundancy in a video sequence. For example, two successive frames of video will contain almost identical information if there is not much motion in the scene, so it would be unnecessary to send the same information twice. Even if there is motion, there may be plenty of redundancy since a moving object may not change one one frame to the next; in some cases, only its position changes. MPEG takes this interframe redundancy into consideration.

MPEG takes a sequence of video frames as input and compresses them into three types of frames, called *I frames* (intrapicture), *P frames* (predicted picture), and *B frames* (bidirectional predicted picture) [55, 70]. Each frame of input is compressed into one of these three frame types. I frames can be thought of a reference frames; they are self-contained, depending on neither earlier nor later frames. To a first approximation, an I frame is simply the JPEG compressed version of the corresponding frame in the video sequence. P and B frames are not self-contained; they specify relative differences from some reference frame. More specifically, a P frame specifies the differences from the previous I frame, while a B frame gives an interpolation between previous and subsequent I or P frames.

Not that because each B frame depends on the later frame in the sequence, the compressed frames are not transmitted in sequential order. For example, the sequence IBBPBBI is transmitted as IPBBIBB. Also, MPEG does does not define the ratio of I frames to P and B frames. However, in practice, the frames are sent in a periodic manner, defined by the deterministic Group of Picture (GOP) pattern, e.g. “IBBPBB” or “IBBPBBPBBPBB”.

2.2.4 Compression of I frames

Each raw video image is divided into a set of 16x16 macroblocks. The RGB macroblocks is transformed into a YUV macroblocks. The YUV macroblocks are encoded using JPEG which goes through the phases of downsampling, DCT transformaton, quantization and encoding. A compression ratio of 6:1 can typically be achieved for I frames.

2.2.5 Compression of P frames

A macroblock in a P frame is encoded using predictive encoding. Each macroblock in a P frame is represented with a 3-tuple: (1) a coordinate for the macroblock in the frame, (2) a motion vector relative to the previous reference frame, and (3) a delta (δ) for each pixel in the macroblock.

The first step is to determine the motion vector (a, b) using some algorithm which searches for the best matching block within a certain area. The motion vector points from the position (x, y) in the block that has moved to a point in its counterpart in the previous I frame. The delta value is computed according to:

$$\delta(x, y) = F_c(x, y) - F_p(x + a, y + b) \tag{2.10}$$

where $F_c(x, y)$ denotes the current pixel value in position (x, y), and $F_p(x + a, y + b)$ denotes the pixel value in the previous I frame.

The delta values are encoded in the same way as pixels in I frames. That is, they are run through DCT and then quantized and encoded. Since the deltas are typically small, most of the DCT coefficients are 0 after the quantization; hence they can be effectively compressed.

A macroblock in a P frame can optionally be decoded using JPEG which is useful when the motion picture is changing too rapidly.

A compression ratio of 15:1 can typically be achieved for P frames.

2.2.6 Compression of B frames

A macroblock in a B frame is encoded using bidirectional predictive encoding. A macroblock is defined with a 4-tuple: (1) a coordinate for the macroblock in the frame, (2) a motion vector relative to the previous reference frame,(3) a motion vector relative to the subsequent reference frame, and (4) a delta (δ) for each pixel in the macroblock.

The delta values are computed according to:

$$\delta(x, y) = F_c(x, y) - (F_p(x + a, y + b) + F_f(x + c, y + d))/2 \tag{2.11}$$

where (c, d) denotes the motion vector that points to a pixel in the subsequent I or P frame, and $F_f(x+c, y+d)$ denotes the pixel value in the subsequent (future) I or P frame. The delta values are encoded the same way as for P frames.

A macroblock in a B frame can optionally be decoded using JPEG which is useful when the motion picture is changing too rapidly.

A compression ratio of 120:1 can typically be achieved for B frames.

2.3 Audio coding and compression

MPEG not only defines how video is compressed, but it also defines a standard for compressing audio. This standard can be used to compress the audio portion of a movie (in which case the MPEG standard defines how the compressed audio is interleaved with the compressed video in a single MPEG stream) or it can be used to compress stand-alone audio (e.g., an audio CD). Real-time encoding and decoding of MPEG audio is possible in software but requires a fast CPU such as Pentium II.

MPEG-1 audio is the standard which defines MP3 or MPEG Layer III. MP3 provides for bit rates from 8 kbps to 320 kbps per channel with sampling frequencies of 32, 44.1 and 48 kHz. The standard enables the use of variable bit-rate coding as well as fixed bit rate coding. MP3 supports 4 different channel modes: a monotonic mode for single channel inputs, dual monotonic mode for two independent channels (for example two different languages), stereo mode for stereo channels and joint stereo mode which tries to take advantage of any cross correlation in the channels. The lower two levels I and II are less sophisticated than layer III and produce higher fidelity audio at higher bit rates. The functionality for layer I is a subset of layer II which itself is a sub set of layer III. MPEG-2 audio has support for three new sampling frequencies, 16, 22.05 and 24 kHz, in addition to the the sampling frequencies supported by MPEG-1 audio.

MPEG audio is a lossy perceptual coder [29, 55]. A perceptual coder uses the properties of the human ear to compress audio signals by removing tones that are inaudible to most people anyway. As mentioned in Chapter 1 the sound reception in the human ear can be divided into roughly 25 critical bandwidths. These bandwidths cover the frequency space from approximately 0 to 20,000 Hz. Within each of these bandwidths the strongest signals tend to overwhelm the other weaker signals and they are effectively masked. Empirical evidence shows that when these masked signals are removed there is statistically significant number of people that cannot perceive the loss of fidelity.

Figure 2.3 illustrates a block diagram structure of an MP3 coder. The raw audio signal is fed in parallel into a polyphase filter bank and to a 1024 point Fast Fourier Transform (FFT). The output of the FFT is fed into a psychoacoustic model. The output of each polyphase filter is broken into a sequence of blocks with $64 - 1024$ samples per block. Each block is transformed using a modified DCT algorithm which computes a set of DCT frequency coefficients for different frequencies. The DCT coefficients are non-uniformly quantized in the next step. Finally, the quantized coefficients are encoded using Huffman coding. The quantization and coding is done in a nested loop. The outer loop (noise control loop) changes scale factors to reduce quantization noise below the masking threshold. The inner loop (rate loop) chages the quantization step size so that full signal range can be coded.

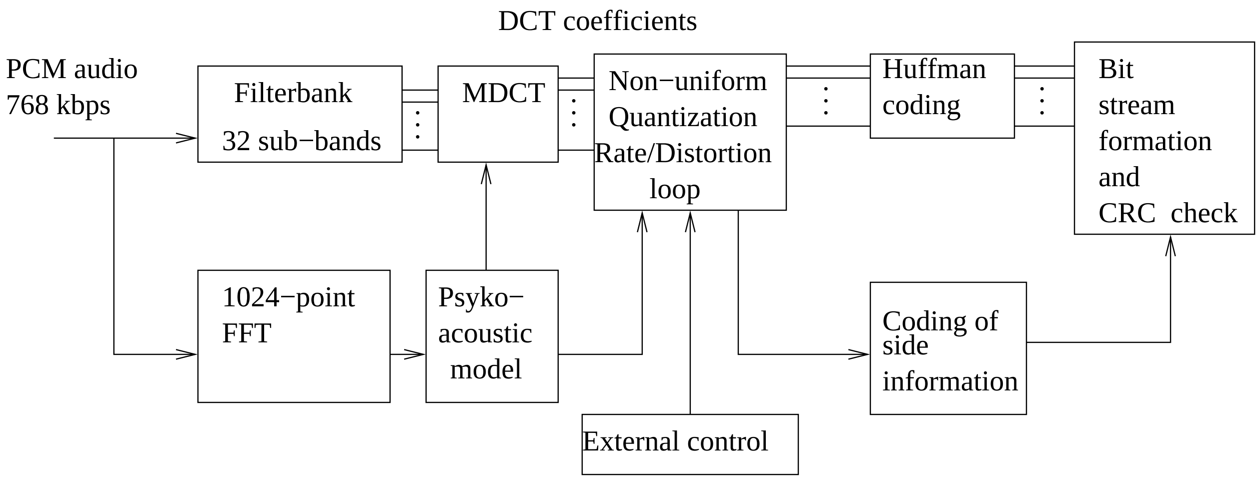


Figure 2.3: Illustration of the MP3 system

2.3.1 Polyphase filter bank

The polyphase filter bank consists of 32 polyphase filters each followed by a modified DCT. The bands can have overlapping frequencies or not, depending on implementation. The output from each of the filters is then downsampled according to the Nyquist rate (i.e. twice the maximum frequency in the signal). This is done because after dividing the signals into different bands, if one has N filters then the numbers of samples has increased by N. Therefore the output from each filter is downsampled such as the number of samples/sec. equals twice the bandwidth of the new signal. The output from each filter is grouped into a set of 12 samples. Each group of 12 samples is assigned a scale factor to normalize the samples to make full use of the quantizer later on.

Since no real filter is perfect, the cut off frequency is always higher than desired. This means your signal has greater bandwidth than it is supposed to. Therefore when you downsample, you are sampling below the Nyquist rate and aliasing (signal distortion) occurs. The purpose of using polyphase filters is to reduce this aliasing.

2.3.2 Psykoacoustic model

The Psykoacoustic model is used in parallel with the filter banks. The signal is broken up into bands roughly approximating those of the human ear. Each band is then analyzed, and recommendations concerning bit-allocations are passed along to the quantization block. For the most part, the decisions made by the psykoacoustic model are based on empirical tables buried in the encoder. For low levels of compression, the psykoacoustic model can be entirely skipped and bits can be assigned based on just the signal-to-noise ratio (SNR) of the signals coming from the polyophase filter banks.

For higher levels of compression, quantization is determined from the signal-to-mask ratio (SMR) which is the difference in sounds pressure between the masker and the masking threshold. The masking threshold is the amount of noise in each frequency band that would be masked and made inaudible by the signal energy at and around that band.

2.3.3 Quantization and coding

Quantization and coding is achieved through an iterative inner and outer optimization loop. A power law quantizer is used so that large values are coded with less accuracy, as a higher signal energy would mask more noise. The quantized values are coded with Huffman coding.

- **Inner iteration loop(rate loop)**

  The Huffman code tables assign shorter code words to (more frequent) smaller quantized values. If the number of bits resulting from coding operation exceeds the number of bits available to code a given block of data, this can be corrected by adjusting the global gain to result in larger quantization step size, leading to smaller quantized values. This operation is repeated with different quantization step size until the resulting bit demand for Huffman coding is small enough. The loop is called rate loop because it modifies the overall coder rate until it is small enough.

- **Outer iteration loop (noise control/distortion loop)**

  To shape the quantization noise according to the masking threshold, scale factors are applied to each scale factor band. The systems starts with a default factor of 1.0 for each band. If the quantization noise in a given band is found to exceed the masking threshold as supplied by the psykoacoustic model, the scalefactor for this band is adjusted to reduce the quantization noise. Since achieving smaller quantization noise requires smaller quantization steps and thus higher bit rate, the rate adjustment loop has to be repeated every time new scalefactors are used. In other words, the rate loop is nested within the noise control loop. The outer (noise control) loop is executed until the actual noise is below the masking threshold for every scale factor band.

Part II

Introduction to multimedia communication

Chapter 3

Communication principles

3.1 Communication architecture

A communication user is connected via his terminal to an access network. In local communication, users exchange information over the same access network. In remote communication, users in different access networks are connected over an core or backbone network, see Figure 3.1.

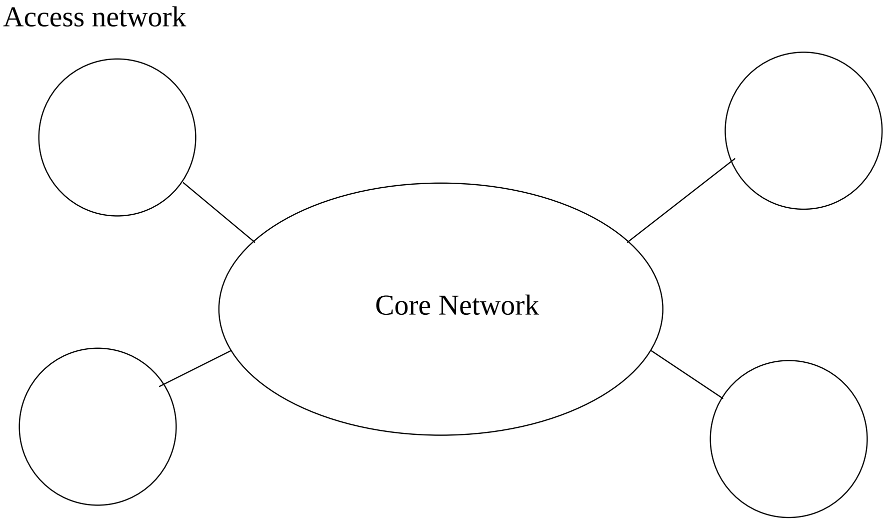


Figure 3.1: Remote communication model

The access networks are classified into wired access networks and wireless access networks.

Wired access solutions include [29]

- Analog modems
- xDSL
- Wired LANs and MANs
- Hybrid fiber coax (HFC) networks
- Fiber networks

Today the analog modem is still the most common way to access the Internet WAN network. ADSL which is an xDSL technique that has recently attracted more and more customers. Fiber networks offer the highest bandwidths but also the highest costs but they may become a cost-effective alternative in future.

Wireless access solutions include [62, 69]:

- Wireless LANs
- Cellular networks
- Satellite networks
- Broadcast systems

Wireless access networks are anticipated to become increasingly important over the next decade. An important example is 3rd generation (3G) cellular systems which are targeted for introduction in several European countries before 2005.

Tradionally, communication networks are classified according to their geographical coverage. *Local Area Networks* (LANs) are privately- owned networks within a single building or campus of up to a few kilometers in size. *Metropolitan Area Networks* (MANs) are private or public and might cover a group of nearby corporate offices or a city. *Wide Area Networks* (WANs) are normally public and spans a large geographical area, often a country or a continent.

There are five major topologies used in LANs:

- **Bus topology**: All stations are connected to a central cable, called the bus or backbone. Ethernet LAN has a bus topology.

- **Ring topology**: All devices are connected to one another in the shape of a closed loop, so that each station is connected directly to two other stations, one on either side of it. Token ring LAN has a ring topology.

- **Star topology**: All stations are connected to a central hub.

- **Tree topology**: All stations are connected to a local hub. A tree or “star of stars” topology interconnects local hubs in a hierarchy.

- **Mesh topology**: All stations are connected to a local node. The local nodes have direct links to two or more other nodes. Examples include ATM and IP LAN networks.

There are three major topologies used in MANs:

- **Bus topology**: An example is the DQDB network.

- **Ring topology**: An example is the FDDI network.

- **Mesh topology**: Examples include ATM and IP MAN networks.

There are two major topologies used in WANs:

- **Mesh topology**: Examples are fully connected telephone networks and sparsely connected ATM and IP networks.

- **Ring topology**: An example is the SDH/SONET ring network.

3.2 Single-service communication networks

Traditionally, WAN communication networks have developed along two tracks: circuit-switched telephone networks and packet-switched data networks [55, 68, 70]. In the former, communication is carried out over “circuits” with dedicated transfer capacity or bandwidth. Apart from constant throughput, circuit-switching also gives 100 % reliability (zero information loss), constant delay and zero delay variability (jitter). In the latter track, information is broken up into pieces of 10s to 1000s of bytes. Each piece is added control information which helps the packet-switching network to guide or route the packet to the appropriate destination.

At each switch, the packet may be broken up into smaller pieces which are encapsulated into data link frames, for transmission to the next switch or the terminal.

User data terminals, attached to a packet-switched network, send packets asynchronously, according to their current communication needs. Although the packets are inserted in the network on an asynchronous basis, the transmission of the packets is synchronous. A packet-switch connects multiple input links to multiple output links. Without any special mechanisms switching conflicts would arise when packets from different connections simultaneously request access to the same output link. Different solutions with input queues, central queues, and/or output queues are possible. Switch buffers have finite size and excessive periods of overload would result in "buffer overflow" forcing packets to be discarded. The queuing of packets in the switches also introduce extra delay and jitter.

The anatomy of a packet switch with input buffers and output buffers is shown in Figure 3.2.

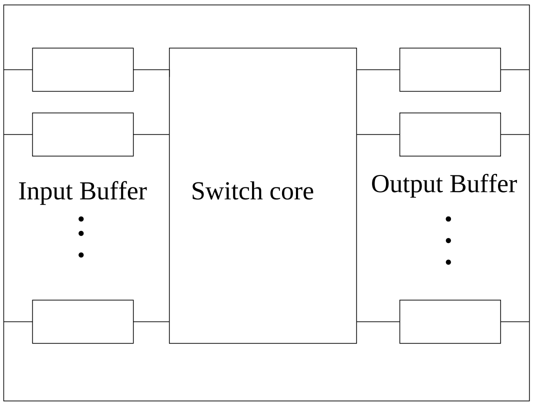


Figure 3.2: Anatomy of a packet switch/router

In connectionless information transfer, the packets are sent into the network without any prior overhead due to connection set up. Each packet can chose its own path to the destination. That is, a packet switch might forward packets with the same source and destination address to different output links. In the connectionless packet-switching mode there is not possible to obtain tight bounds on the packet loss probability, packet delay or jitter.

The global Internet is the most well known example of a connectionless packet-switched network. It is primarily used for data transport purposes such as transfer of files and world wide web documents. Internet is defined as the world-wide network of smaller networks interconnected using the Internet Protocol (IP). IP is a network protocol based on connectionless switching of variable-size packets. The packet switch in IP networks is called *router*. The main drawback of the Internet used today is that it only provides best-effort IP service. Best-effort means that the network does its best to deliver the packets to the destination host. However, there is no guarantees on certain packet delay and jitter, or to say the least, on successful packet delivery. Reliability in Internet is provided by a transport protocol called *Transmission Control Protocol* (TCP). TCP is an end-to-end protocol, only implemented at the network hosts.

In connection-oriented information transfer the communication is carried out over connections, called *virtual circuits* (VCs) in packet-switched networks. The packets sent over a VC will normally use the same path to the destination. Traditional packet-switched networks such as X.25 do not reserve any resources at VC set up. Reliability is based on error detection and error recovery. Error detection and error recovery can in general be provided on three levels: link level, network level, or end-to-end (host) level.

Error detection is based on giving each packet a sequence number, in order to detect lost packets, duplicate packets, and packets out of sequence. Bit errors can be detected by a checksum.

Error recovery is implemented by *forward error correction* (FEC) or *backward error correction* (BEC). FEC is suitable for real-time services which have no time for retransmissions. FEC is based on error correcting codes such as parity codes and Reed-Solomon codes. BEC is suitable for non-real-time services. BEC-based error recovery relies on positive and/or negative acknowledgments. Normally, a timer is set when the packet is transmitted. If the positive acknowledgment takes too long time before it reaches the sender, the timer will expire and the corresponding packet be re-transmitted. If the round trip time has a large variance compared to its mean, the timers can not be tightly set, in which case negative ackowledgements will speed up the re-transmission process.

Flow control adapts the sender's transmission rate to the available storage and processing capacity at the receiver. Error detection, BEC and flow control are often integrated.

Early LAN networks for data communication include Ethernet networks, introduced in the mid 70s. Ethernet connects stations via a shared media. The media access scheme is said to be of the broadcast type. This means that destinations can be reached without switching, by sending frames on media which all stations listens to. No real time service is provided. The users can chose between unreliable and reliable Ethernet data service. Reliability is based on error detection and BEC. Ethernet can also provide flow control.

3.3 Multi-service communication networks

From the start of the data communication era in beginning of the 70s and until the mid-80s physically separate WAN networks was used for data- and telecommunication. This was changed in 1984 when the International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), standardized the Narrowband Integrated Services Digital Network (N-ISDN). The goal of N-ISDN was to provide voice and nonvoice services over a digital circuit-switched network. N-ISDN connections can have a bit rate of 144 kbps in case of basic access (US and Europe), and 1488 kbps (US) or 1936 kbps (Europe) in case of primary access.

Broadband ISDN was standardized in 1988 by ITU-T. B-ISDN was anticipated to become an universal network providing any kind of communication service, including multimedia service. Asynchronous Transfer Mode (ATM) was chosen by ITU-T as the switching and multiplexing technique for implementing B-ISDN. ATM is based on switching of fixed-sized packets (cells) over virtual circuits. ATM cells are transferred over a synchronous (slotted) time division multiplex (TDM) fiber channels. The TDM channels have bandwidths of multiples of 155 Mbps.

In order for Internet to become a truly multi-service network suitable for the 21st century, it must be extended with real-time service capabilities. To this end, in the late 90s, the Internet Engineering Task Force (IETF) standardized two complementary QOS architectures or frameworks called Integrated Services (IntServ) and Differentiated Services (DiffServ). Both these architectures define an IP flow as a form of connectionless equivalent to a VC that carries packets with same QOS requirements between a certain origin-destination host pair. The future Internet is assumed to use ‘’route pinning”, i.e. not to change the route for successive packets within a flow unless necessary.

LAN and MAN multi-service networks were introduced in the late 80s and early 90s. Token bus LANs both provide real-time service as well as data service. The same is true for

FDDI and DQDB MAN networks. All these LANs and MANs interconnects the hosts over a shared medium.

Multi-service ATM networks, IP networks, broadcast LANs and MANs all rely on reservation of resources for real-time traffic. In broadcast LANs and MANs resources are allocated deterministically. Hence, it is possible to obtain zero frame loss, and worst case bounds on delay and jitter. In packet-switched networks such as ATM and IP networks, resource usage is based on *statistical multiplexing* or *deterministic multiplexing*. In the former, the network takes advantage of that different users will have overlapping periods of high and low bandwidth demand. The network need not to reserve capacity according to the aggregate peak demand, but to a lower demand, closer to the aggregate average bandwidth demand. Tight bounds on the packet loss probability, packet delay and jitter are possible in statistical multiplexing. In deterministic multiplexing the network allocate resources according to the aggregate peak demand of the users. The result is virtually zero packet loss and worst case bounds on delay and jitter.

Multi-service packet-switched networks will also monitor and enforce the traffic stream entering the network to make sure that the declared traffic parameters are not violated. Excessive packets will be discarded or marked as low priority. During congestion periods in the network, packet switches first drop the low priority packets.

Multi-service networks must charge the users for their use of the network in order to avoid that too many users request the best and most expensive services. It is believed that usage-based charging, in contrast to flat-rate charging, is most suitable for multi-service packet-switched networks.

3.4 Layered services and protocols

Communication in packet-switched networks is carried out by means of protocols implemented by the switching nodes and the terminals or hosts [55, 68, 70]. A *protocol* is a set of rules defining how information should be exchanged between two entities. Protocols normally are organized in a layered model. Each layer has its own protocol.The details of the protocol on a certain layer is hidden for other layers. Layer $n + 1$ use the service of the layer n immediately below it. The entities comprising the corresponding layers on different machines are called *peers*. In other words, it is the peers that communicate using the protocol. Between each pair of adjacent layers there is an *interface*. The interface defines which primitive operations and services the lower layer offers to the upper layer.

The set of layers and protocols is called *network architecture*. Neither the details of the protocol implementation nor the specification of the interface are part of the network architecture. The set of protocols used by one system with one protocol per layer is called *protocol stack*.

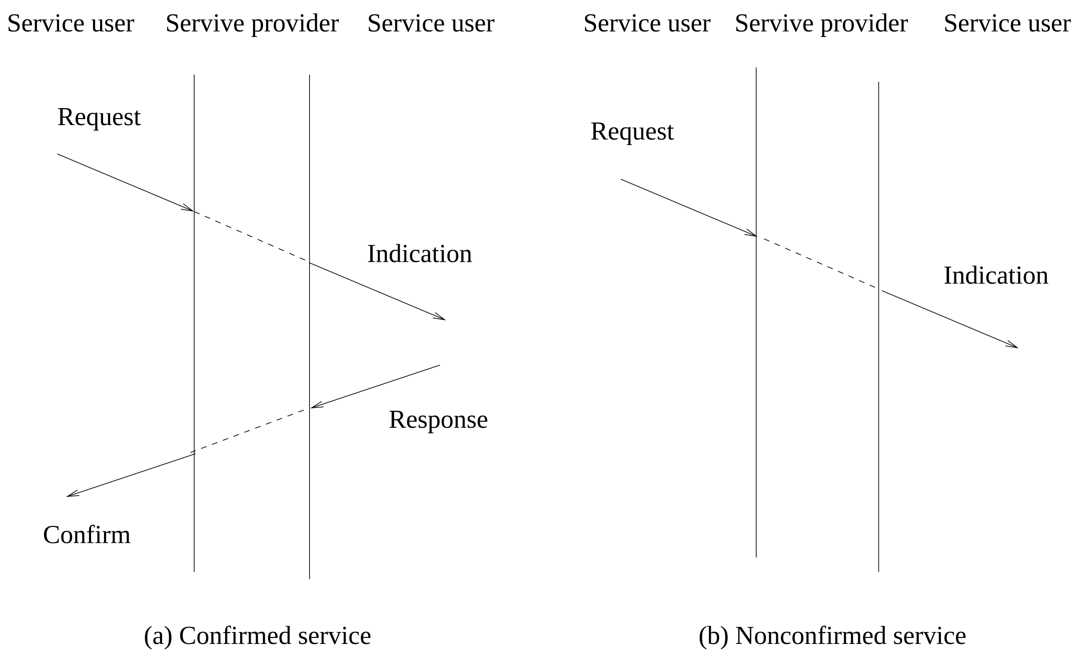
Services are available at *Service Access Points* (SAPs). Each SAP has an address that uniquely identifies it. To make things clearer, the SAPs in the telephone system are the sockets in which modular telephones can be plugged, and the SAP addresses are the telephone numbers of these sockets.

There are four basic types of service primitives:

- **Request**: A primitive issued by a service user to invoke some service and to pass the parameters needed to specify fully the requested service.
- **Indication**: A primitive issued by a service provider either to:
  1. indicate that a procedure has been invoked by the peer service user on the connection and to provide the associated parameters, or
  2. notify the service user of a provider-initiated action.
- **Response**: A primitive issued by a service user to acknowledge or complete some procedure previously invoked by an indication to that user.
- **Confirm**: A primitive issued by a service provider to acknowledge or complete some procedure previously invoked by a request by the service user.

Figure 3.3 shows the time sequence diagrams for confirmed and non-confirmed service.

The primitives invoked on layer $n + 1$ take control and data parameters as input and pass them as a layer n *Interface Data Unit* (n-IDU) to layer n. The data consists of the layer-n *Service Data Unit* (n-SDU). The control parameters are of two types. First, control parameters can be for internal layer control, e.g. the number of bytes in the n-SDU. Such control parameters are contained in the layer-n *Interface Control Information* (n-ICI). Second, control



(a) Confirmed service

(b) Nonconfirmed service

Figure 3.3: Time sequence diagrams for service primitives.

parameters can contribute to *Protocol Control Information* (n-PCI) to layer n. An example of such control parameters are the source and destination host address. The n-PCI forms together with the n-SDU form the layer-n *Protocol Data Unit* (n-PDU). The PCI are conveyed in a PDU header and/or trailer. It is used by the peer entities to carry out their peer protocol. They identify which PDUs contain data and which contain control information, provide sequence numbers, checksums and so on.

3.5 ISO OSI reference model

The International Standards Organization (ISO) proposed in 1984 a reference model for layered organization of communication protocols [55, 68, 70]. The model was called the the Open Systems Interconnection (OSI) reference model. The ISO OSI model contains seven layers, see Figure 3.5.

3.5.1 Physical layer

The physical layer is concerned with transmitting raw bits over a communication channel. The design issues large deal with mechanical, electrical, and procedural interfaces, and the

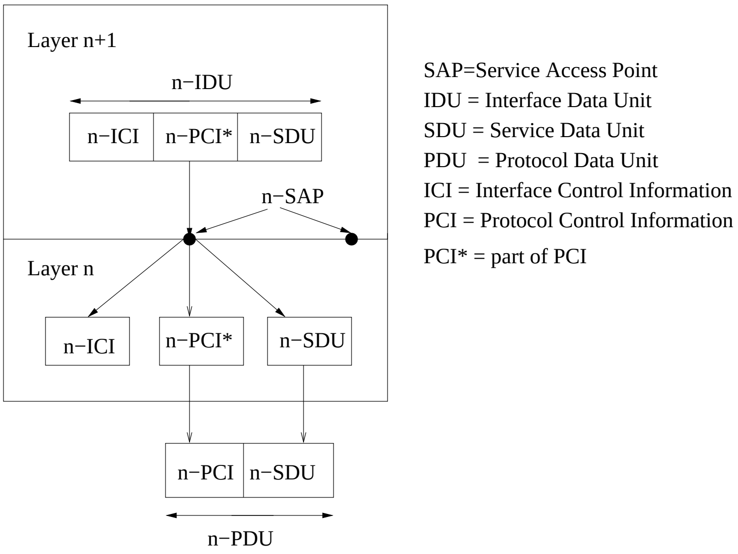


Figure 3.4: Relation between layers at an interface

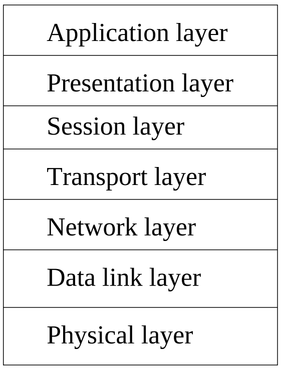


Figure 3.5: The ISO OSI reference model

physical transmission medium, which lies below the physical layer. Issues include reliable transfer of bits, bit voltage levels, time duration of a bit, how many pins the network connector has, and what each pin is used for. A final issue is security, e.g. detection of wiretapping.

3.5.2 Data link layer

The data link layer takes the raw transmission facility and transforms it into a line that appears free of undetected transmission errors to the network layer. The data link layer performs framing, which collects the bits from the physical layers into frames, typically contain 10s to 1000s of bytes. The framing can be done with a special bit sequence that detects start and end of the frame. The data link layer also deals with error detection, error recovery and flow control, and security in the form of privacy, integrity and authentification.

Broadcast networks have two additional issues in the data link layer: how to control access to a shared channel and how to address the hosts attached to the shared channel. The access control deals with issues such as bandwidth and buffer scheduling. A special sub layer of the data link layer, the medium access control (MAC) sub layer, deals with this problem.

3.5.3 Network layer

The network layer is responsible for functions such as addressing, call admission control, routing, fragmentation and reassembly, error detection, error recovery, flow control, congestion control, traffic enforcement (policing), traffic shaping, scheduling of bandwidth and buffers, network dimensioning, charging and security.

3.5.4 Transport layer

The transport layer is the first end-to-end protocol in the reference model. It is only implemented by the hosts, not inside the network. Its main purpose is to facilitate reliable transport of transport PDUs called segments between hosts. Transport protocols enhances the unreliable service of the network layer. Its functions include error detection, error recovery, flow and congestion control, user process identification, and security. Some transport protocols do not provide error recovery. If the transport connection requires high throughput, the transport layer might create multiple network connections to improve throughput. On the other hand, if creating or maintaining a network connections is expensive, the transport layer might multiplex several transport connections onto the same network connection.

3.5.5 Session layer

The session layer allows users on different hosts to establish *sessions* between them. A sessions allows ordinary data transport, as does the transport layer, but it also provides enhanced services useful in some applications. A session might be used to allow a user to log into a remote timesharing system or to transfer a file between two hosts. One of the services of the session layer is to manage *dialogue control*. Sessions can allow traffic to go in both directions at the same time, or in only one direction at a time. A related session service i *token management*. For some protocols, it is essential that both sides do not attempt the same operations at the same time. To manage between activities, the session layer provides a token that can be exchanged. Only the side holding the token may perform the critical operation. Another service is *synchronization*. The session layer may provide a way to insert checkpoints into the data stream, so that after a system crash, only the data transferred after the last checkpoint have to be repeated.

3.5.6 Presentation layer

The presentation layer is, among other things, concerned with the syntax and semantics of the information transmitted. A typical example is encoding data in standard agreed upon way. Most user programs do not exchange random binary strings. They exchange things such as people's names, dates, amounts of money, and invoices. These items are represented as character strings, integers, floating-point numbers, and data structures composed of several simpler items. In order to make it possible for computers with different representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used “on the wire”. The presentation layer manages these abstract data structures and converts from the representation used inside the computer to the network standard representation and back.

The presentation layer also is responsible for coding and compression of images, audio and video. Compression is used to reduce the bandwidth requirement of multimedia packet streams. Compression works by reducing the redundancy in the information flow.

3.5.7 Application layer

The application layer contains a variety of protocols that are commonly needed. One service is the *network virtual terminal*. Similar to having abstract data structures, a network virtual terminal is a technology independent terminal that has general functionality. Special software in the application layer maps the virtual terminal functions onto the real terminal. A second application layer series is *file transfer*. Different file systems have file naming conventions, different ways of representing text lines, and so on. Transferring a file between two systems requires handling these and other incompatibilities. Other application layer services are electronic mail, word wide web (WWW), remote job entry, directory lockup, and multimedia. Security is also an issue in the application layer, such as secure transfer of files, electronic mails and WWW documents.

Chapter 4

History of communication networks

4.1 History of telephone networks

Ever since the advent of the telephone in 1876 by Alexander Graham Bell, voice communication between distant locations have been possible. The first telephone networks introduced a few years later consisted of manually operated switching offices connected to each others and to the customers' telephones.The copper-based twisted pair was introduced in the local loop between the customer and the local switching office in the 1890s. The human operators where replaced by electro-mechanical switches in the 1940s. The advent of the transistor in 1948 paved the way for computer controlled switches which were introduced in the 1960s. Starting in the 1940s until the 1980s, the local, regional and central switches were inter-connected by coax cable. In the mid-1980s high-speed optical fibers were introduced in the switching network.

4.2 History of Internet

Paul Baran, employed at the RAND corporation in US, introduced in 1962 the concepts of packet-switching and distributed networks in his attempt to design a fault-tolerant US communication network which should survive a nuclear war. September 1969 marked the birth of ARPANET, the predecessor of Internet. ARPANET connected US universities which were supported by the US Department of Defense. In the late 70s the first version of the Internet Protocol (IP) and the Transmission Control Protocol (TCP) were introduced in ARPANET. In

1983 ARPANET became Internet. The definition of Internet being the global inter-connected collection of smaller networks which implements the TCP/IP protocol suite still holds today. The number of hosts in the Domain Name System (DNS) has evolved from 200 in 1981 to about 170 million in January 2003. The number if autonomous systems (domains) was 14,000 in January 2003, and the number of IP subnetworks was 160,000. The history of Internet applications has landmarks such as the introduction of the first email system in 1971 and the introduction of world wide web, invented at CERN, in 1991. Early attempts of video conferencing and IP telephony over Internet were carried out in the late 1990s.

4.3 History of data networks

Data networks offers a packet-oriented transport service without any guarantees on transfer delay. Historically, data network standards have evolved for both LANs and WANs. The first LAN standard, the Ethernet standard, came in 1976. Later, in 1985, the Token ring LAN standard was approved. In both these two LAN standards, stations are connected to a shared media which they access according to some control scheme. The first WAN standard, the X.25 standard, was approved by the ITU in 1976. X.25 is a packet- and connection-oriented transfer technique offering low bit rate (64 kbps) connections. Most Automatic Teller Machines (ATMs) are connected through X.25 over the D-channel using N-ISDN basic access. The Switched Multimegabit Data Service (SMDS) is a WAN network standard primary aimed at LAN connectivity. SMDS was developed by Bellcore in the early 1990s. SMDS is packet-oriented and connectionless.

4.4 History of wireless networks

The first radio transmission across Atlantic Ocean was demonstrated in 1901 by Marconi. In 1920 the first public radio transmission took place in Germany. In the 1920s police radio in cars were introduced in metropolitan New York area. In 1946 the first mobile telephone service in USA was introduced by AT&T. In 1949 Claude Shannon et al. developed the basic ideas of CDMA. In 1979 and 1981 the first generation (1G) analog mobile systems AMPS and NMT was introduced in USA and Sweden, respectively. Second generation (2G) mobile systems such as GSM, IS-95 and PDC were introduced in 1991 (Europe), 1993 (USA) and 1994 (Japan), respectively. In 2000 countries all over the world gave out licenses to network operators for operation of third generation (3G) mobile systems. European 3G mobile systems will be based on UMTS. The UMTS networks will be inter-operable with current GSM/GPRS networks. Important events in the history of wireless LANs are the introduction of the wireless Ethernet (IEEE 802.11) in 1997 and Bluetooth in 2000.

4.5 History of multi-service networks

Multi-service networks have an important advantage over pure data networks, namely real-time service capabilities. Multi-service networks have evolved since the mid 1980s and standards exist today for LANs, MANs and WANs. The Token bus LAN standard was approved in 1985. An important application environment for Token bus is factory automation. The FDDI and DQDB MAN network were developed during the early 1990s. Multi-service WANs include Frame relay, Narrowband ISDN, Broadband ISDN/ATM and the QoS enhanced Internet. The Frame relay standard was developed in 1984 and was first seen as a competitor to the X.25 standard. Frame relay is designed to be efficient for fiber communication characterized by low transmission error rates. The N-ISDN and B-ISDN are standards of the ITU approved in 1984 and 1988, respectively. N-ISDN is circuit-switched and offers bandwidths up to 2 Mbps. B-ISDN is based on ATM which is a packet-switched technology. B-ISDN offers high bit rates, starting from 155 Mbps. It is not clear when (and even if) B-ISDN will offer international connectivity. If this happens B-ISDN will be a direct competitor to the QoS enhanced Internet, which is believed to be gradually introduced in the coming decade.

Part III

Multimedia communication networks

Chapter 5

B-ISDN/ATM

In 1988 the ITU-T standardized *Asynchronous Transfer Mode* (ATM) as the network protocol for implementing *Broadband ISDN* [55, 68, 70]. The B-ISDN was anticipated as the universal network providing all kinds of communication services. However, time has shown that ATM has made small progress in delivering world-wide B-ISDN connectivity. The main use of ATM has been in private LANs and to some extent in national public WANs. ATM is also deployed in the Internet as one of the bearer services for IP. ATM cells are normally transported over SDH/SONET. The standard rate of an ATM link is 155.52 Mbps which can be provided by STS-3 and STM-1 in SONET and SDH, respectively.

5.1 ATM reference model

The ATM reference model is shown in Figure 5.1. It consists of three layers, the physical, ATM and ATM adaptation layers, plus whatever the users want to put on top of that.

Unlike the OSI reference model, the ATM model is defined as being three-dimensional. The *user plane* deals with data transport, flow control, error detection and error recovery, and other user functions. In contrast, the *control plane* handles all relevant issues on signalling. This include set-up, maintenance, and clear of calls and connections, while supporting different kinds of unicast, multicast, broadcast, and multipeer communication scenarios. The control plane also deals with negotiation and renegotiation of QoS parameters during set-up, admission control functions, ongoing QoS monitoring during the data transfer phase, and routing of set-up requests through the network. Finally, the layer and plane management

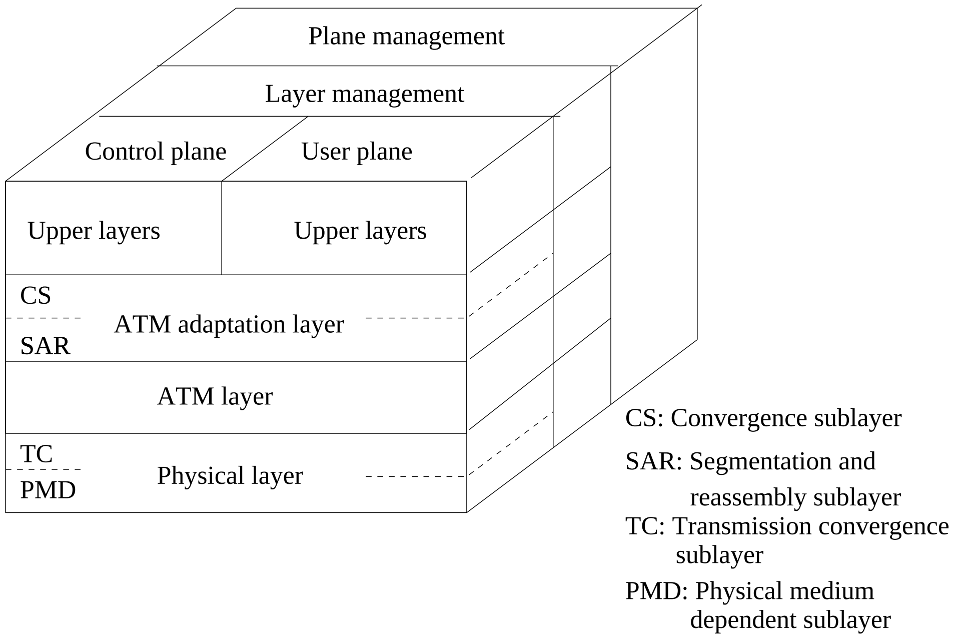


Figure 5.1: ATM reference model

functions relate to interlayer coordination.

The functionality of the different layers is summarized in Figure 5.2.

5.2 Physical layer

The physical layer deals with the physical medium: voltages, bit timing, and various other issues. ATM does not prescribe a particular set of rules, bit instead says that ATM cells may be sent on a wire or fiber themselves, but they also be packaged inside the payload of other carrier systems. In other words, ATM has been designed to be independent of the transmission medium.

The Physical Medium Dependent (PMD) sub layer interfaces the actual cable. It moves bits on and off and handles the bit timing. For different carriers and cables, this layer will be different.

The Transmission Control (TC) sub layer translates between ATM cells and strings of bits which it send and receive to/from the PMD sub layer. The TC sub layer performs framing, i.e. detects when the cell starts and ends in the bit stream.

OSI layer	ATM layer	ATM sublayer	Functionality
3/4	AAL	CS	Providing the standard interface (convergence)
		SAR	Segmentation and reassembly
2/3	ATM		Cell header generation/extraction Cell multiplexing/demultiplexing Call admission control Routing Flow control Congestion control Traffic policing Traffic shaping Network dimensioning Charging
2	Physical	TC	Cell rate decoupling Header checksum generation and verification Cell generation Packing/unpacking cells from enclosing envelope Frame generation
1		PMD	Bit timing Physical network access

Figure 5.2: The ATM layers and sub layers, and their functions

5.3 ATM layer

The ATM layer deals with cells and transport of cells. It defines a layout of a cell and tells what the header fields mean. It also deals with establishment and release of virtual circuits and management of network resources. In ATM virtual circuits are called *Virtual Channel Connections* (VCCs). By using the concept of *Virtual path* a second sub layer of processing is introduced. A *Virtual Path Connection* (VPC) is a bundle of VCCs that have the same end points. Thus, all VCCs belonging to the same VPC are switched together.

The ATM cell contains 53 bytes of which 5 bytes is header and 48 bytes is payload. The relative short packet length is motivated by the relatively short packetization delay which is required by e.g. voice services. The format of the ATM cell header is shown in Figure 5.3.

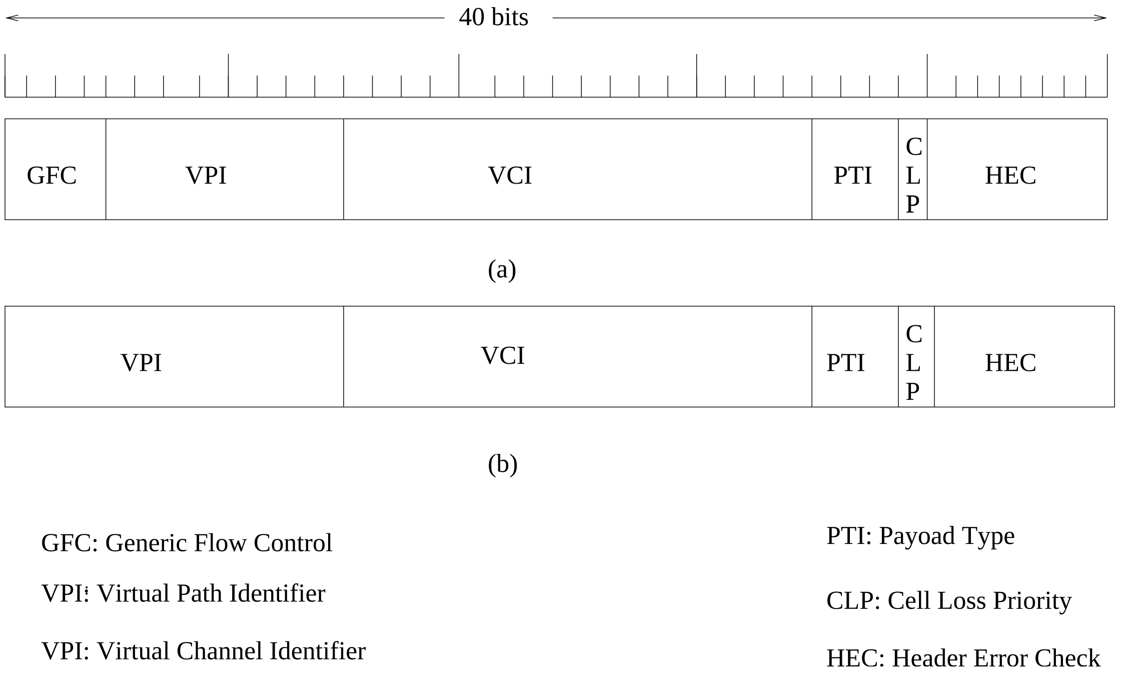


Figure 5.3: (a) ATM layer header at the UNI. (b) The ATM layer header at the NNI.

The User-Network Interface (UNI) has slightly different cell header format than the Network-Network Interface (NNI), used between switches inside the network. Depending on whether the switch is owned and located at the customer’s premises or publicly owned and operated by a telephone company, UNI and NNI can be further subdivided into public and private UNIs and NNIs. A private UNI connects an ATM endpoint and a private ATM switch. Its public counterpart connects an ATM endpoint or private switch to a public switch. A private NNI connects two ATM switches within the same private organization. A public one connects two ATM switches within the same public organization. An additional specification, the Broadband Interexchange Carrier Interconnect (B-ICI), connects two public switches from different service providers.

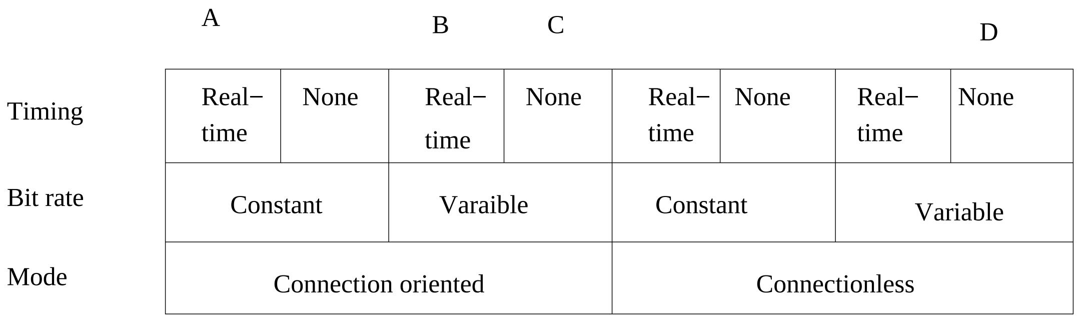
The UNI header has a *Generic Flow Control* (GFC) field which can be used to control the network access cell flow. The rest of the fields are the same in the UNI and NNI cell header. The *Payload Type Identifier* (PTI) field indicates whether the cell is a user cell, Operation and Maintenance (OAM) cell, or a Resource Management (RM) cell. The *Cell Loss Priority* (CLP) bit classifies the cell into high or low priority. The policing function can set the CLP bit to low priority (CLP=1) and these cells will be the first discarded when congestion occurs inside the network. The *Header Error Checksum* (HEC) is a checksum over the header, not the payload. The checksum can detect all single bit errors and about 90 % of all multi bit errors. The *Virtual Channel Identifier* (VCI) and the *Virtual Path Identifier* (VPI) together uniquely identifies the VC. The VCI/VPI values for a VC are local to each ATM link.

Switching of ATM cells is controlled by the VPI field (VP switching) or by the combined VCI/VPI field (VC/VP switching). The ATM switch has routing tables with entries of the form $<$ VCI/VPI in, output port, VCI/VPI out $>$. That is, the VCI/VPI value of the cell is used as an index in the routing table to look up which output port to forward the cell to. Before the ATM cell leaves the switch the contents of the VCI/VPI field is replaced by a new VCI/VPI value.

5.4 ATM adaptation layer

The ATM layer provides an ordered unreliable cell transfer service to the upper layer. All the ATM service categories can lose cells due to congestion in the switches. Moreover, cell transfer delay might vary from cell to cell (jitter) which may not be tolerable for some real-time applications. In order to improve the service of the ATM layer, the *ATM adaption layer* (AAL) is used. The AAL layer can be viewed as a form of transport layer which delivers end-to-end service. However, additional transport protocols be be used above the AAL layer, e.g. when TCP/IP or UDP/IP runs over AAL/ATM.

ITU-T has defined four ATM service classes named class A to D. The classes are characterized by different requirements on timing control, bit rate and information transfer mode, see Figure 5.4. The original idea was to give each class its own AAL protocol. However, it was soon discovered that the requirements for class C and D were so similar that AAL-3 and AAL4 were combined to AAL-3/4. Since then, another AAL protocol, has been proposed: AAL-5. It is mainly used for class C and D.



Timing	Real−time	None	Real−time	None	Real−time	None	Real−time	None
Bit rate	Constant		Varaible		Constant		Variable	
Mode	Connection oriented				Connectionless			

Figure 5.4: ITU service classes supported by AAL.

The AAL is divided into two sub layers. The upper sub layer is called *Convergence Sub layer* (CS). The lower sub layer is called *Segmentation and Reassembly* sub layer (SAR). The CS sub layer provides an interface to the application. It consists of subpart that is common to all applications (for a given AAL protocol) and an application specific subpart. The CS sub layer may add a header and/or trailer to the message received from the upper layer. The SAR sub layer breaks up the CS-PDU in smaller parts which are added a SAR header and/or trailer. The SAR-PDUs are given to the ATM layer which put each SAR-PDU in the payload of an ATM cell.

AAL-1 and AAL-5 are the most popular AAL protocols. AAL-1 is used for circuit emulation purposes suitable for uncompressed voice and video. AAL-5 is used to support IP, LAN emulation and frame relay and other network services. Even though AAL-5 does not provide any support for jitter control and FEC, it is sometimes used for compressed video (MPEG-2) applications. AAL 3/4 has been used to transport SMDS packets.

5.4.1 AAL-1

AAL-1 is the protocol used for transmitting class A traffic, that is, real-time, constant bit rate, connection-oriented traffic, such as uncompressed audio and video. Bits are fed by the application and must be delivered at the far end at the same constant rate, with low loss and a minimum of delay, jitter and overhead.

The CS sub layer of AAL-1 detects lost and miss-inserted cells. AAL-1 has FEC capability to be used e.g. for CBR video. The data is broken up into rows of 128 cells, 124 cells of data and 4 cells of Reed-Solomon error-correcting code. An entire block is made of 47 rows one for each byte of data in the AAL1 ATM cell. This type of CS sub layer can correct up to 4 lost or corrupted cells in the block. The CS sub layer also smoothes out incoming traffic, using a so-called playout buffer, to provide delivery of bits at a constant rate. The playout of bits is controlled by the receiver's service clock, which is synchronized against the source's service clock. The AAL-1 CS sub layer does not have any protocol headers of its own.

The SAR sub layer breaks up the CS PDU into 46- or 47-byte units. A 1-byte header is added to each unit. The header contains a 4-bit *Sequence Number* (SN) field and a 4-bit *Sequence Number Protection* (SNP) field.

The SN field is composed of a 1-bit *Convergence Sublayer Indicator* (CSI) subfield and a 3-bit *Sequence Count* (CS) subfield. The CSI of the odd numbered cells are used to carry time stamping information for clock recovery. The CSI of even numbered cells are used to indicate Pointer (P)- format cells. The SC field counts ATM cells in a modulo-8 fashion.

The SNP field is composed of a 3-bit *Cyclic Redundancy Code* (CRC) subfield, and a 1-bit *Parity Check* (PC) subfield. The CRC is computed over the first 4 bits in the AAL-1 header. The parity is even, computed over the first 7 bits of the AAL-1 header.

The P-format SAR PDU is used when message boundaries must be preserved. In this case, the first byte in the 47 byte payload consists of a 1-bit *Parity* field, and a 7-bit *Structure Pointer* field. The former is an even parity over the SP field. The latter contains the offset of the start of the next message. The offset is in the range 0 to 92, to put it within the payload of either its own cell or the one following it.

AAL-1 provides two ways to synchronize the ingress and egress service clocks to provide jitter-free delivery of information bits. In the *synchronous* case, the service clock is assumed locked to a common clock and its recovery is done directly from the network clock. In the *asynchronous* case, AAL-1 provides two alternatives for recovery of the clock at the receiver: the *Synchronous Residual Time Stamp* (SRTS) method and the *adaptive clock* method. In the SRTS method, absolute clock information is exchanged for synchronization, using 4 CSI bits per time stamp. This information, together with the common ATM network clock, makes it possible to reconstruct the original service clock sequence at the receiver. In the adaptive clock method, the buffer fill levels are used in order to synchronize the transmitter and receiver.

5.4.2 AAL-2

AAL-2 is the protocol for transmitting class B traffic, that is, real-time, variable bit rate, connection-oriented traffic, such as compressed audio or video. AAL-2 is still under development. One proposal is as follows.

In this proposal, the CS sub layer does not have any protocol, but the SAR sub layer does. Similarly to AAL-1 , AAL-2 also smoothes out the variability in message transfer delay. The support for clock recovery or FEC has not been specified yet.

Each SAR PDU has a 1-byte header and a 2-byte trailer. The header contains a 4-bit *Sequence Number* (SN) field, a 4-bit *Information Type* (IT) field. The IT field indicates whether the cell is the start, middle or end of a message. The trailer contains a 6-bit *Length Indicator* (LI) and 10-bit *Cyclic Redundancy Code* (CRC) field. The LI field gives the length of the payload in bytes.

5.4.3 AAL-3/4

AAL-3/4 is the protocol for transmitting class C/D traffic, that is, non-real-time, variable bit rate traffic, such as file and web transfers. AAL-3/4 can operate in two modes: stream or message. Message boundaries are preserved in message mode, but not in stream mode. In each mode, reliable or unreliable transport is available. Reliability is achieved by error detection and BEC-based error recovery. AAL-3/4 has both a CS and SAR protocol. Each CS-PDU has a 4-byte header and a 4-byte trailer.

The CS header contains the 1-byte *Common Part Indicator* (CPI) field, 1-byte *Btag* field and 2-byte BA size field. The CPI gives the message type and the counting unit for BA size and Length fields. The Btag field is used to mark the beginning of the frame. The BA size field is used for buffer allocation.

The CS trailer contains the 1-byte *Etag* field and the 2-byte *Length* field. The Etag field is used to mark the end of the frame. The Length field gives the payload length.

The SAR breaks up the CS PDU into 44 byte units and adds a 2-byte SAR header and 2-byte SAR trailer to each piece. The SAR header contain the 2-bit *Segment Type* (ST) field, the 4-bit *Sequence Number* (SN) field, the 10-bit *Multiplexing ID* (MID). The ST field is used for message framing. The MID is used when several sessions are multiplexed into one VC. The SAR trailer contains a 6-bit *Length Indicator* (LI) field, and a 10-bit *Cyclic Redundancy Code* (CRC) field.

5.4.4 AAL-5

AAL-5 is a protocol for transmitting mainly class C and D traffic. AAL-5 offers reliable or unreliable service. In addition, both unicast and multicast are supported, but multicast does not provide reliable delivery. Like AAL-3/4, AAL-5 supports message mode or stream mode.

In message mode, a trailer is added by the CS sub layer. The CS trailer contains a 1-byte *User to User*(UU) field, 2-byte a payload *Length* field, and 4-byte *Cyclic Redundancy Code* (CRC) field. The UU field is available for a higher layer for its own purposes, e.g. multiplexing. One byte in the trailer is reserved for future use.

The message is transmitted by passing it to the SAR sub layer, which does not add any headers or trailers. Instead it breaks the message into 48-byte units and passes these to the ATM layer for transmission. It also tells ATM to set the PTI field of the last cell, so message boundaries are preserved (note that this violates the layered engineering principle).

A principal advantage of AAL-5 over AAL-3/4 is much greater efficiency. While AAL-3/4 adds only 4 bytes per message, it also adds 4 bytes per cell, a loss of 8 % on long messages. AAL-5 has a slightly larger trailer per message (8 bytes) but has no overhead in each cell. The lack of sequence numbers in the cells is compensated for by the longer checksum, which can detect lost, miss-inserted, or missing cells without using sequence numbers.

5.4.5 S-AAL

S-AAL is a ATM adaption layer supporting signalling protocols such as the Q.2931/Q.298x. S-ALL is divided into a service specific CS layer and a common CS and SAR sub layer. The former is based on a Service Specific Co-ordination Function (SSCF) sub layer for UNI and

NNI, and a Service Specific Connection Oriented Protocol (SSCOP) sub layer. The SSCFs map the requirements of the layer above to the requirements of the next lower layer. The SSCOP sub layer is used for connection establishment and release, and reliable information exchange between signalling entities. The common part CS and SAR sub layer is implemented using AAL-5.

5.5 Connection control signalling

Set-up and release of VPCs and VCCs are coordinated by a special signalling protocol. ITU-T has defined the Q.2931 signalling protocol for control of point-to-point connections, and the Q.298x protocol for control of point-to-multipoint connections. ATM Forum has its own version of these protocols defined in UNI Spec 4.0.

Three types of virtual signalling channels in ATM networks are distinguished:

- Meta Virtual Signalling Channel (MVSC)
- Broadcast Virtual Signalling Channel (BVSC)
- Point-to-Point Virtual Signalling Channel (PVSC)

MVSCs are always bidirectional and are used to establish BVSCs or PVSCs as necessary. These, in turn, are use to signal all types of ATM connection signalling messages between different ATM end systems and ATM switches. BVSCs use to be unidirectional, while PVSCs are bidirectional.

Various addressing formats such as IP addressing, public ISDN E.164 addressing or OSI NSAP addressing may be used for ATM. A B-ISDN *call* consist of a number of connections between two end points. The connections may be set up and disconnected independently of one another. B-ISDN connection control messages in the Q.2931/Q.298x protocols are listed in Figure 5.5.

An example of signalling for set-up and release of a point-to-point connection is shown in Figure 5.6.

Information included in the SETUP message includes:

- Call reference

Class	Type	Q.2931/Q.298x
Point−to−point	Set−up	Alerting
		Call proceeding
		Connect
		Connect acknowledge
		Set−up
	Clearing	Release
		Relase complete
	Managing	Notify
		Status
		Status enquiry
Point−to−multipoint	Joining	Add−connection
		Connection−added
		Connection−added acknowledge
	Leaving	Relase−connection
		Relase−connection complete

Figure 5.5: Signalling messages for Q.2931/Q.298x

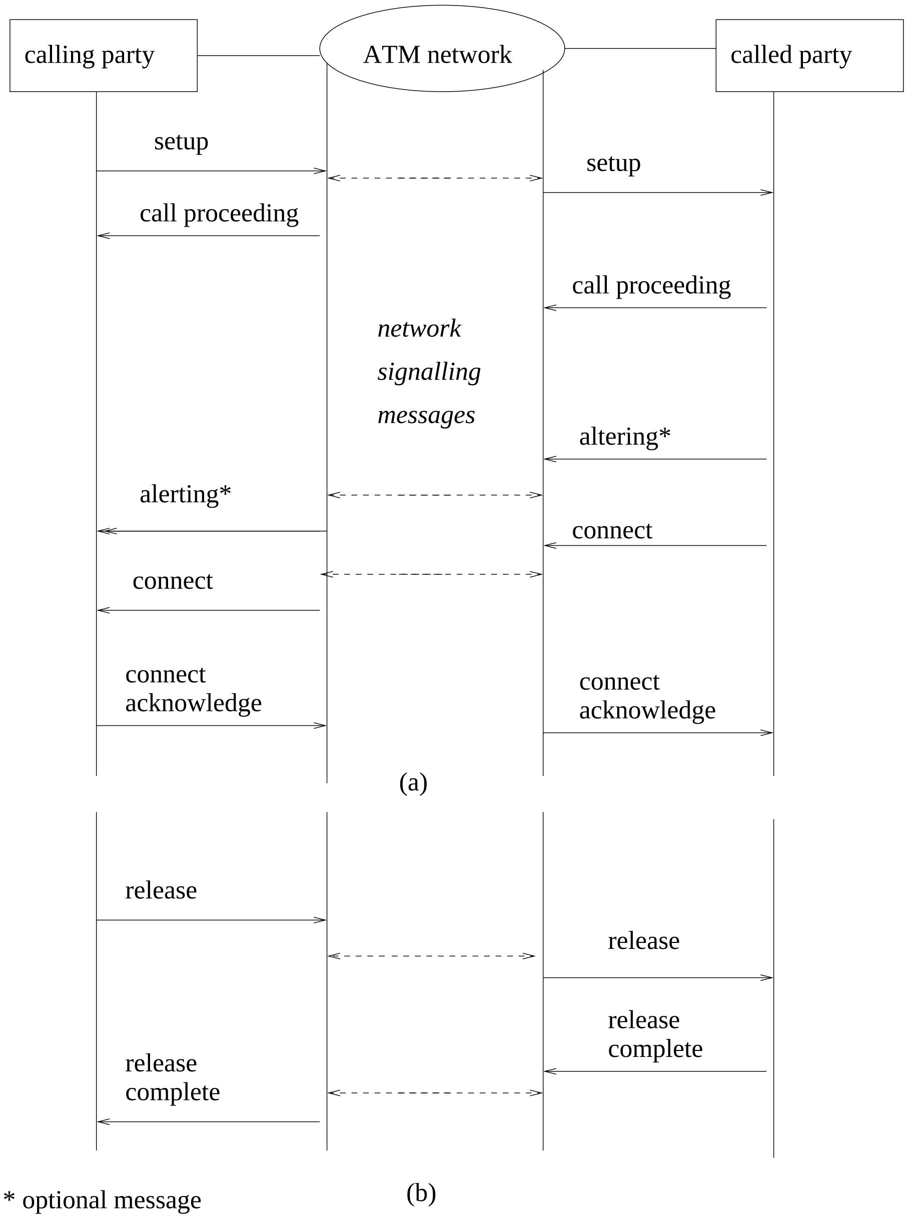


Figure 5.6: Signalling for point-to-point connection set up (a) and release (b)

- Message type: Setup indication
- Message length
- AAL parameters: AAL type, AAL parameter values
- Traffic descriptor: PCR, SCR, MBS, CDVT
- Called party number
- Calling party number
- Connection identifier: VPI/VCI
- Broadband bearer capability: point-to-point or point-to-multipoint
- QoS parameters: CLR, CTD, CDV
- Transit network selection

Chapter 6

Internet

6.1 TCP/IP reference model

The TCP/IP reference model forms the basis for the global Internet [55, 68, 70]. It was first defined by Cerf and Kahn in 1974. Compared to the ISO OSI reference model, the TCP/IP reference model contains fewer layers. The session and presentation layers are not present in the TCP/IP model. Instead these layers are incorporated in the TCP/IP application layer. The physical layer and the data link layer in the ISO OSI model are substituted with a host-to-network layer in the TCP/IP model.

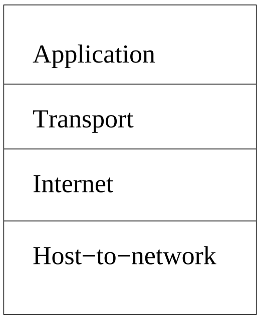


Figure 6.1: TCP/IP reference model

6.2 Host-to-network layer

The TCP/IP reference model does not real say much about what happens here, except to point out that the host has to connect to the network using some protocol so it can run IP packets over it. This protocol is not defined and varies from host to host and network and network.

6.3 Internet layer

The Internet layer provides connectionless unreliable service to the transport layer. The Internet layer defines an official packet format and a protocol called the *Internet Protocol* (IP). The only IP service provided in Internet is the best-effort service. This service gives no guarantees certain average delay before packet delivery. In fact, the packet is not guaranteed to be delivered at all.

The current version of IP in Internet today is IP version 4. IPv4 was defined by IETF in RFC 791 in 1981 [31]. The next generation of IP is version 6. IPv6 was defined by IETF in RFC 1883 in 1995 [15]. IPv6 is slowly being introduced in islands of the Internet. It will take several years before most of the routers in Internet understand IPv6.

The major differences between IPv4 and IPv6 are:

- The IPv6 packet header has a minimum of 40 bytes, the IPv4 packet header as a minimum of 20 bytes.
- The IPv6 header without extension headers contains less number of fields (7) than the IPv4 header (13). Packet processing in IPv6 can therefore be faster which improves the packet throughput.
- IPv6 has longer addresses (128 bits) than IPv4 (32 bits)
- IPv6 header has no checksum field as IPv4 has.
- IPv6 has better support for QoS than IPv4.

6.3.1 IP version 4

IPv4 provides internetworking between networks with unique network numbers. Hosts have unique host numbers within the LANs and MANs. An IPv4 address is of the form $<$

network number, host number >. IPv4 addresses are 32 bits long. Three classes of IPv4 address are used for unicast communication: class A, B and C. The classes assigns different number of bits to the network and host part of the IP address, see Figure 6.2. A forth class (D) is used for multicast purposes.

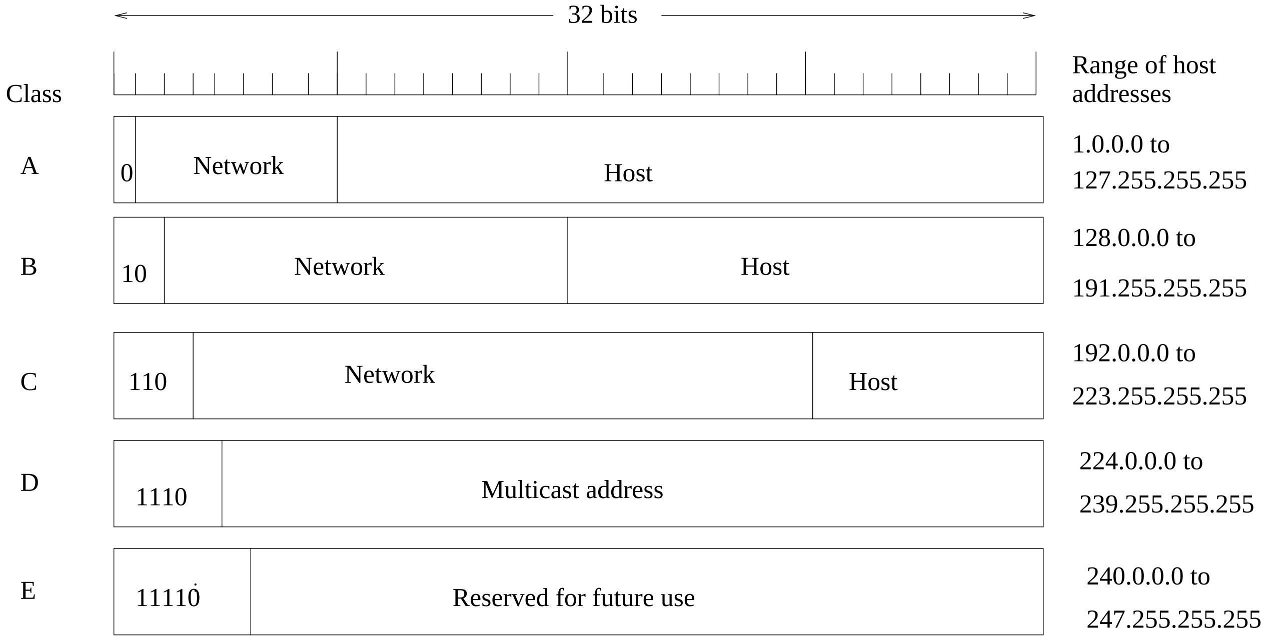

32 bits
Class
Range of host addresses
A
0
Network
Host
1.0.0.0 to 127.255.255.255
B
10
Network
Host
128.0.0.0 to 191.255.255.255
C
110
Network
Host
192.0.0.0 to 223.255.255.255
D
1110
Multicast address
224.0.0.0 to 239.255.255.255
E
11110
Reserved for future use
240.0.0.0 to 247.255.255.255

Figure 6.2: IPv4 address formats

Both IPv4 and IPv6 allows multiple IP addresses per interface. This feature is useful when several network operators provide service in the Internet. Each network operator may assign it own IP address to a host interface.

IPv4 use the concept of *subnetting* to increase the efficiency in allocation of network numbers. An owner of a LAN or MAN who wants to access the Internet is assigned a new network number from the Network Information Center (NIC). Without subnetting the network owner can only connect one network for each network number he/she receives. However, with subnetting he/she can assign one subnet number to each LAN, all which have the same network number. This is a flexible way of introducing new LANs and MANs without having to connect the NIC. Instead of having one LAN with many hosts, multiple LANs with fewer hosts can be used. Few hosts means in case of Ethernet a lower risk of frame collision and higher throughput.

In subnetting, the bits of the original host part in the IP address is divided into a subnet part and a new host part. The *subnet mask* is used to find out the network number and subnet number of the packet. The routers do a Boolean AND with the subnet mask and the IP address to get rid of the host part. The result is used to determine which interface to forward the packet to.

The *routing table* in IPv4 contains rows of the form < network number, Next hop IP address> and <subnet number, Next hop IP address>. The *forwarding table* in IPv4 contains rows of the form <network number, Interface ID, MAC address> and <subnet number, subnet mask, Interface ID, MAC address>.

The forwarding table contain the MAC address of the next router provided it is on a LAN or MAN. The forwarding table can also contain the MAC address of the destination host in case the packet has reached the last hop. To find out which MAC address a device with a given IP address has, the *Address Resolution Protocol* (ARP) is used.

Classless interdomain routing (CIDR) is a technique that has two objectives:small routing and forwarding tables, and efficient allocation of IP addresses. CIDR is defined in RFC 1519 from 1993 [24]. Recall that a class C network can contain up to 255 hosts, and a class B network can contain up to 65,535 hosts. Assume that we want to connect 1000 hosts. The normal way would be to use a class B network. With CIDR, we are instead given four continuous class C networks capable of comprising a total of 1024 hosts. The block of C networks can be identified by the common leading bits in their addresses, called prefix. This allows CIDR to reduce the size of the routing table. Only network numbers representing common prefixes is stored in the routing and fordwarding tables.

Prefixes in CIDR may contain 2 to 32 bits. Furthermore, it is possible that prefixes “overlap” in the sense that some addresses may match more than one prefix. For example, we might find both 171.69 (a 16-bit prefix) and 171.69.10 (a 24-bit prefix) in the forwarding table of a single router. In this case, a packet destined to, say 171.69.10.5 clearly matches both prefixes. The rule in this case is based on the principle of “longest match”; that is, the packet matches the longest prefix, which would be 171.69.10 in this case. On the other hand, a packet destined for 171.69.29.5 would match 171.69 and *not* 171.69.10, and in the absence of any other matching entry in the forwarding table, 171.69 would be the longest match.

IP also provides the capability to fragment the IP packets into smaller pieces. This is done when some of networks along the path to the destination has a smaller Maximum Transmission Unit (MTU) than the IP packet size. The fragments are reassembled at the next hop (router) or at the destination host. IPv4 allows each router along the way to perform fragmentation.

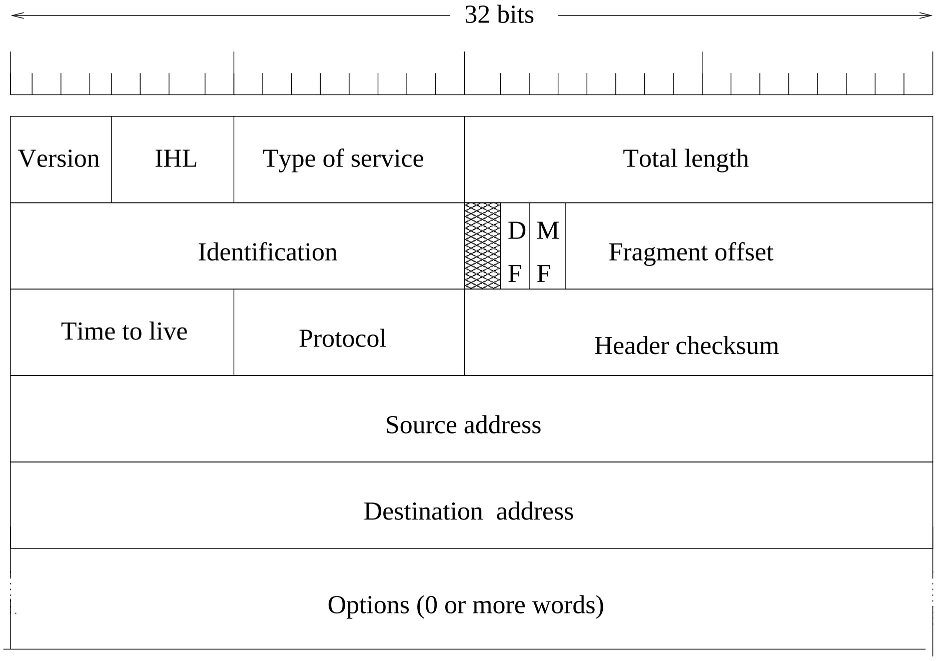
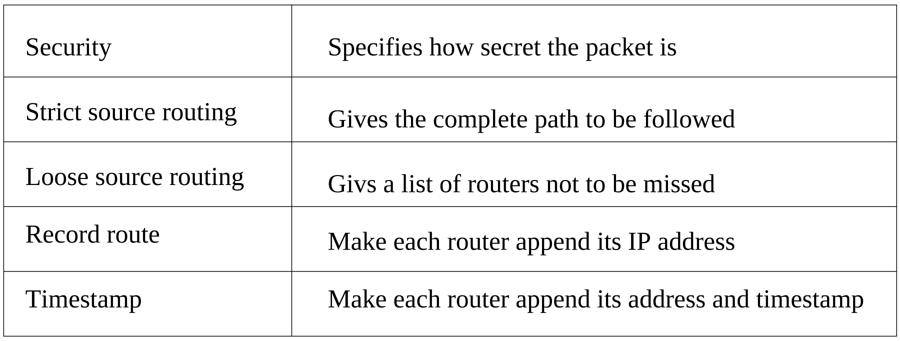


Figure 6.3: Packet header format for IP version 4

The *Version* field keeps track of which IP version the packet belongs to. The *IHL* or *IP Header Length* tells how long the header is, in 32-bit words. The minimum header length is 20 bytes, which applies when no options are present. The maximum header length is 60 bytes. The *Type of Service* field allows the host to tell the subnet what kind of service it wants. High or low reliability, throughput and delay can be specified. In addition a precedence (priority) field with eight levels can be used to differentiate between packets. In practice, current routers in Internet ignore the Type of service field. The *Total length* include everything in the packet – both header and data. The maximum length is 65,535 bytes. The *Identification* field is needed to allow the destination host to determine which packet a newly arrived fragment belongs to. All the fragments of a packet contain the same Identification value. The *DF* or *Don't Fragment* bit can be used to order the routers not to fragment the packet because the destination is incapable of putting the pieces together again. The *MF* or *More Fragments* bit is set for all but the last fragment belonging to the same packet. The *Fragment offset* tells where in the current packet this fragment belongs. A maximum of 8192 fragments can be used for a packet. The *Time To Live* field is a counter used to limit packet lifetimes. It is initialized to 255, and decremented at each hop. When it hits zero, the packet is discarded and a warning packet is sent back to the source host. The *Protocol* field tells which transport protocol is used for this packet.TCP is one possibility, but so is UDP and some others. The *Header checksum* field verifies the header only. Such a checksum is useful for detecting errors generated by bad memory words inside a router. The algorithm add up all 16-bit halfwords as they arrive, using one's complement arithmetic and then take one's complement of the result. Note that the header checksum must be recomputed at each hop, because at least one field always changes (the *Time to Live* field). The *Source address* and *Destination address* indicate the network number and host number. The *Options* field is of variable length. Each begins with a 1-byte code identifying the option. Currently five options are defined, see Figure 6.4.



Security	Specifies how secret the packet is
Strict source routing	Gives the complete path to be followed
Loose source routing	Givs a list of routers not to be missed
Record route	Make each router append its IP address
Timestamp	Make each router append its address and timestamp

Figure 6.4: IPv4 options

6.3.2 IP version 6

The *Version* field is used to arbitrate between the IPv4 and IPv6 packets. The *Priority* field is used to divide the packets into QoS classes. The *Flow label* field is still experimental but will be used to allow a source and destination to set up a pseudo-connection with particular properties and requirements. The *Payload length* field tells how many bytes follow the 40-byte header. The *Next header* field tells which of the optional extensions headers follow this header, see Figure 6.6. The *Hop limit* is used to restrict the lifetime of a packet. It corresponds to the Time to live field in IPv4. The *Source address* and *Destination address* are 128 bits (16 bytes) long. The exact use of the 128 bits has not been standardized. However, it has been suggested that the extra bits can be used to introduce new hierarchies in the address space

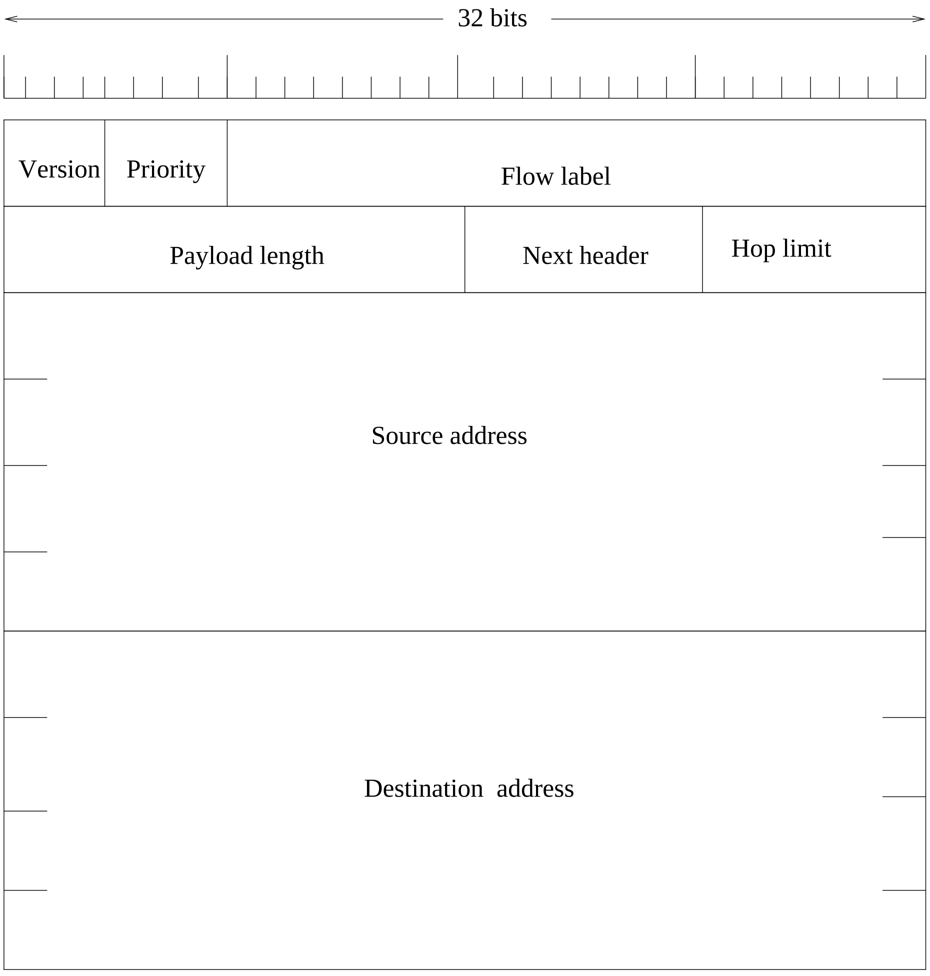
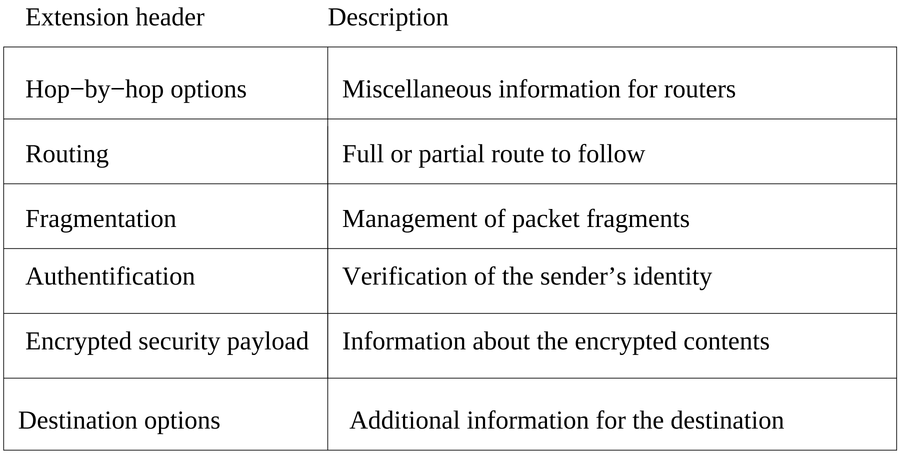


Figure 6.5: Packet header format for IP version 6

e.g. based on geographical and/or company memberships.

Apart from standard unicast and multicast IPv6 will also support a new kind of addressing: anycast. *Anycasting* identifies a group of network interfaces. A packet sent to an anycast address is sent to the interface which is "nearest" to the source, according to some distance measure.



Extension header	Description
Hop−by−hop options	Miscellaneous information for routers
Routing	Full or partial route to follow
Fragmentation	Management of packet fragments
Authentification	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents
Destination options	Additional information for the destination

Figure 6.6: IPv6 extension headers

6.3.3 MPLS

Multi Protocol Label Switching (MPLS) is a technology that integrates the label-swapping paradigm with network-layer routing. It supports various network layer protocols, including IP, and various data link layer protocols, including ATM, SDH/SONET, Frame Relay, Ethernet and Token ring.

MPLS is an advanced forwarding scheme described in RFC 3031 [60]. It extends routing with respect to packet forwarding and path controlling. Each MPLS packet has a header. In an ATM environment, the header contains only a label encoded in the VPI/VCI field of the ATM cell. In a Frame Relay environment, the header contains only a label encoded in the DLCI field of the Frame Relay data link header. In a non-ATM/Frame relay environment, the header contains a 20-bit *Label*, a 3-bit *Experimental* field (formerly know as *Class of Service* field), a 1-bit *Label stack indicator* field, and an 8-bit *Time-to-Live* (TTL) field.

A MPLS-capable router, termed *Label Switched Router* (LSR), examines the label and possibly the experimental field before forwarding the packet. At the ingress LSRs of an

MPLS-capable domain the IP packets are classified and routed based on a combination of the information carried in the IP header of the packets and the local information maintained by the LSRs. Specifically, the IP packets are mapped into a *Forwarding Equivalence Class* (FEC). The FECs are used as indexes the switching table, specifying the next hop, the label to incorporate in the MPLS header, and queuing and scheduling rules. An example of a FEC is the set of unicast packets whose destination address match a particular IP address prefix. The core LSRs use the labels of the incoming packets as indexes in the switching table to look up the next hop, the new label which should replace the current label, and queuing and scheduling rules. The egress LSR removes the MPLS header before the IP packet leaves the MPLS domain.

The path between ingress LSR to an egress LSR is called *Label Switched Path* (LSP). An LSP is similar to an unidirectional ATM VC. There are two kinds of LSPs based on the method used for determining the route: hop-by-hop routed LSPs and explicitly routed LSPs. The hop-by-hop LSPs are routed along the shortest path between edge LSRs using a standard *Interior Gateway Protocol* (IGP). Explicitly routed LSPs are routed using constrained based routing, also known as QoS routing. Constraint based routing selects routes based on multiple constraints in terms of available bandwidth, packet delay and cost among other metrics.

In order for a LSP to be set up, labels are negotiated and distributed through signalling messages that LSRs use to inform their peers of the label/FEC bindings they have made. For hop-by-hop LSP set up, this signalling information can be carried by the *Label Distribution Protocol* (LDP). For explicitly routed LSPs, two approaches have been considered by IETF: Extension of the LDP protocol or extension of the Resource Reservation Protocol (RSVP) to cope with explicitly routed LSPs.

A “tunnel” is a connection between two routers that not necessarily follow the shortest path between them. It is possible to implement a tunnel as a LSP. In fact, MPLS even supports LSP tunnels within LSP tunnels. Stacks of MPLS labels (in fact, headers) are useful for this purpose. Whenever a packet enters a tunnel on a new lower level, a new label is pushed onto the label stack. When the packet reaches the endpoint of the lower level tunnel the label is popped from the stack.

MPLS and ATM

The overlay model is a technique that was used during the later part of the 90s to circumvent some of the limitations of IP systems regarding traffic engineering. The basic idea is to introduce a secondary technology such as ATM, with VC and traffic management capability into the IP infrastructure in an overlay configuration. The VCs of the secondary technology serve as point-to-point links between IP routers.

There are fundamental drawbacks with the IP over ATM overlay model. Perhaps the most significant problem is the need to build and manage two networks with dissimilar technologies. The overlay model also increases the complexity of network architecture and network design. Scalability is an issue because the number of required *Permanent Virtual Connections* (PVCs) increases quadratically with the number of routers, thereby increasing the CPU and network resource consumption associated with routing.

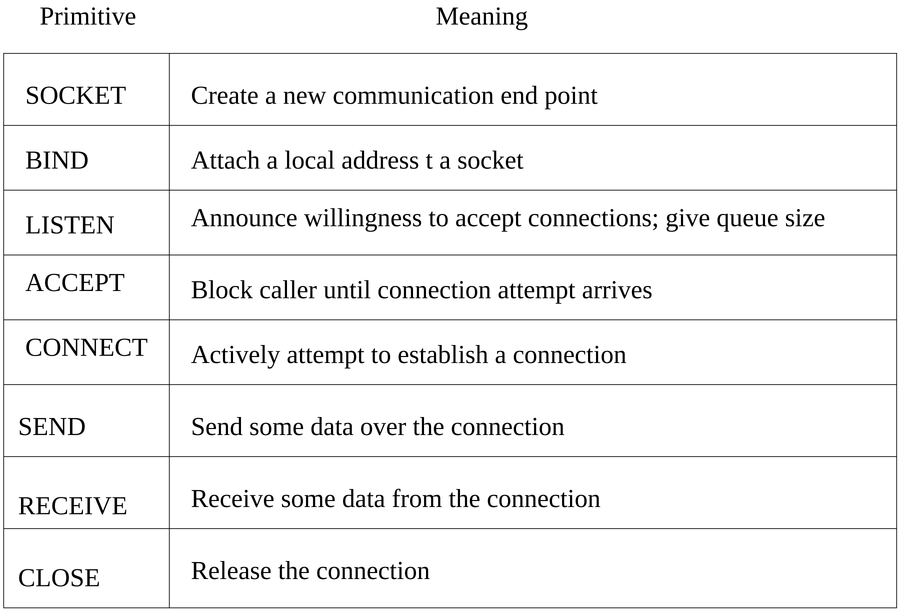
MPLS by ATM is based on dynamic LSP setup between the edge LSRs. No prior establishment of PVCs is required. The ATM switches become IGP routing peers with their neighbors. They become IGP peers by having their ATM control plane replaced with an IP control plane running an instance of the network's IGP. With the addition of LSP signalling capabilities each ATM switch becomes a core LSR, while each participating IP router becomes an edge LSR. Core LSRs provide transit service in the middle of the network, and edge LSRs provide the interface between external networks and internal ATM switched paths.

6.4 Transport layer

In the Internet three different transport protocols are mainly used: TCP, UDP and RTP. TCP is used for connection-oriented reliable transfer of transport PDUs, called segments in case of TCP, between user processes running at the hosts. The reliability is implemented by error detection and BEC-based error recovery. UDP provides connectionless unreliable transfer of datagrams between user processes. UDP also provides bit error detection through a checksum. RTP supports real-time applications such as audio and video. It provides time stamps for synchronized presentation at the destination host. RTP is connectionless and usually runs over UDP.

6.4.1 TCP protocol

The *Transport Control Protocol* (TCP) was formally defined by IETF in RFC 793 in 1981 [32]. TCP service is obtained by having both the sender and receiver create end points, called *sockets*. Each socket has a socket number consisting of an IP address of the host and a 16-bit number local to that host called a *port*. The TSAP is identified by the same information as a socket: IP address and port number. To obtain TCP service, a connection must be explicitly established between a socket on the sending machine and socket on the receiving machine. The TCP transport primitives, called socket primitives, are listed in Figure 6.7.



Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address t a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block caller until connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Figure 6.7: The socket primitives for TCP

All TCP connections are full-duplex (bi-directional) and point-to-point. TCP does not support multicasting or broadcasting. A TCP connection is a byte stream, not a message stream. Message boundaries are not preserved end-to-end.

The main objective of TCP is to enhance the unreliable service of the underlying IP protocol. TCP integrates error detection, BEC and flow control. For this purpose, it relies on the *sliding window protocol* implemented by the go-back-n protocol. When a sender transmits a segment, it also starts a timer. When the segment arrives at the destination, the receiving TCP entity sends back a segment (with data if any exists, otherwise without data) bearing an positive acknowledgment number equal to the next sequence number is expects to receive.

If the sender's timer goes off before the acknowledgment is received, the sender transmits the segment again. The TCP may also use negative acknowledgments which are used by the receiver to request missing segments.

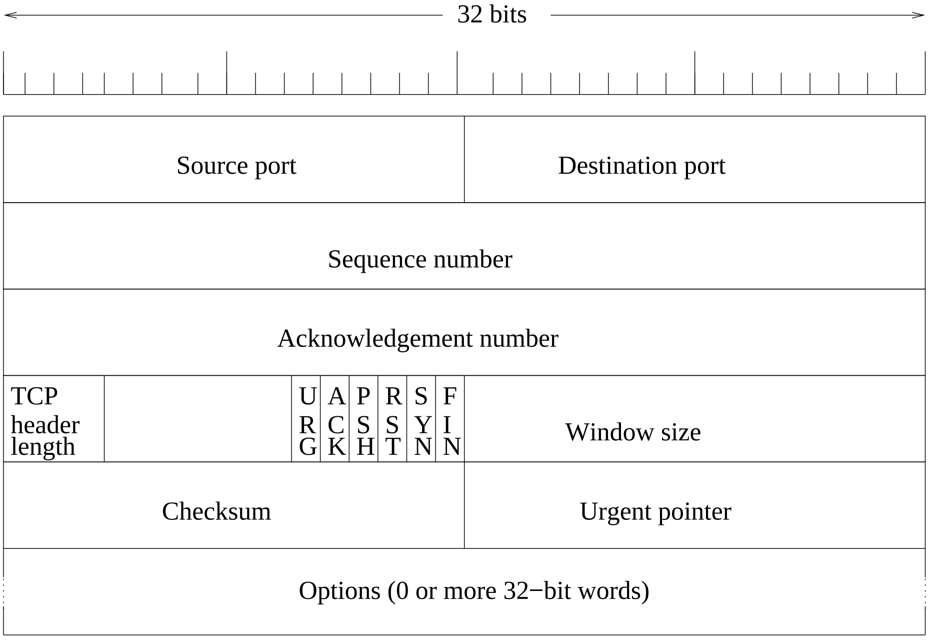


Figure 6.8: Segment header format for TCP

The *Source port* and *Destination port* fields identify the local end points of the connection. The *Sequence number* and *Acknowledgment number* fields perform their usual functions. Both are 32 bits long because every byte if data is numbered in a TCP stream. The *TCP header length* tells how many 32-bit words are contained in the TCP header. This information is needed because the TCP *Options* field is of variable length. The *URG* is set to 1 if the *Urgent pointer* is in use. The urgent pointer is a byte offset from the current sequence number at which urgent data is to be found. The *ACK* is set to 1 to indicate that the *Acknowledgment number* is valid. ACK equal to 0 means that no acknowledgment carried by the segment. The *PSH* bit indicates PUSHed data. That is, the receiver is requested to deliver the data to the application upon arrival and not to buffer until a full buffer has been received. The *RST* bit is used to reset a connection that has become confused due to a host crash or some other reason. It is also used to reject an invalid segment or to refuse an attempt to open a connection. The *SYN* bit is used to establish connections. The connection request has SYN=1 and ACK=0 to indicate that the piggy-back acknowledgment field is not in use. The connection reply does bear an acknowledgment, so it has SYN=1 and ACK=1. In essence the SYN bit is used to denote CONNECTION REQUEST and CONNECTION ACCEPTED, with the ACK bit differentiating between the two possibilities. The *FIN* bit is used to release a connection. It specifies that the sender has no more data to transmit. However, after closing a connection, a process may continue to receive data indefinitely. Both SYN and FIN segments have sequence numbers and are thus guaranteed to be processed in the correct order. The *window size* field tells how many bytes may be sent starting at the byte acknowledged. A window size field of 0 is legal and says the the bytes up to and including Acknowledgement-1 have been received, but would like no more data for the moment. Permission to send can be granted by sending a segment with the same Acknowledgement number and a nonzero window size field. A *Checksum* is also provided for extreme reliability. It checksums the header, the data, and the conceptual pseudoheader shown in Figure 6.9.The checksum algorithm is same as for IPv4. In this computation, the checksum field is set to zero, and the data field is padded out with an additional zero byte if the length is an odd number. The *Options* field provides extra facilities not provided by the regular header. The most important option is the one which allows each host to specify the maximum TCP payload it is willing to accept. The *window scale* option allows the sender and receiver to negotiate a window scale factor. This number allows both sides to shift the window size field up to 14 bits left, thus allowing windows of up to 2^{30} bytes. The *selective repeat* option allows the use of negative acknowledgments to speed up the re-transmission process.

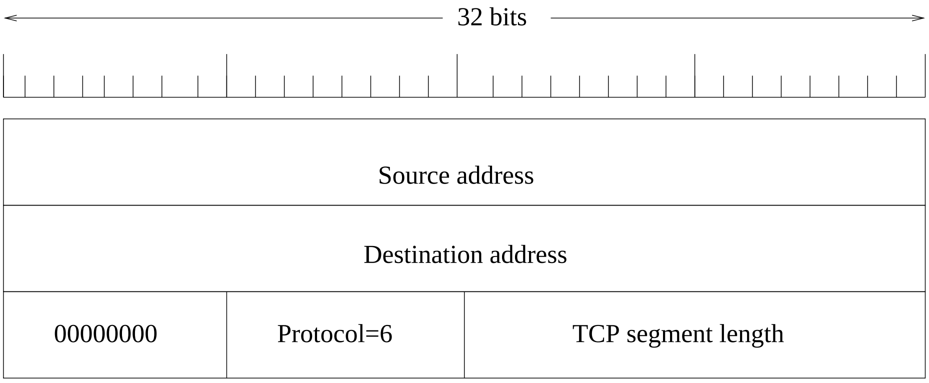


Figure 6.9: The pseudoheader included in the TCP checksum

TCP connection management

Connections are established in TCP using three-way handshake. To establish a connection, one side, say the server, passively waits for an incoming connection by executing the LISTEN and ACCEPT primitives, either specifying a specific source or nobody in particular.

The other side, say the client, executes a CONNECT primitive, specifying the IP address and port to which it wants to connect, the maximum TCP segment size it is willing to accept, and optionally some user data (e.g. a password). The CONNECT primitive sends a TCP segment with the SYN bit on and the ACK bit off and waits for a response.

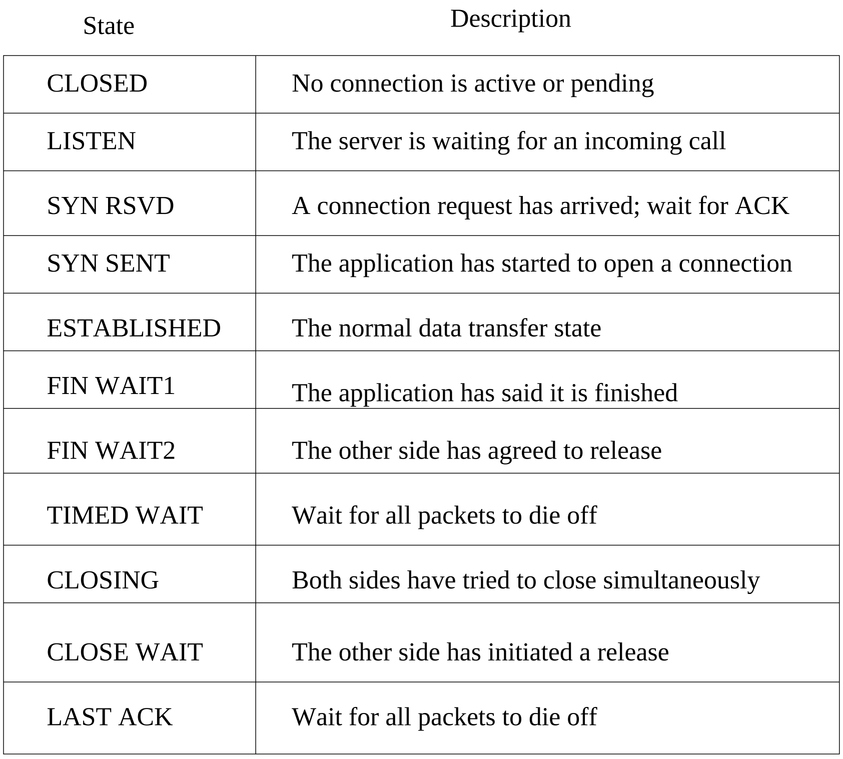
When this segment arrives at the destination, the TCP entity checks to see if there is a process that has done LISTEN on the port given in the Destination port field. If not, it sends a reply with the RST bit on to reject the connection. If some process is listening to the port, that process is given the incoming TCP segment. It can either accept or reject the connection. If it accepts, an acknowledgment segment is sent back.

Release of a TCP connection is done by independent release of the the two simplex connection making up the full duplex TCP connection. To release a connection, either party can send a TCP segment with the FIN bit on, which means that it has no more data to transmit. When the FIN is acknowledged, that direction is shut down for new data. Data may continue to flow indefinitely in the other direction. When both directions have been shut down, the connection is released. Normally, four TCP segments are needed to release a connection, one FIN and one ACK for each direction.

If a response to a FIN is not forthcoming within two maximum packet lifetimes, the sender of the FIN releases the connection. The other side will eventually notice that nobody seems to be listening to it anymore, and time out as well.

The steps required to establish and release connections can be represented in a finite state machine with 11 states listed in Figure 6.10. In each state certain events are legal. When a legal event happens, some action may be taken. If some other event happens, an error is reported. Each connection starts in the CLOSED state. It leaves the state when is does either a passive open (LISTEN), or an active open (CONNECT). If the other side does the opposite one, a connection is established and the state becomes ESTABLISHED. Connection release can be initiated by either side. When it is complete, the state returns to CLOSED.

The finite state machine is shown in Figure 6.11. The common case of a client connecting



State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RSVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT1	The application has said it is finished
FIN WAIT2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

Figure 6.10: The states used in the finite state machine

to a passive server is shown with heavy lines – solid for the client, dotted for the server. The light-face lines are unusual event sequences. Each line is marked with an *event/action* pair. The event can either be a user-initiated system call (CONNECT, LISTEN, SEND, or CLOSE), a segment arrival (SYN,FIN,ACK, or RST), or in one case, a timeout of twice the maximum packet lifetime. The action is the sending of a control segment (SYN,FIN,RST) or nothing, indicated by –. Comments are shown in parentheses.

TCP timer management

TCP uses multiple timers to do its work. The most important of these is the *retransmission timer*. When a segment is sent, a retransmission timer is started. If the segment is acknowledged before the timer expires, the timer is stopped. If, on the other hand, the timer goes off before the acknowledgment comes in, the segment is retransmitted (and the timer started again). Most TCP implementations set the timeout interval to:

$$Timeout = RTT + 4D \tag{6.1}$$

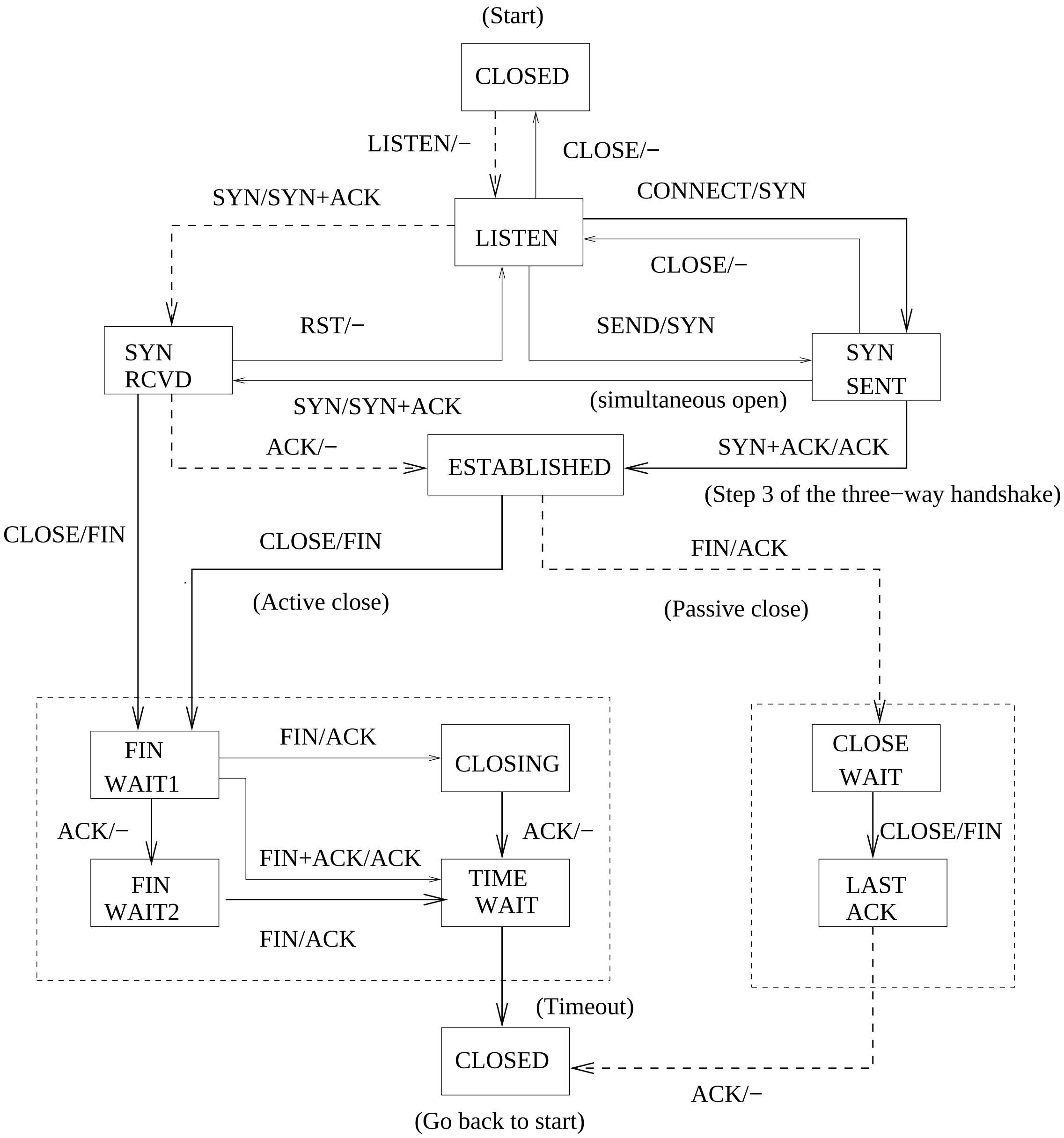


Figure 6.11: TCP connection management fine state machine

where RTT denotes average round trip time and D the standard deviation of the round trip time. RTT is given by an exponentially weighted moving average:

$$RTT = \alpha RTT + (1 - \alpha)M \tag{6.2}$$

where M denotes the measured time of one round trip and α is a constant, normally set to 7/8. D is estimated by:

$$D = \alpha D + (1 - \alpha)|RTT - M| \tag{6.3}$$

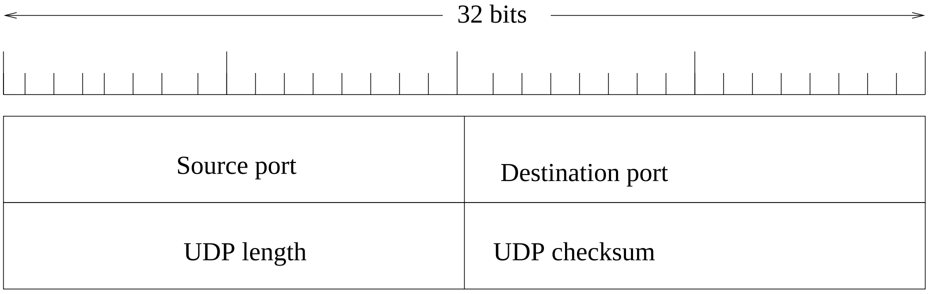
A second timer is the *persistence timer*. It is designed to prevent the following deadlock. The receiver sends acknowledgment with a window size of 0, telling the sender to wait. Later, the receiver updates the window, but the packet with the update is lost. Now both the sender and the receiver are waiting for each other to do something. When the persistence timer goes off, the sender transmits a probe to the receiver. The response to the probe gives the window size. If it is still zero, the persistence time is set again, and the cycle repeats. If it is nonzero, data can now be sent.

A third timer that some implementations use is the *keep-alive timer*. When a connection has been idle for a long time, the keep-alive timer may go off to cause on side to check if the other side is still there. If it fails to respond, the connection is terminated.

A last timer used on each TCP connection is the one used in the TIMED WAIT state while closing. It runs for twice the maximum packet life time to make sure that when a connection is closed, all packets created by it have died off.

6.4.2 UDP protocol

The *User Datagram Protocol* (UDP) was formally defined by IETF in RFC 768 in 1980 [58]. UDP provides connectionless unreliable transport of datagrams between end points (sockets).The two main functions of the UDP protocol is identification of user processes using port numbers and detection of header bit errors using a checksum. In contrast to TCP, UDP supports multicast besides normal unicast. UDP does not provide error recovery. It is up to the application layer to implement error recovery if necessary. For example, FEC-based error recovery can be used for loss sensitive real-time applications.



Source port	Destination port
UDP length	UDP checksum

Figure 6.12: Datagram header format for UDP

The *source port* and *destination port* are used identify end points within the source and destination machine. The *UDP length* field includes the 8 byte header and the data. The *UDP checksum* includes the same format pseudoheader as for TCP. The checksum is optional and stored as 0 if not computed.

6.4.3 RTP protocol

The *Real-Time Transport Protocol* (RTP) is specified in RFC 1889 from 1996 [63]. RTP runs normally over UDP. Nevertheless it is called transport protocol since it provides end-to-end service which is commonly needed by multimedia applications.The following services are provided:

- Negotiation of multimedia coding scheme
- Time stamping for jitter control and synchronization of multiple media
- Indication of congestion and packet loss
- Indication of application frames boundaries

The *Version* (V) field indicates the current version of RTP. The *Padding* (P) bit is set when the RTP payload has been padded for some reason. The *Extension* (X) bit is use to indicate that the presence of an extension header, which would be defined for a specific application and follow the main header. Such headers are rarely used, since it is generally possible to define a payload-specific header as part of the payload format definition for a particular application. The *CC* field count the number of contributing sources. The *Marker* (M) bit can be used to indicate the start of an application frame (e.g. start of a talk spurt). The *Payload Type* (PT) field indicates what type of multimedia data is carried by this packet. The *Sequence number* field is used to enable the receiver of an RTP stream to detect missing and misordered packets. The sender simply increments the value by one for each transmitted packet. It is up to the application to take actions when missing or misordered packets are discovered. The *Synchronization Source* (SSRC) uniquely identifies a single source of an RTP stream. The *Contributing Source* (CSRC) field is optional and is only used when a number of RTP streams pass through a *mixer*. A mixer can be used to reduce the bandwidth requirements for a conference by receiving data from many sources and sending it as a single stream. For example, the audio stream form several concurrent speakers could be decoded as a single audio stream. In this case, the mixer list itself as the synchronization source but also list the contributing sources – the SSRC values of the speakers who contributed to the packet in question.

A *translator* is an intermediate system that forwards RTP packets with their synchronization source identifier intact. Examples of translators include devices that convert encodings without mixing, replicators from multicast to unicast, and application-level filters in firewalls.

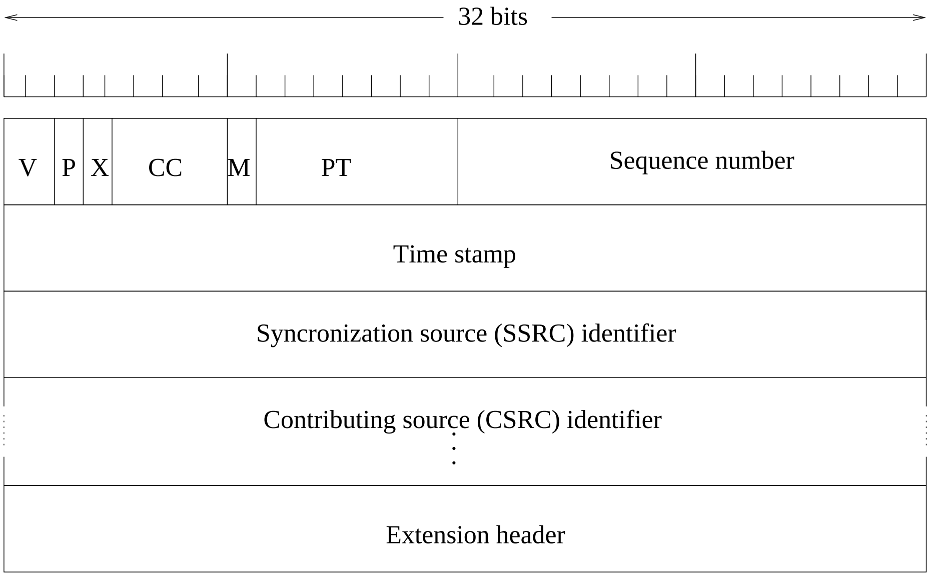

32 bits
V
P
X
CC
M
PT
Sequence number
Time stamp
Syncronization source (SSRC) identifier
Contributing source (CSRC) identifier
Extension header

Figure 6.13: Packet header format for RTP

The timestamp value in the packet is a number representing the time at which the *first* sample in the packet was generated. The timestamp is not a reflection of the time of day; only the differences between timestamps are relevant. At the sender, the timestamp is incremented with a value given by the number of samples until the next packet.

The *Real-time Transport Control Protocol* (RTCP) provides a control stream that is associated with the data stream for the multimedia application. The control stream provides three main functions:

- feedback on the performance of the application and the network
- a way to correlate and synchronize different media streams that have come from the same sender
- a way to convey the identity of a sender for display on a user interface

The first function may be useful for rate-adaptive applications, which may use performance data to decide to use a more aggressive compression scheme to reduce congestion, or to send a higher-quality stream when there is little congestion.

RTCP defines a number of different packet types, including sender reports, receiver reports and source descriptors. Reports contain statistics such as the number of packets sent, number of packets lost and inter-arrival jitter.

To reduce the risk of congestion RTCP has a set of mechanisms by which participants scale back their reporting frequency as the number of participants increases. Typically, the RTCP bandwidth is limited to 5 % of the session bandwidth, divided between the sender reports (25 %) and receiver reports (75 %).

Part IV

Service and application framework

Chapter 7

ATM service framework

7.1 ATM service architecture

ATM Forum has defined the following service categories [4]:

- Constant Bit Rate (CBR)
- Real-time Variable Bit Rate (rt-VBR)
- Non-real-time Variable Bit Rate (nrt-VBR)
- Available Bit Rate (ABR)
- Unspecified Bit Rate (UBR)
- Guaranteed Frame Rate (GFR)

Figure 7.1 summarizes what QoS and traffic contract parameters are specified for each category.

7.1.1 Constant Bit Rate (CBR) category

The CBR service category emulates circuit switching. It is indented for applications such as constant bit rate voice and video. The traffic contract specifies *peak cell rate* (PCR) and *cell delay variation tolerance* (CDVT). The CDVT specifies the maximum cell delay variation for the stream entering the UNI. The CDVT is used by the policing function. The negotiated

QoS parameters of CBR are *cell loss ratio* (CLR), *maximum cell transfer delay* (maxCTD) and *peak-to-pek cell delay variation* (peak-to-peak CDV).

7.1.2 Real-Time Variable Bit Rate (rt-VBR) category

The real-time VBR service category is intended for real-time applications, i.e. those requiring tightly constrained delay and delay variation, as would be appropriate for voice and video applications. rt-VBR connections are characterized in terms of a PCR, CDVT, Sustainable Cell Rate (SCR), and Maximum Burst Size (MBS). Sources are expected to transmit at a rate that varies with time. Equivalently the source can be described as “bursty”. The negotiated QoS parameters of rt-VBR are CLR, maxCTD and peak-to-peak CDV. Cells that are delay beyond the value specified by maxCTD are assumed to be of significantly reduced value to the application. rt-VBR service may support statistical multiplexing of real-time sources.

7.1.3 Non-Real-Time VBR (nrt-VBR) category

The non-real-time VBR service category is intended for non-real-time applications which have bursty traffic characteristics and which are characterized in terms of PCR, CDVT, SCR and MBS. For those cells which are transferred within the traffic contract, the application expects a low CLR. No delay bounds are associated with this service category. nrt-VBR service may support statistical multiplexing of connections.

7.1.4 Available Bit Rate (ABR) category

The ABR service category is intended for data traffic with requirements on loss probability but not on delay. PCR and CDVT are part of the service contract, along with with a *minimum cell rate* (MCR). The *allowed cell rate* (ACR) of an ABR traffic source therefore takes on values in the range MCR $\leq$ ACR $\leq$ PCR. The ACR is periodically updated by the network’s congestion control mechanism. Initially, ACR is set to the *initial cell rate* (ICR).

7.1.5 Unspecified Bit Rate (UBR) category

The UBR service category is designed for best-effort data transfer without tight constraints on loss or delay. No bandwidth is reserved at for UBR calls. However, the PCR and CDVT may be used for admission control and policing.

7.1.6 Guaranteed Frame Rate (GFR) category

The GFR service category is intended to support non-real-time applications. It is designed for applications that may require a minimum rate guarantee and can benefit from accessing additional bandwidth dynamically available in the network. It does not require adherence to a congestion control protocol. The service guarantee is based on AAL-5 PDUs (frames) and, under congestion conditions, the network attempts to discard complete PDUs instead of discarding cells without reference to frame boundaries. On the establishment of a GFR connection, the end-system specifies a PCR, and a Minimum Cell Rate (MCR) that is defined along with a Maximum Burst Size (MBS) and a Maximum Frame Size (MFS). The GFR traffic contract can be specified with an MCR of zero. The user may always send cells at a rate up to PCR, but the network only commits to carry cells in complete frames at MCR. Traffic beyond MCR will be delivered within the limits of available resources. There are no delay bounds associated with this service class.

	CBR	real−time VBR	non−real−time VBR	ABR	UBR	GFR
CLR Cell Loss Ratio	specified				unspecified	specified
CTD Cell Transfer Delay	specified		unspecified			
CDV Cell Delay Variation	specified		unspecified			
Traffic descriptors (service contract)	PCR/ CDVT	PCR/CDVT SCR/MBS		PCR/CDVT MCR/ACR	PCR/CDVT	PCR/CDVT MCR/MBS MFS
Congestion control	no			yes	no	

Figure 7.1: Service category attributes specified by the ATM Forum

7.2 ATM Quality of Service

The following QoS parameters are negotiated [4]:

- Maximum Cell Transfer Delay (maxCTD)
- Peak-to-peak Cell Delay Variation (peak-to-peak CDV)
- Cell Loss Ratio (CLR)

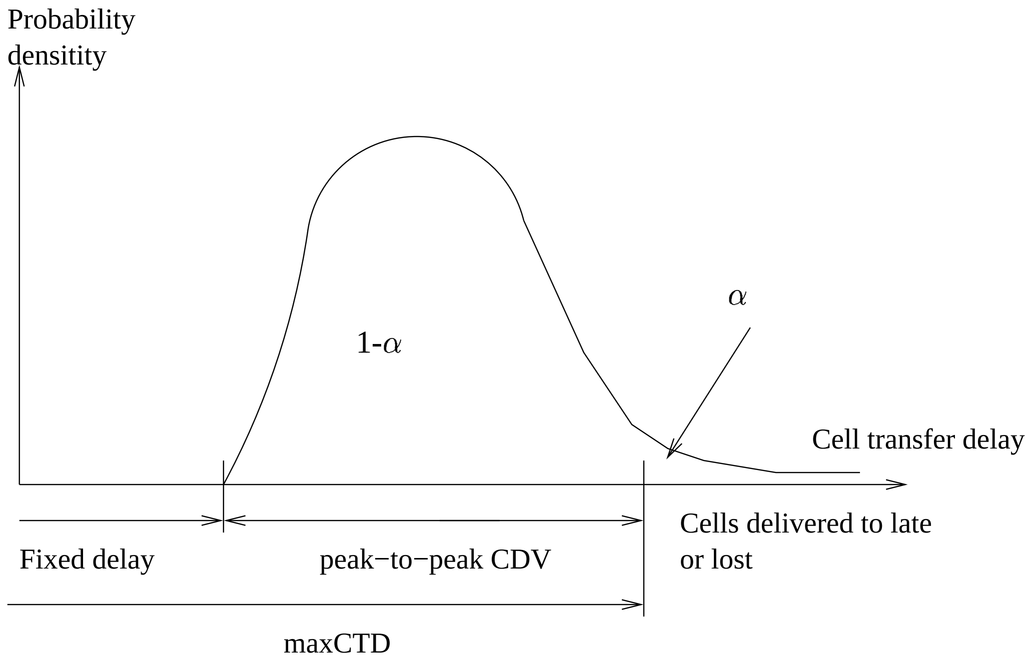


Figure 7.2: Cell transfer delay probability density model

The maxCTD is defined as the (1-α) quantile of the CTD. The peak-to-peak CDV is defined as the difference between maxCTD and the minimum CTD. The definitions of maxCTD and peak-to-peak CDV are illustrated in Figure 7.2. The CLR is defined as:

$$\text{CLR} = \frac{\text{Lost Cells}}{\text{Total Transmitted Cells}} \tag{7.1}$$

The following QoS parameters are not negotiated [4]:

- Cell Error Ratio (CER)
- Severely Errored Cell Block Ratio (SECBR)

- Cell Misinsertion Rate (CMR)

The CER is defined as:

$$\text{CER} = \frac{\text{Errored Cells}}{\text{Successfully Transferred Cells} + \text{Errored Cells}} \tag{7.2}$$

The SEBCR is defined as:

$$\text{SEBCR} = \frac{\text{Severly Errored Cell Blocks}}{\text{Total Transmitted Cell Blocks}} \tag{7.3}$$

A cell block is a sequence of N cells transmitted consecutively on a given connection. A severely errored cell block outcome occurs when more than M errored, lost or misinserted cell outcomes are observed in a received cell block. For practical measurement purposes, a cell block will normally correspond to the number of user information cells transmitted between successive OAM cells.

The CMR is defined as:

$$\text{CMR} = \frac{\text{Misinserted Cells}}{\text{Time Interval}} \tag{7.4}$$

Cell misinsertion on a particular connection is most often caused by an undetected error in the header of a cell being transmitted on a different connection. This performance parameter is defined as a rate (rather than the ratio) since the occurrence of misinserted cells is independent of the number of transmitted cells received on the corresponding connection.

7.3 ATM traffic contract

7.3.1 Traffic parameters

A traffic parameter describes an inherent characteristic of a traffic source. It may be quantitative or qualitative. Traffic parameters include Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), Maximum Burst Size (MBS), Minimum Cell Rate (MCR), and Maximum Frame Size (MFS) [4].

7.3.2 Traffic contract specification

A traffic contract specifies the negotiated characteristics of a connection. The traffic contract at the public UNI consist of a set of traffic parameters and a set of QoS parameters for each direction of the connection. The private UNI may optionally support the same traffic contract as the public UNI or a different traffic contract.

For CBR, rt-VBR, nrt-VBR, and UBR, a conformance definition based on the Generic Cell Rate Algorithm (GCRA) is used to unambiguously specify the conforming cells of a connection at the UNI. For ABR, the conformance definition refers to the behavior specified for ABR sources, destinations, and switches, but allows for delays between the source and the UNI, which may perturb the traffic flow. For GFR, the conformance definition includes a GCRA and other considerations.

The conformance definition should not be interpreted as the policing algorithm. The network is free do use any policing algorithm as long as the operation of the policing does not violate the QoS objectives of compliant connections.

The values of the traffic contract parameters can be specified either explicitly or implicitly. A parameter value is explicitly specified when its value is assigned by the end-system using signalling for SVC, or when it is specified by the Network Management System (NMS) for PVCs. A parameter value specified at subscription time is also considered to be explicitly specified. A parameter value is implicitly specified when its value is assigned by the network using default rules, which in turn depend on the information explicitly specified by the end-system.

7.3.3 Cell Delay Variation Tolerance (CDVT) for PCR and SCR

ATM layer functions (e.g. cell multiplexing) may alter the traffic characteristics of connections by introducing Cell Delay Variation. When cells from two or more connections are multiplexed, cells of a given connection may be delayed while cells of another connection are being inserted at the output of the multiplexer. Similarly, some cells may be delayed while physical layer overhead or OAM cells are inserted. Consequently with reference to the peak emission interval T (i.e. the inverse of the contracted PCR), some randomness may affect the inter-arrival time between consecutive cells of a connection as monitored at the UNI (private or public). The upper bound on the “clumping” measure is the CDVT.

Similarly, with reference to the sustained emission interval T$_s$ (i.e. the inverse of the contracted SCR), some randomness may affect the inter-arrival time between consecutive cells of a connection at the UNI (private or public).

CDVT is not signaled. In general, CDVT need not have a unique value from a connection. Different values may apply at each interface along the path of a connection.

7.3.4 Generic Cell Rate Algorithm (GCRA)

The GCRA is used to defined conformance with respect to the traffic contract. For each cell arrival, the GCRA determines whether the cell conforms to the traffic contract of the connection [4].

The GCRA is a virtual scheduling algorithm or a continuous-state Leaky Bucket Algorithm as defined by the flowchart in Figure 7.3. The GCRA is defined with two parameters: the Increment (I) and the Limit (L). The I and L parameters need not be restricted to integer values. GCRA(I, L) denotes the GCRA algorithm with increment parameter I and limit parameter L.

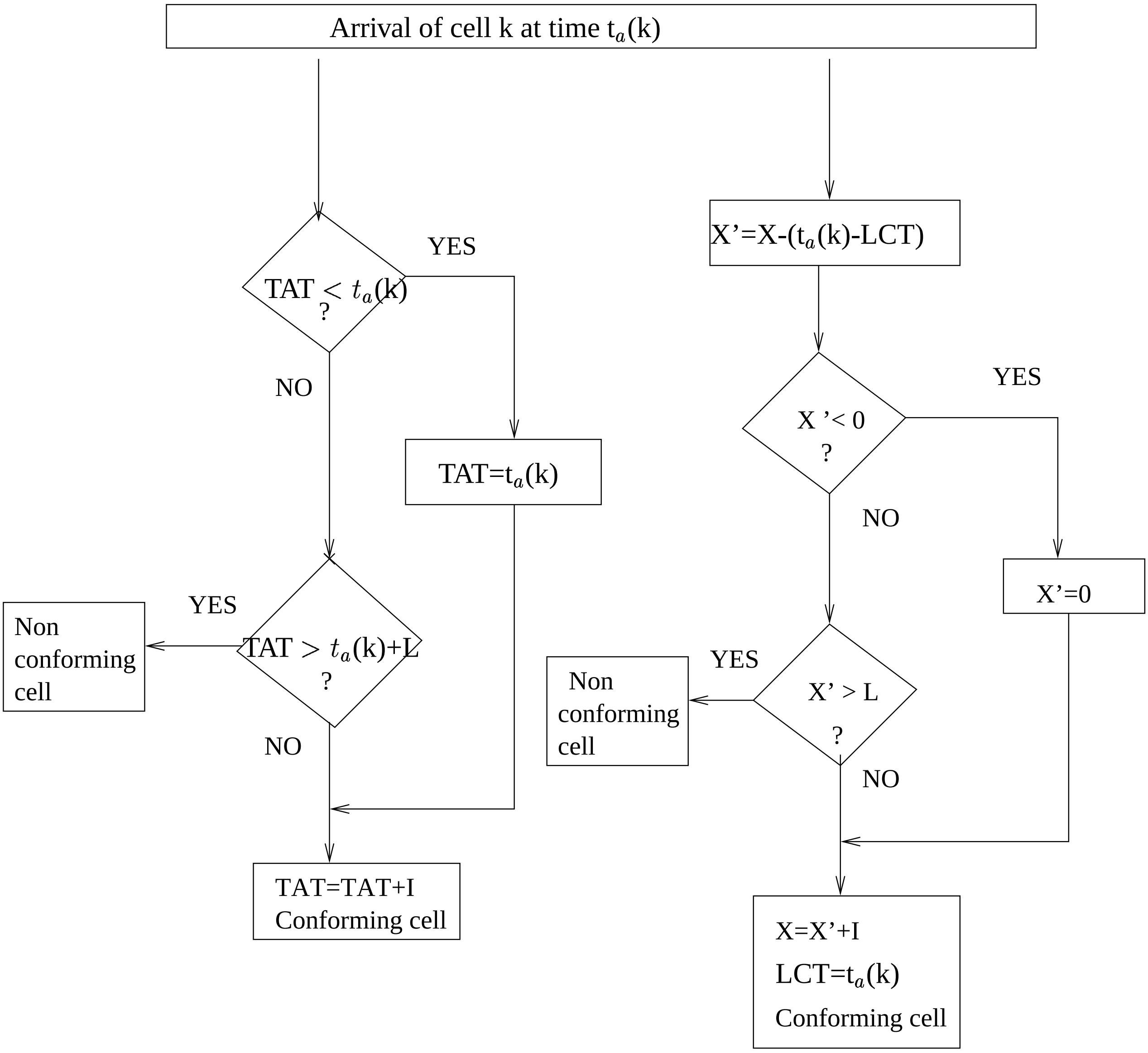
7.3.5 Traffic enforcement for CBR and UBR

The PCR is enforced by GCRA(T,CDVT), where T=1/PCR denotes the cell inter-arrival time when the source is sending at peak rate. The CDVT parameter specifies the amount of CDV the flow entering the UNI can have.

7.3.6 Traffic enforcement for rt-VBR and nrt-VBR

For PCR is enforced by GRCA(T,CDVT). The MBS and SRC are enforced by GCRA(T$_s$,BT+CDVT), where T$_s$=1/SCR denotes the cell inter-arrival time when the source is sending at SCR, and BT denotes the burst tolerance time. The MBS is the maximum number of consecutive cells that a source can send at the peak rate. The MBS is a function of PCR=1/T, SCR=1/T$_s$ and BT=τ_s:

$$MBS = \lfloor 1 + \frac{\tau_s}{T_s - T} \rfloor \tag{7.5}$$



VIRTUAL SCHEDULING ALGORITHM

CONTINUOUS LEAKY BUCKET ALGORITHM

TAT : Theoretical Arrival Time
t_a(k): Time of arrival of cell

X: Value of the Leaky Bucket counter
X': auxilary varaible
LCT: Last Conformance Time

At the time of arrival t_a of the first cell of the connection, TAT=t_a(1)

At the time of arrival t_a of the first cell to the connection, X=0 and LCT=t_a(1)

Figure 7.3: Equivalent versions of the GCRA algorithm

where $\lfloor x \rfloor$ denotes the integer part of x.

7.3.7 Traffic enforcement for ABR

A modified version of the GCRA algorithm, called Dynamic GCRA, is used to enforce the cell flow of an ABR connection. The DGCRA differs from the GCRA algoritm primarily in that the increment I changes with time, as determined by ABR feedback information conveyed on the corresponding backward connection. The DCGRA checks the conformance of all CLP=0 cells on the ABR connection. The increment I may change on the arrival of any CLP=0 cell on the connection. The increment calculated on the arrival of the k^{th} CLP=0 cell on the connection is called $I(k)$. The DGCRA algorithm with parameters traffic parameters PCR, MCR, ICR, and τ_1, τ_2, τ_3 is denoted DGCRA(1/MCR,1/PCR,1/ICR, τ_1, τ_2, τ_3). The τ_1 is the CDVT for the ABR connection. The τ_2 and τ_3 are the upper and lower bounds on the delay after which the rate change induced by a backward RM cell departing from an interface (in the backward direction) is expected to be observed at the interface (in the forward direction).

7.3.8 Traffic enforcement for GFR

For a frame to be deemed QoS Eligible, every cell must pass the cell conformance test and the frame must pass F-GCRA.

Cell conformance

A frame is conforming if all its cells are conforming, and is non-conforming if one or more of its cells are non-conforming. A user cell is conforming if all the following three conditions are met:

- The cell conforms to GCRA(1/PCR, CDVT), where PCR is defined for the CLP=0+1 cell stream.

- The CLP bit of the last cell has the same value as the CLP bit of the first cell of the frame.

- The cell either is the last cell of the frame or the number of cells in the frame up to and including this cell is less than MFS.

The GCRA test is applied to every cell and the GCRA algorithm is thus updated for every cell that conform to the GCRA test.

F-GCRA

F-GCRA is a modified GCRA test that is used by the network to identify conforming frames that should be eligible for service guarantees. There are two reasons for a frame failing F-GCRA:

- The level of the “leaky bucket” could be above its limit or the first cell could have CLP=1
- Overflowing of the bucket occurs because of too many cells arriving at a high rate. For example, the frame inter-arrival time can be too short as compared to the contract, or the traffic was sent at PCR for longer than MBS.

Chapter 8

IP service framework

8.1 IP service architecture

The ITU-T has defined six IP QoS classes in recommendation Y.1541 [34]. Figure 8.1 summarizes what QoS and traffic contract parameters are specified for each category.

- Class 0 is intended for real-time, jitter sensitive applications with high degree of interaction such as voice and video teleconferencing. The QoS parameters include IP packet loss ratio (IPLR), IP packet error ratio (IPER), IP packet transfer delay (IPTD), the IP packet delay variation (IPDV).

- Class 1 is intended for similar applications as class 0. However, the degree of interaction is not as high as in class 0. The QoS parameters specified for this class are IPLR, IPER, IPTD and IPDV.

- Class 2 is intended for transaction data and highly interactive applications. Signalling is an application example. The QoS parameters specified for this class are IPLR, IPER and IPTD.

- Class 3 is intended for similar applications as class 2. However, the degree of interaction is not as high as in class 2. The QoS parameters specified for this class are IPLR, IPER and IPTD.

- Class 4 is intended for applications which only requires low loss. Examples include short transactions, bulk data and video streaming. The QoS parameters specified for this class are IPLR, IPER and IPTD.

- Class 5 is intended for traditional applications of default IP networks. No QoS parameters are specified for this class.

	QoS classes					
QoS parameter	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5
IPTD	100 ms	400 ms	100ms	400 ms	1 s	U
IPDV	50 ms	50 ms	U	U	U	U
IPLR	$1 * 10^{-3}$	$1 * 10^{-3}$	$1 * 10^{-3}$	$1 * 10^{-3}$	$1 * 10^{-3}$	U
IPER	$1 * 10^{-4}$	$1 * 10^{-4}$	$1 * 10^{-4}$	$1 * 10^{-4}$	$1 * 10^{-4}$	U

Figure 8.1: IP QoS class definitions

8.2 IP Quality of Service

Four QoS parameters are defined for the IP QoS classes defined by ITU-T:

- IP packet transfer delay (IPTD)
- IP packet delay variation (IPDV)
- IP packet loss ratio (IPLR)
- IP packet error ratio (IPER)

The IPTD is defined as the mean IP packet transfer delay. The IPDV is defined exactly as peak-to-peak CDV for ATM connections, with the quantile 1-10^{-3}. The IPLR and IPER are defined as their counterparts in ATM.

8.3 IP traffic contract

8.3.1 Traffic parameters

The traffic parameters for a general IP source are the peak byte rate (PBR), mean byte rate (MBR), maximum burst size (MBS) in bytes, minimum packet size (m) in bytes and maximum packet size (M) in bytes.

8.3.2 Traffic contract specification

The IP traffic contract is called Service Level Agreement (SLA). An SLA is a bi-lateral agreement between two operators or an operator and a service customer. A SLA is defined as a service contract between customer and a service provider that specifies the forwarding service a customer should receive. A Service Level Specification (SLS) is the technical part of the SLA. The SLA/SLS is negotiated before service begins and specifies the traffic parameters and the QoS parameters of the flow.

8.3.3 Enforcement by the token bucket algorithm

The traffic parameters of the IP source are enforced at the ingress node of the IP domain by a token bucket algorithm [46], see Figure 8.2. Two buckets are used:the first is used to enforce the MBR and MBS, and second is used to enforce the PBR. The buckets are filled with tokens at a characteristic constant rate, given by the MBR and PBR for the first and second bucket, respectively. A control step is associated with each bucket. A packet is only allowed to pass the control step if the bucket contains at least as many bytes worth of tokens as the packet size. If the packet is allowed to pass, the level of the token bucket is reduced with a number of bytes given by the packet size. After a packet has passed the first bucket it is placed in a FIFO queue prior to the second control step. The queue is served when the second bucket contains at least as many tokens as the size of the first packet in the queue. When the packet is allowed to pass the second control step, the level of the second bucket is reduced by the size of the passing packet. A packet which has passed the second control step may enter the network.

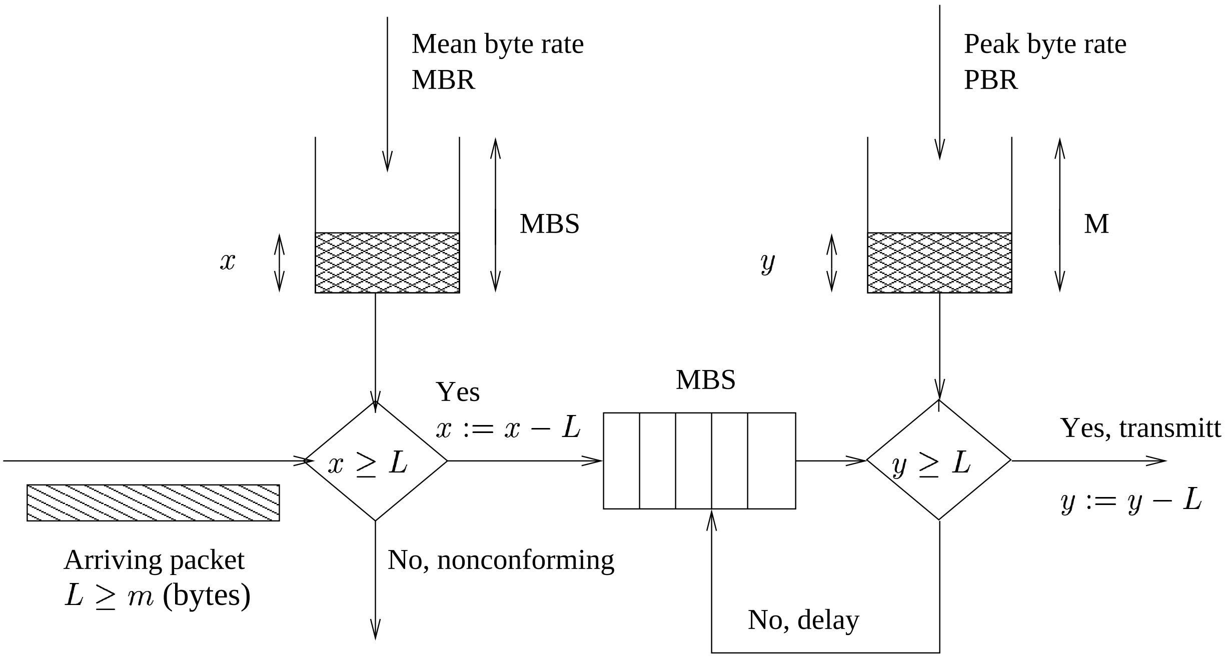


Figure 8.2: Token bucket algorithm

Chapter 9

Multimedia application framework

9.1 Classification

The ITU-T recommendation I.211 specifies two main service categories [33]: interactive services and distribution services. The interactive services are further classified into:

- Conversational services
- Messaging services
- Retrieval services

The distribution services are further classified into:

- Distribution services without user individual presentation control.
- Distribution services with user individual presentation control.

Conversational services: Conversational services in general provide the means for individual communication with real-time (no store-and-forward) end-to-end information transfer from user to user or between user and host (e.g. for data processing). The flow of the user information may be bidirectional symmetric, bidirectional asymmetric and in some specific cases (e.g. such as video surveillance), the flow may be unidirectional. The information generated by the sending user or users, and is dedicated to one or more communication partners at the receiving side. Examples of broadband conversational services are videotelephony, video conference and high speed data transmission.

Messaging services: Messaging services offer communication between individual users via storage units with store-and-forward, mailbox and/or message handling (e.g. information editing, processing and conversion). Examples of broadband messaging services are message handling services and mail services for moving pictures (films), high resolution images and audio information.

Retrieval services: The user of retrieval services can retrieve information stored in information centers provided for public use. This information will be send to the user on his demand only. The information can be retrieved on an individual basis. Moreover, the time at which the information sequence is to start is under the control of the user. Examples of broadband retrieval services are retrieval services for film, high resolution image, audio information, and archival information.

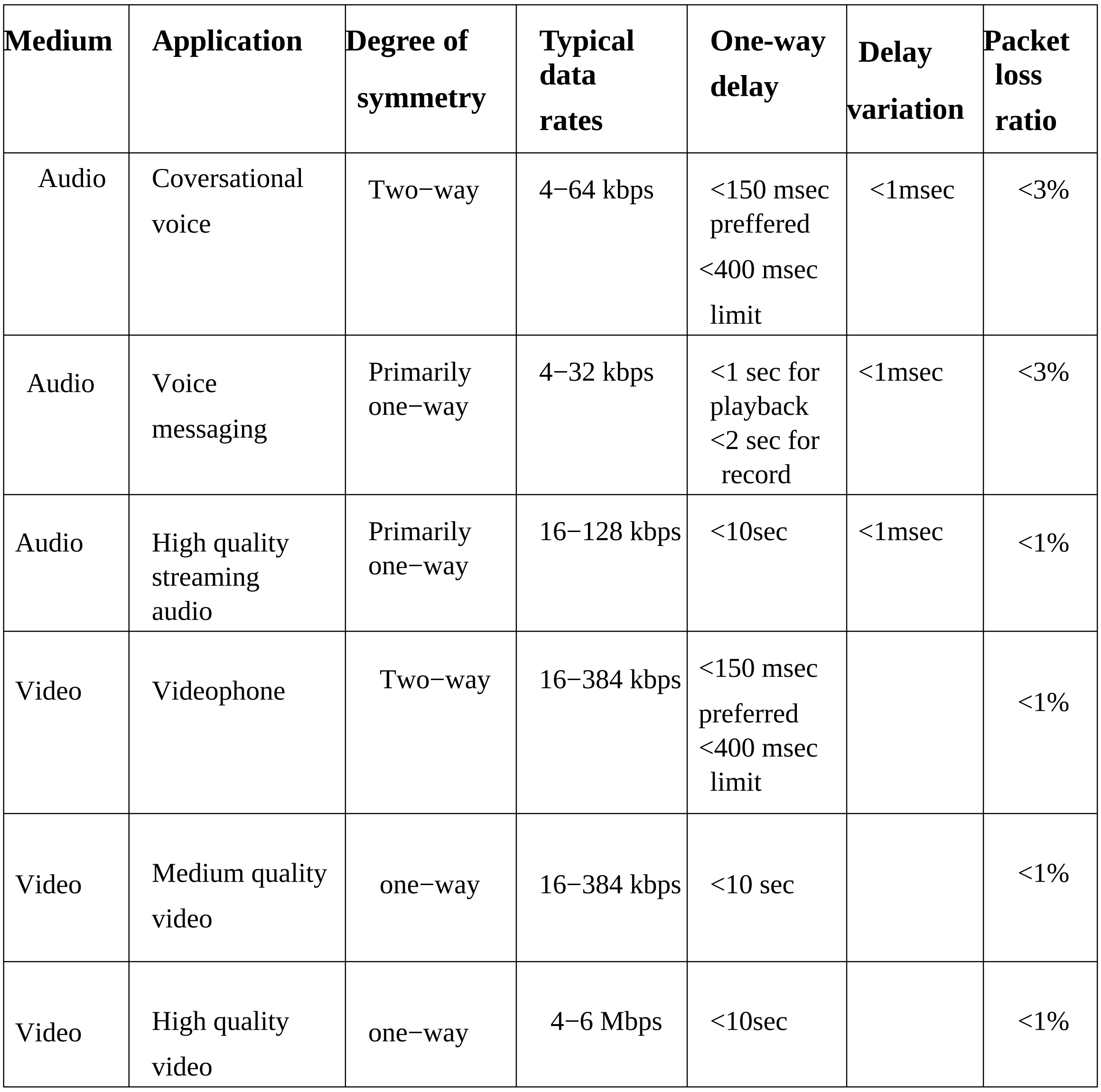
Distribution services without user individual presentation control: These services include broadcast services. They provide a continuous flow of information which is distributed from a central source to an unlimited number of authorized receivers connected to the network. The user can access this flow of information without the ability to determine at which instant the distribution of a string of information will be started. The user cannot control the start and order of presentation of the broadcasted information. Depending on the point of time of the user's access, the information will not be presented from the beginning. Examples are broadcast services for television and audio programmes.

Distribution services with user individual presentation control: Services in this class also distribute information from a central source to a large number of users. However, the information is provided as a sequence of information entities (e.g. frames) with cyclic repetition. So, the user has the ability of individual access to the cyclical distributed information and can control start and order of presentation. Due to the cyclical repetition, the information entices selected by the users will always be presented from the beginning. One example of this service is full channel broadcast videography.

9.2 QoS requirements

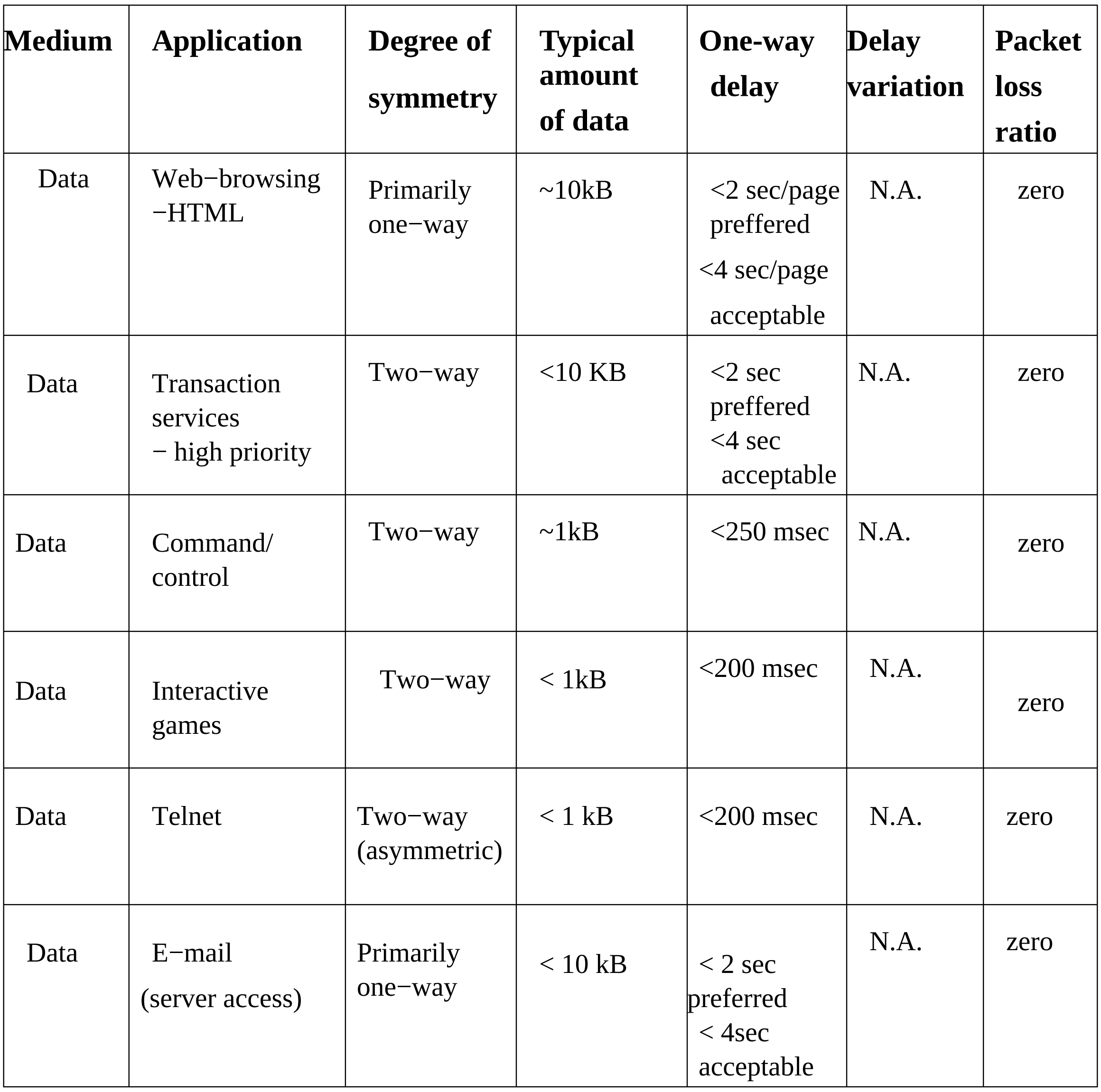
In this section we present target QoS parameter values for applications based on transfer of audio, video and data. The QoS parameters are packet delay, delay variation (jitter) and packet loss ratio. An additional QoS constraint is *lip synchronization* in video telephony. There are two synchronization cases. Either the audio comes before the video or the video comes before the audio. The first case is more severe. The time difference must be below 20 ms in the first case and 80 ms in the second case.

Figure 9.1 to 9.3 shows the numerical QoS values and Figure 9.4 presents a graphical summary. The presentation is based on the ITU draft recommendation G.QoSrqt [35].



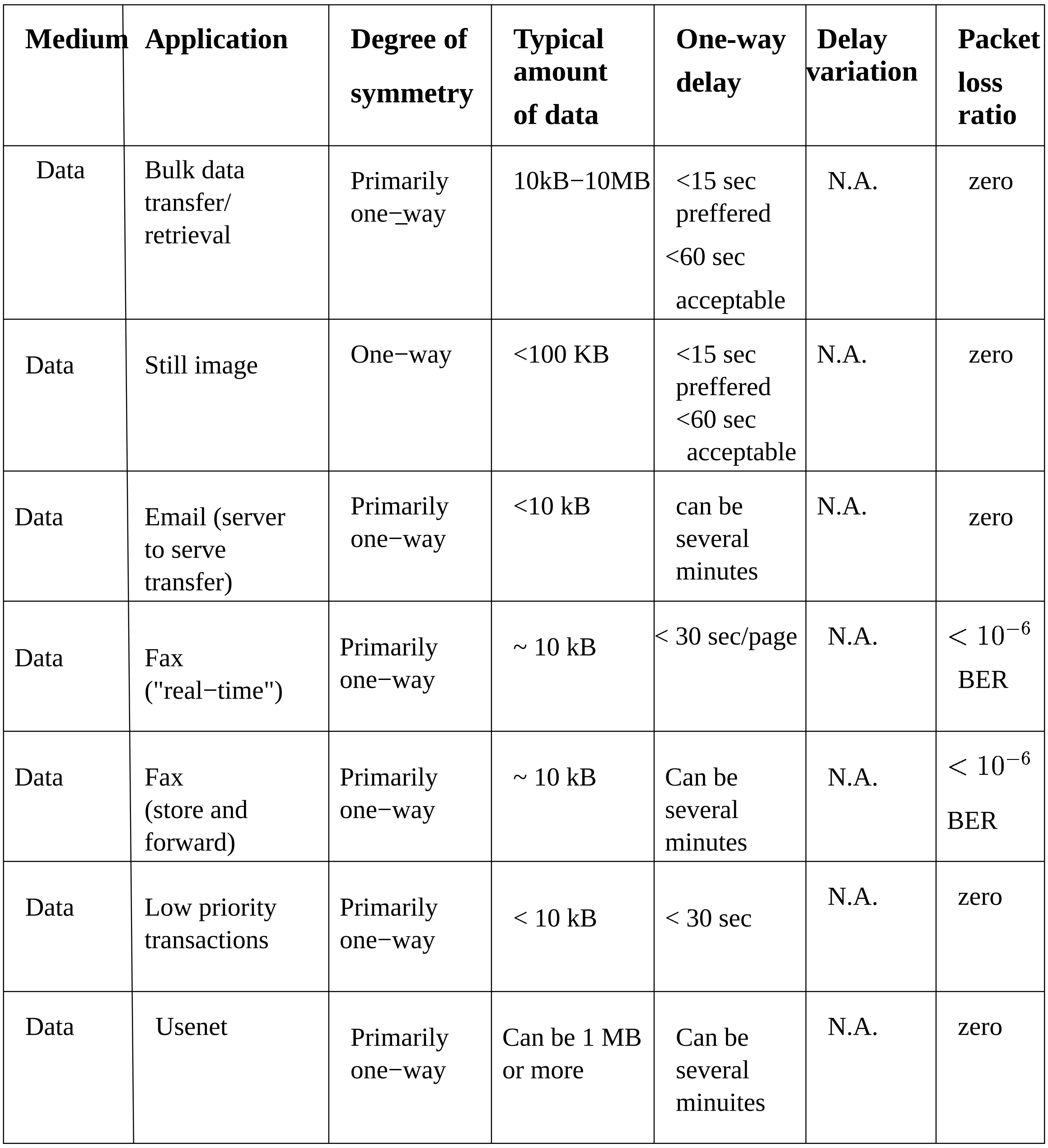
Medium	**Application**	**Degree of symmetry**	**Typical data rates**	**One-way delay**	**Delay variation**	**Packet loss ratio**
Audio	Coversational voice	Two−way	4−64 kbps	<150 msec preffered <400 msec limit	<1msec	<3%
Audio	Voice messaging	Primarily one−way	4−32 kbps	<1 sec for playback <2 sec for record	<1msec	<3%
Audio	High quality streaming audio	Primarily one−way	16−128 kbps	<10sec	<1msec	<1%
Video	Videophone	Two−way	16−384 kbps	<150 msec preferred <400 msec limit		<1%
Video	Medium quality video	one−way	16−384 kbps	<10 sec		<1%
Video	High quality video	one−way	4−6 Mbps	<10sec		<1%

Figure 9.1: Performance targets for audio and video applications



Medium	Application	Degree of symmetry	Typical amount of data	One-way delay	Delay variation	Packet loss ratio
Data	Web−browsing −HTML	Primarily one−way	~10kB	<2 sec/page preffered <4 sec/page acceptable	N.A.	zero
Data	Transaction services − high priority	Two−way	<10 KB	<2 sec preffered <4 sec acceptable	N.A.	zero
Data	Command/ control	Two−way	~1kB	<250 msec	N.A.	zero
Data	Interactive games	Two−way	< 1kB	<200 msec	N.A.	zero
Data	Telnet	Two−way (asymmetric)	< 1 kB	<200 msec	N.A.	zero
Data	E−mail (server access)	Primarily one−way	< 10 kB	< 2 sec preferred < 4sec acceptable	N.A.	zero

Figure 9.2: Performance targets for data applications (part 1)



Medium	Application	Degree of symmetry	Typical amount of data	One-way delay	Delay variation	Packet loss ratio
Data	Bulk data transfer/ retrieval	Primarily one−way	10kB−10MB	<15 sec preffered <60 sec acceptable	N.A.	zero
Data	Still image	One−way	<100 KB	<15 sec preffered <60 sec acceptable	N.A.	zero
Data	Email (server to serve transfer)	Primarily one−way	<10 kB	can be several minutes	N.A.	zero
Data	Fax ("real−time")	Primarily one−way	~ 10 kB	< 30 sec/page	N.A.	$< 10^{-6}$ BER
Data	Fax (store and forward)	Primarily one−way	~ 10 kB	Can be several minutes	N.A.	$< 10^{-6}$ BER
Data	Low priority transactions	Primarily one−way	< 10 kB	< 30 sec	N.A.	zero
Data	Usenet	Primarily one−way	Can be 1 MB or more	Can be several minuites	N.A.	zero

Figure 9.3: Performance targets for data applications (part 2)

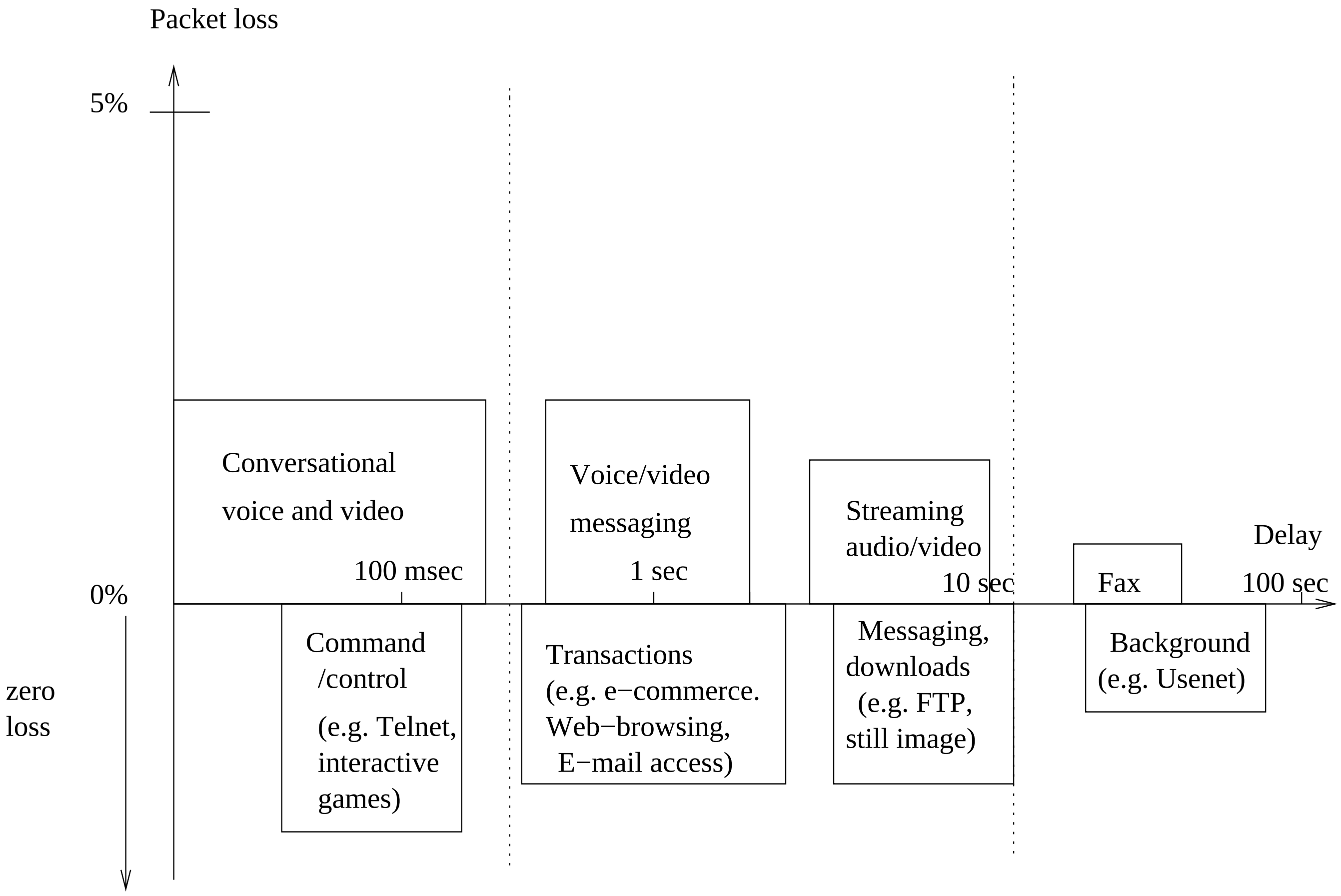


Figure 9.4: Summary of delay and loss requirements for audio, video and data applications.

Chapter 10

QoS architectures in Internet

This chapter describes the two QoS architectures for Internet defined by the IETF: Integrated Services (IntServ) and Differentiated Services (DiffServ). The service models defined in IntServ and DiffServ complement the best effort service model. Field experiments are currently under way for both IntServ and DiffServ. A combination of both architectures will most likely be deployed in the Internet within the next decade.

10.1 Integrated Services Architecture

IntServ is a per-flow based QoS framework with dynamic resource reservation [8]. Its fundamental philosophy is that routers need to reserve resources in order to provide quantifiable QoS for specific traffic flows. RSVP (Resource Reservation Protocol) serves as a signalling protocol for applications to reserve network resources. RSVP adopt a receiver-initiated reservation style which is designed for multicast environment and accommodates heterogeneous receiver service needs. RSVP works as follows. The flow source sends a PATH message to the intended flow receiver(s), specifying the characteristic of the traffic. As the PATH message propagates towards the receiver(s), each network router along the way records path characteristics such as available bandwidth. Upon receiving a PATH message, the receiver responds with a RESV message to request resources long the path recorded in the PATH message in reverse order from the sender to the receiver. Intermediate routers can accept or reject the request of the RESV message. If the request is accepted, link bandwidth and buffer space are allocated for the flow, and the flow-specific state information is installed in the routers.

Reservations can be shared along branches of the multicast delivery trees.

RSVP takes the *soft state* approach, which regards the flow-specific reservation state (given by the flow spec, see below) at routers as cached information that is installed temporarily and should be periodically refreshed by the end hosts. State that is not refreshed is removed after a timeout period. If the route changes, the refresh messages automatically install the necessary state along the new route. The soft state approach helps RSVP to minimize the complexity of connection setup and improves robustness, but can lead to increases flow setup times and message overhead.

The IntServ architecture adds two service models to the existing best-effort model, guaranteed service and controlled load service. Guaranteed service provides an upper bound on the end-to-end delay. Moreover, it guarantees zero loss in buffers, but keep in mind that packet loss can still occur due to random bit errors. No average delay or jitter guarantees are given. This guaranteed service model is aimed to support applications with hard real-time requirements. Controlled-load service provides a quality of service similar to best-effort service in an underutilized, non-congested network, with almost no loss and queuing delay. It is aimed to share aggregate bandwidth among multiple traffic streams in a controlled way under overload conditions. The guaranteed service model may be mapped to class 1 in ITU-T recommendation Y.1541, and controlled load service may be mapped to class 2.

The RSVP *flow descriptor* is carried in the RSVP messages. The flow descriptor consists of a *filterspec* and a *flowspec*. The filterspec is used by the routers to select which packets to give special QoS service and which to give best-effort service. The flowspec contains the QoS class, traffic parameters (TSpec), and requested resources (RSpec). The TSpec contains five parameters: the peak rate p, maximum burst size b, mean rate r, minimum policed unit m and maximum policed unit M. Both the guaranteed service flows and the controlled-load service flows will have a TSpec. The RSpec contains the requested bandwidth R and a slack term S. The slack term represents the amount of which the end-to-end delay bound will be below the end-to-end delay requested by the application, assuming each router along the path reserves R bandwidth for guaranteed service flows according to the Weighted Fair Queuing (WFQ) discipline (see the chapter of Performance evaluation). Only the guaranteed service flow will specify a RSpec.

Lets consider resource reservation for a multicast situation where there may be multiple senders to a group and multiple receivers. First, let's first deal with multiple receivers for a single sender. As a RESV message travels up the multicast tree, it is likely to hit a piece of the tree where some other receiver's reservation has already been established. It may the case that the resources reserved upstream of this point are adequate to serve both receivers. For example, if receiver A has already made a reservation that provides for a guaranteed delay of less than 100 ms, and the new request from receiver B is for a delay less than 200 ms, then no new reservation is required. On the other hand, if the new request were for a delay of less than 50 ms, the router would first see if it could accept the request, and if so, it would send the request upstream. The next time receiver ask for a minimum of 100 ms delay, the router would not need to pass this request on. In general, reservation can be merged in this way to meet the needs of all receivers downstream of the merge point.

If there are also multiple senders in the three, receivers need to collect the TSpecs from all senders and make the reservation that is large enough to accomodate the traffic for all senders. However, this may not mean that the TSpecs need to be added up. For example, in an audio conference with 10 speakers, there is not much point in allocating resources to carry 10 audio streams, since the result of 10 people speaking at once would be incomprehensible. Thus, we could imagine a reservation that is large enough to accomondate two speakers and no more. Calculating the correct overall TSpec from all the sender TSpecs is clearly application specific.

By using per-flow resource reservation, IntServ can deliver fine-grained QoS guarantees. However, introducing flow-specific state in the routers represents a fundamental challenge to the current Internet architecture. Particularly in the Internet backbone, where hundred thousand flows may be present, this may be difficult to manage, as router may need to maintain a separate queue for each flow.

Although RSVP can be extended to reserve resources for aggregation of flows, many people in the Internet community believe that the IntServ framework is more suitable for intra-domain QoS or for specialized applications such as high-bandwidth flows. IntServ also faces the problem that incremental deployment is only possible for controlled-load service, while ubiquitous deployment is required for guaranteed service, making it difficult to be realized across the network.

10.2 Differentiated Services Architecture

10.2.1 Per-Hop-Behavior classes

To address some of the problems associated with IntServ, Differentiated Services (DiffServ) has been proposed by the IETF with scalability as the main goal [6]. DiffServ is a per-aggregate-class based service discrimination framework using packet tagging. Packet tagging uses bits in the packet header to mark a packet for preferential treatment. In IPv4, the type-of-service (TOS) byte is used to mark packets. The TOS byte consists of a 3-bit precedence field, a 4-bit field indicating requests for minimum delay, maximum throughput, maximum reliability and minimum cost, and one unused bit. However, these bits were never widely used. DiffServ redefined this byte as the DS field, of which six bits make up the DSCP (Differentiated Service CodePoint) field, and the remaining two bits are unused. In IPv6 the Traffic Class byte will convey the DSCP label. The interpretation of the DSCP field is currently being standardized by the IETF.

DiffServ uses DSCP to select the per-hop-behavior (PHB) a packet experiences at each node. A PHB is an externally observable packet forwarding treatment which is usually specified in a relative format compared to other PHBs, such as relative weight for sharing bandwidth or relative priority for dropping. The mapping of DSCPs to PHBs at each node is not fixed. Before a packet enters a DiffServ domain, its DSCP field is marked by the end-host or the first-hop router according to the service quality the packet required and entitled to receive. Within the DiffServ domain, each router needs to look at the DSCP to decide the proper treatment for the packet. No complex classification of per-flow state is needed.

DiffServ has two important design principles, namely pushing the complexity to the network boundary and separation of policy and supporting meachainisms. The network boundary refers to application hosts, leaf (of first-hop) routers, and edge routers. Since a network boundary has relative small number of flows, it can perform operations at fine granularity, such as complex packet classification and traffic conditioning. In contrast, a network core router may have a large number of flows, and should perform fast and simple operations. The differentiation of network boundary and core routers is vital for the scalability of DiffServ.

The separation of control policy and supporting meachanisms allows these to evolve independently. DiffServ only defines several PHBs as the basis building block for QoS provisioning, and leaves the control policy as an issue for further study. The control policy can be changes as needed, but the supporting PHBs should be kept relatively stable. The separation of these two components is key for flexibility of DiffServ. A similar example is Internet routing. It has very simple and stable forwarding operations, while the construction of routing table is complex and may be performed by a variety of different protocols.

Currently, DiffServ provides two service models besides best effort. Premium service is a guaranteed peak rate service, which is optimized for very regular traffic patterns and offers small or no queuing delay. One example of using it is to create "virtual leased lines", with the purpose of saving the cost of building and maintaining a separate network. Assured service is based on statistical provisioning. It tags packets are *In* or *Out* according to their service profiles. *In* packets are unlikely to be dropped, while *Out* packets are dropped first if needed. This service relies on relative QoS guarantees.

Up to now IETF has standardized two PHB classes: Expedited Forwarding (EF) and Assured Forwarding (AF). The EF class provides quantitative (absolute) QoS guarantees. The AF class provides qualitative (relative) QoS guarantees. The EF class is defined for the premium service model, while the AF class is defined for the assured service model. The EF class has statistical loss, delay and jitter guarantees. The AF class only specifies only dropping priorities, no relative delay or jitter guarantees are given. Four AF sub classes each with three drop priority levels are standardized. The EF class may be mapped to class 1 in ITU-T recommendation Y.1541, and AF class may be mapped to class 2.

EF and AF can be realized with Priority Queuing (PQ) or WFQ. In PQ or WFQ, the EF class, the four AF sub classes and the best effort class should get their own queue. Within each AF sub class, two buffer thresholds can be used to implement the three dropping priorities. In PQ, the queues are served with different priorities, with the EF class having the highest priority, and followed by the four AF sub classes and the best effort class in decreasing priority order. The weights in WFQ should be set according to the expected traffic and the priority order among the classes. The EF and AF flows are described by the same five traffic parameters used by the Guaranteed and Controlled load service models in IntServ.

10.2.2 DiffServ domains and DiffServ regions

A DiffServ (DS) domain consists of DS boundary nodes and DS interior nodes. DS boundary nodes interconnect the DS domain to other DS or non-DS capable domains, while the DS interior nodes only connect other DS interior or boundary nodes within the same DS domain.

Both DS boundary nodes and interior nodes must be able to apply the appropriate PHB to packets based on the DSCP; otherwise unpredictable behavior may result. In addition, DS boundary nodes may be required to perform traffic conditioning functions as specified by the traffic conditioning agreement (TCA) between their DS domain and the peering domain which they connect to.

Interior nodes may be able to perform limited traffic conditioning functions such as DSCP re-marking. Interior nodes which implement more complex classification and traffic conditioning are analogous to DS boundary nodes.

A DS region is a set of one or more DS domains. DS regions are capable of supporting differentiated services along paths which span domains within the same region.

Differentiated services are extended across a DS domain boundary by establishing a service level agreement (SLA) between an upstream network and a downstream DS domain. The SLA may specify packet classification amd re-marking rules and may also specify traffic profiles and actions to traffic streams which are in- or out-of-profile. The TCA between the domains are derived (explicitly or implicitly) from this SLA.

10.2.3 Traffic conditioning

Traffic conditioning performs metering, shaping, policing and/or re-marking to ensure that the traffic entering the DS domain conforms to the rules specified by the TCA, in accordance with the domain's service provisioning policy.

Packet classifiers selects packets in a traffic stream based on the content of some portion of the packet header, typically the DSCP value.

A traffic profile specifies the temporal properties of the traffic stream selected by a classifier. It provides rules for determining whether a particular packet is in-profile or out-of-profile. Different conditioning actions may be applied to the in-profile and out-of-profile packets, or different accounting actions may be triggered. In-profile packets may be allowed to enter the DS domain without further conditioning; or, alternatively their DSCP may be changed. Out-of-profile packets may be queued until they are in-profile (shaped), discarded (policed), marked with a new DSCP (re-marked), or forwarded unchanged while triggering some accounting procedure.

Traffic meters measure the temporal properties of the stream of packets selected by the classifies against a traffic profile specified in a TCA. A meter passes state information to other conditioning functions to trigger a particular action for each packet which is either in- or out-of-profile. A traffic meter may be implemented by two token buckets.

Packet markers set the DSCP field of a packet to a particular codepoint, adding the marked packet to a particular DS behavior aggregate.

Shapers delay some or all of the packets in a traffic steam in order to bring the stream into compliance with a traffic profile. A shaper usually has a finite-size buffer, and packets may be discarded if there is not sufficient buffer space to hold the delayed packets.

Droppers discard some or all of the packets in a traffic stream in order to bring the stream in compliance with a traffic profile. This process is known as "policing" the stream. Note that a dropper can be implemented as special case of a shaper by setting the shaper buffer size to zero (or a few packets).

10.2.4 IntServ over DiffServ

One possible evolution scenario is that DiffServ will be employed in the Internet backbone (core), while IntServ will be used in access domain of the users. Thus, Internet will consist of IntServ regions connected to DiffServ regions. The micro-flows of the users will be policed at the end-host, at the IntServ edge router or at the DiffServ border router. The DiffServ domains in the DiffServ region will police the aggregate flows at their border routers.

Requests for IntServ services must be mapped onto the underlying capabilities of the DS region. Aspects of the mapping include:

- selecting an appropriate PHB, or set of PHBs, for the requested service;
- performing appropriate traffic conditioning at the edges of the DS region;
- exporting IntServ parameters from the DS region;

- performing admission control on the IntServ requests that takes into account the resource availability in the DS region.

The guaranteed service class in IntServ may be mapped to the EF PHB class, while the controlled load service class may be mapped to the AF PHB class.

A variety of options exist for management of resources (bandwidth, buffers) in the DS region to meet the needs of end-to-end IntServ flows. These options include:

- statically provisioned resources;
- resources dynamically provisioned by RSVP;
- resources dynamically provisioned by a bandwidth broker.

RSVP-aware routers in the DS region have a RSVP control plane but a DiffServ data plane. When the DS region is RSVP aware, the admission control agent is part of DiffServ network. Admission control can be linked to the availability of resources along a specific path that would be impacted.

Border routers might not use any form of RSVP signalling within the DS region but might instead use custom protocols to interact with a bandwidth broker. The bandwidth broker is an centralized agent that has sufficient knowledge of resource availability and network topology to make admission control decisions. The bandwidth broker allocates intra-domain resources and arranges inter-domain agreements. In its inter-domain role, a bandwidth broker negotiates with its neighbor domains, sets up a bilateral agreement with each of them, and sends the appropriate configuration parameters to the domains' edge routers. Bilateral agreements means that the bandwidth broker only needs to coordinate with its adjacent domains. End-to-end QoS is provided by the concatenation of these bilateral agreements across domains, together with adequate intra-domain resource allocation.

Part V

Multimedia traffic control

Chapter 11

Traffic control overview

In this chapter we present an overview of traffic control in multi-service networks. The starting point is a network consisting of a set of nodes, interconnected by links according to some network topology. In case of ATM or IP networks the network topology is often partly meshed. We concentrate on traffic control on the OSI network and transport layers. Such a layer is either connection-oriented or connectionless. In the former case, the communication involves three phases: connection set-up, data transfer, and connection release. In the latter communication involves just the data transfer phase. We use the term *virtual circuit* (VC) to denote a network layer connection in an ATM network, and the term *flow* to denote the connectionless equivalent to the VC in an IP network. The ATM VC is identified by the VPI/VCI value in the ATM cell header. The IP flow is identified by the source and destination IP address, source and destination port number and the transport protocol (TCP/UDP). Alternatively, the flow identifier in the IPv6 packet header can be used. Often, the presentation refers to both ATM VCs and IP flows. In this case we use the term *call* to refer to both ATM VCs and IP flows.

We use an hierarchical model to describe traffic in the network [30]. The model consists of three layers operating on different time scales, see Figure 11.1. The upper layer is the *call layer* which operates in a time scale of seconds. The traffic on this layer is characterized by the call arrival process and the call holding time distribution. Calls generate packets, either at a constant rate or at variable rate. In the later case, the *burst layer* is used to model the burst arrival process. The burst later is an intermediate traffic layer which operates in the time scale of milli seconds. The lowest layer is the packet layer which operates in the time scale

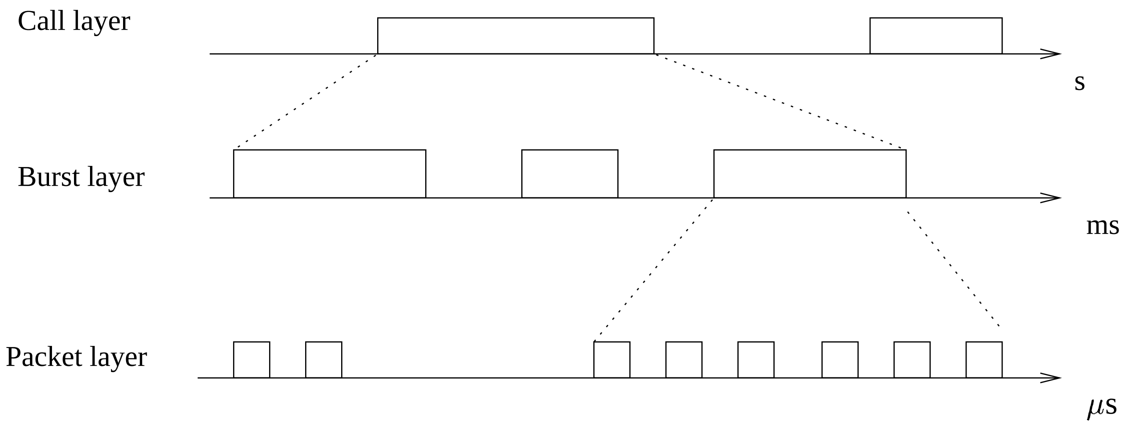


Figure 11.1: Hierarchical traffic model

of micro seconds.

The network performance on the call layer is called Grade of Service (GoS) and consists of the call blocking probability and the call set up delay. The call blocking probability measures the network availability. Telephone networks are usually dimensioned to block (reject) less than 1 % of the call requests. The call set up delay in ATM networks should be in the order of tens of milliseconds.

Network performance on the burst and packet layer is called Quality of Service (QoS) and consists of end-to-end packet loss probability, delay, and delay variability (jitter). The delay can be described by the mean delay or the $(1 - \alpha)$ quantile of the delay distribution. The jitter is defined as the difference between the $(1 - \alpha)$ quantile of the end-to-end delay and the minimum end-to-end delay.

Network services are divided into guaranteed services and elastic services. Guaranteed services offers deterministic or statistical QoS guarantees. A deterministic guarantee could be zero packet loss and guaranteed packet delivery within 100 ms. Statistical guarantees could be packet loss probability of 10^{-6}, delivery of 98 % of the packets within 150 ms, and jitter less than 50 ms. Elastic services can tolerate relatively large variations in throughput and delay. However, they typically require low packet loss.

Traffic control is of utmost importance in multi-service communication networks. The goal of traffic control is to maintain the QoS and network availability, while using the network resources as efficiently as possible. Then latter behavior is crucial for network operators who want to maximize their revenue in a competitive environment.

Traffic control is concerned with the three-way relationship between traffic volume, network resources and realized QoS and GoS. Fixing two of them, the third quantity can be determined. The traffic control algorithms for realizing traffic control in B-ISDN and future Internet will not be standardized.

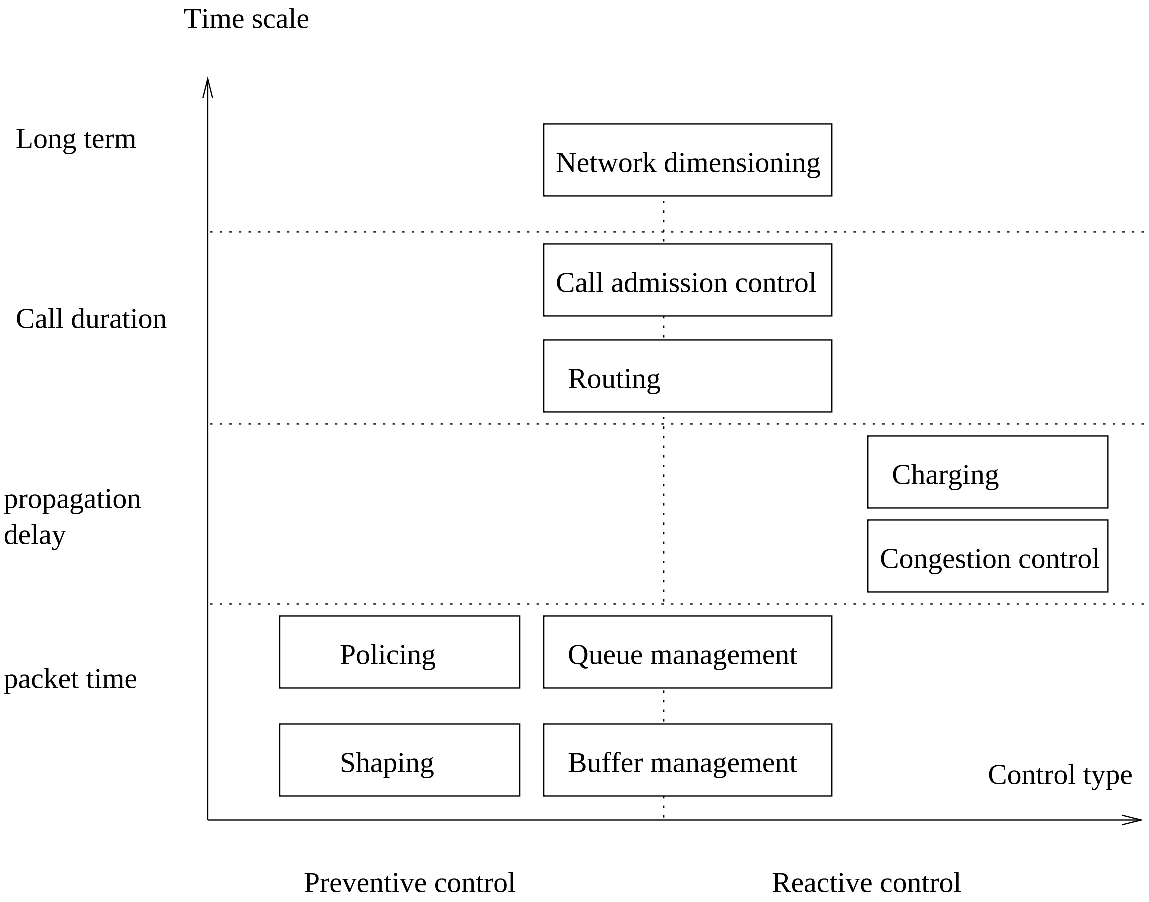


Figure 11.2: Classification of traffic control functions

Traffic control functions are classified into three main categories: preventive traffic control, reactive traffic control, and combined preventive/reactive traffic control functions. Preventive traffic control aims at avoiding congestion from occurring too often. Reactive traffic control resolves congestion situations occurring during the information transfer phase.

Traffic control functions are also classified according to the time scale they operate on. We can distinguish the packet time scale, the propagation delay time scale, the call duration time scale, and the long term time scale.

Figure 11.2 shows a classification of the traffic control functions studied in this report.

Call admission control (CAC) and traffic policing are the foundation of preventive traffic control for guaranteed services. CAC decides whether a new call request can be accepted or if it must be rejected. The decision is based on QoS, GoS and revenue considerations. The policing function enforces the traffic parameters agreed upon in the traffic contract negotiated by the user and the network at call set up. Traffic shaping can be used by the end terminals to produce a packet stream which conforms to a given traffic profile. The traffic shaper can be implemented by a packet buffer togheter with a traffic policing unit. The packets are delayed in the buffer until they conform to the declared traffic profile according to the policing unit. The queue and buffer management functions are used in the switches/routers to schedule the use of buffer and link capacity. Congestion control is based on adapting the rate of traffic sources to the available capacity in the network. Congestion control is aimed at elastic services which can adapt to relatively large variations in throughput and delay. Charging based on usage of resources is most promising in multi-service networks. The charging and rate allocation problem can be solved together. Routing is performed in close operation with the CAC function. Network dimensioning determines network topology and capacities of physical and virtual networks, given the call traffic demand and GoS requirements.

The teletraffic engineer designing the various traffic control functions faces an accuracy-simplicity dilemma. Accurate QoS and GoS performance models are desired to enable efficient resource allocation decisions, which are neither too pessimistic nor too optimistic. Simple QoS and GoS performance models are necessary to allow practical implementations which meet the timing constraints of the decision making.

Chapter 12

Traffic models

12.1 Introduction to traffic modelling

12.1.1 Definition of traffic processes

Simple traffic consist of single arrivals of discrete entities (calls, packets etc.) [23]. It can be mathematically described a point process, consisting of a sequence of arrival instants $T_1, T_2, ..., T_n, ...$ measured from the origin 0; by convention, $T_0 = 0$.

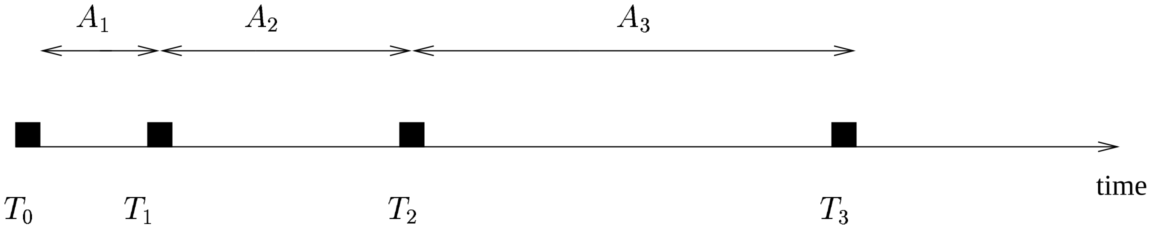


Figure 12.1: Example of traffic arrivals.

There are two additional equivalent descriptions of point processes: counting processes and inter-arrival time processes. A counting process $\{N(t)\}_{t=0}^{\infty}$ is a continuous-time, non-negative integer-valued stochastic process, where $N(t) = \text{Max}\{n : T_n \leq t\}$ is the number of (traffic) arrivals in the interval $(0, t]$. An inter-arrival time process is a non-negative random sequence $\{A_n\}_{n=1}^{\infty}$, where $A_n = T_n - T_{n-1}$ is the length of the time interval separating the n-th arrival from the previous one, see Figure 12.1. The equivalence of these descriptions follows from the equality of events:

$$\{N(t) = n\} = \{T_n \le t \le T_{n+1}\} = \{\sum_{k=1}^{n} A_k \le t < \sum_{k=1}^{n+1} A_k\} \qquad (12.1)$$

since $T_n = \sum_{k=1}^{n} A_k$ Unless otherwise stated, we assume throughout the chapter that $\{A_n\}$ is a stationary sequence and that the common variance of A_n is finite.

Compound traffic consists of batch arrivals; that is, arrivals may consist of more than one unit at arrival instant T_n. In order to fully describe compound traffic, one also needs to specify a real-valued random sequence $\{B_n\}_{n=1}^{\infty}$, where B_n is the (random) number of units in the batch. At a higher level of abstraction, B_n may represent some general attributes of the n-th arrival, e.g. the amount of "work" associated with the n-th arrival.

In addition to arrival times and batch sizes, it is often useful (and sometimes essential) to incorporate the notion of *workload* into the traffic description. The workload is a general concept describing the amount of work W_n brought to the system by the n-th arriving unit; it is usually assumed independent of interarrival times and batch sizes. A typical example is the sequence of service time requirements of arrivals at a queuing system, though in queuing, one usually refers to the arrival process alone as traffic. On the other hand, traffic reduces to workload description, when interarrival times are deterministic. A case in point is compressed video, also known as coded video. The compressed frames have random sizes (bit rates) which are then transported over the network and decoded at their destination. Coded video frames (arrivals) must be delivered deterministically every 1/30 of a second or so, for high-quality video. The workload consists of coded frame sizes (say, in bits), since a frame size is roughly proportional to its transmission time (service requirement).

12.1.2 Second-order statistics

Second-order statistics express relations between traffic occurencies at two time instants. The following are second order properties of a stochastic process $\{X_n\}$ in discrete time [47]. For example, $\{X_n\}$ can be the workload process associated with a sequence of compressed video frames, or the number of arrivals in time slot n (batch size) at a certain buffer.

- *Autocovariance C(k)*:

$$C(k) = E[(X_n - \overline{X})(X_{n+k} - \overline{X})] = E[X_n X_{n+k}] - \overline{X}^2 \tag{12.2}$$

- *Autocovariance for zero mean process R(k)*:

$$C(k) = E[X_n X_{n+k}] \tag{12.3}$$

- *Autocorrelation* $r(k)$:

$$r(k) = C(k)/\mathit{Var}[X_k] \tag{12.4}$$

- *Power spectrum* $\Phi(\omega)$:

$$\Phi(\omega) = \mathcal{F}\{R(k)\} = \sum_{k=-\infty}^{\infty} R(k) e^{-i\omega k} \tag{12.5}$$

- *Autocovariance spectrum* $S(\omega)$:

$$S(\omega) = \mathcal{F}\{C(k)\} = \sum_{k=-\infty}^{\infty} C(k) e^{-i\omega k} \tag{12.6}$$

The autocovariance spectrum $S(\omega)$ can be expressed as

$$S(\omega) = lim_{M\to\infty} E\left\{ \frac{1}{2M+1} \left| \sum_{n=-M}^{M} x(n) e^{-i\omega n} \right|^2 \right\} \tag{12.7}$$

The squared $\mathcal{F}$-transform of $x(n)$ (divided by the record length) may be used as an estimator of the autocovariance spectrum. As such, it is known as the *periodogram* $\hat{S}_{PER}(\omega)$:

$$\hat{S}_{PER}(\omega) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-i\omega n} \right|^2 \tag{12.8}$$

12.1.3 Traffic burstiness

A recurrent theme relating to traffic in multi-service networks is the traffic "burstiness" exhibited by key services such as compressed video, file transfer etc. Burtiness is present in a traffic process if the arrival points $\{T_n\}$ appear to form visual clusters; that is, $\{A_n\}$ tends to give rise to runs of several relatively short inter-arrival times followed by a relatively long one. The mathematical underpinning of burstiness is more complex. Two main sources of burstiness are due to the shapes of the marginal distribution and the autocorrelation function of $\{A_n\}$. For example, burstiness would be facilitated by a bimodal marginal distribution of $\{A_n\}$, or by short-term autocorrelation in $\{A_n\}$. Strong positive autocorrelation are a particularly major cause of burstiness. Since there seems to be no single widely-accepted notion of burstiness, we call briefly describe some of the commonly-used mathematical measures that attempt to capture it [23].

The two simples measures of burstiness take account only of first-order properties of traffic (they are each a function of the marginal distribution only of inter-arrival times). The first one is the ratio of the peak rate to the mean rate – a very crude measure, which also has the shortcoming of dependence on the interarrival length utilized for rate measurement. A more elaborate measure of burstiness is the coefficient of variation, defined as the ratio of the standard deviation to mean $c_A = \sigma[A_n]/E[A_n]$ of inter-arrival times.

In contrast, the peakedness measure and the index-of-dispersion measures do take account of temporal dependence in traffic (second-order properties). For a given time interval of length τ, the index of dispersion for counts (IDC) is the function $I_c(\tau) = \mathit{Var}[N(\tau)]/E[N(\tau)]$; i.e., the variance-to-mean ratio of the number of arrivals in the interval $[0, \tau]$. Since the number of arrivals is related to the sum of inter-arrival intervals via Equation (12.1), the numerator of the IDC includes the autocorrelation of $\{A_n\}$. The index of dispersion for intervals (IDI) is defined as

$$J_k = \frac{\mathit{Var}\left[\sum_{n=1}^{k} A_{i+n}\right]}{kE^2[A]} \tag{12.9}$$

where $\{A_i\}$ is a stationary sequence of inter-arrival times.

The peakedness concept is related, but more involved. Assume that the traffic stream $\{A_n\}$ is offered to an infinite server group consisting of independent servers with common service times distribution F. Let S be the equilibrium number of busy servers. The peakedness is the functional $z_A[F] = \mathit{Var}[S]/E[S]$, which maps a service time distribution to a real number. A commonly used peakedness is $z_{exp}(0)$, obtained as a limiting case for an exponential service distribution with service rate approaching 0.

Finally, the Hurst parameter H can be used as measure of burstiness via the concept of self-similarity. The Hurst parameter is defined below.

12.1.4 Characteristics of self-similar traffic

Recent studies of high-quality, high-resolution traffic measurements have revealed a new phenomenon with potentially important ramifications to the modeling, design, and control of multi-service networks. These include an analysis of hundreds of millions observed packets over an Ethernet LAN in a R & D environment at Bellcore [43], an analysis of few millions of observed frame data generated by VBR video services [25], and analysis of tens of thousands of TCP connection arrivals on the Internet [22].

In these studies, packet and connection traffic appears to be statistically self-similar. A self-similar (or fractal) phenomenon exhibits structural similarities across a wide range of time scales. In the case of packet traffic, self-similarity is manifested in the absence of natural length of a burst: at every time scale ranging from a few milliseconds to minutes to hours, similar-looking traffic bursts are evident [47].

The following definition of self-similarity for stochastic processes is widely adopted. Assume X_k to be a wide-sense stationary process with mean $E[X_k] = \overline{X}$ and autocorrelation function $r(k)$.

Consider next the process $X_k^{(m)}(m = 1, 2, ...)$ that are constructed out of X_k as $X_k^{(m)} = \sum_{n=0}^{m-1} X_{km+n}/m$, i.e. by averaging over non overlapping blocks of size m. The process $X_k^{(m)}$ are also wide-sense stationary, with mean $\overline{X}$ and autocorrelation function $r^{(m)}(k)$. The process X_k is called *exactly self-similar* if it satisfies

$$X_k \overset{d}{=} m^{1-H} X_k^{(m)} \tag{12.10}$$

That is, the processes X_k and $m^{1-H}X_k^{(m)}$ should have the same finite-dimensional distributions for all aggregation levels m.

The process X_k is called *asymptotically second-order self-similar* if it satisfies

$$r^{(m)}(k) = r(k) \text{ for } m, k \to \infty. \tag{12.11}$$

Self-similarity manifests itself in a variety of ways:

- slowly (hyperbolically)decaying variances: *Var*$[X_k^{(m)}] \sim m^{-\beta}$ if $m \to \infty$, whereby $0 < \beta < 1$;
- a slowly decaying autocorrelation function: $r(k) \sim k^{-\beta}$ if $k \to \infty$;
- a power spectral density $S(f)$ that behaves like that of $1/f$ noise around the origin: $S(f) \sim f^{-(1-\beta)}$ if $f \to 0$

Self-similarity also implies *long-range dependence* (LRD), i.e., $\sum_{k=-\infty}^{+\infty} r(k) = \infty$. A process for which $\sum_{k=-\infty}^{+\infty} r(k) < \infty$ is said to be *short-range dependent* (SRD). Such a process differs from a LRD process in the sense that

- the variances *Var*$[X_k^{(m)}]$ decay as m^{-1};
- $r(k)$ decays exponentially fast: $r(k) \sim \sum_{n=1}^{p} c_n \rho_n^k$ for large k;
- the power spectral density remains finite (and approximately constant) around the origin;
- the process behaves like (second-order) pure noise for large m.

An important parameter of a LRD process is the so-called self-similarity of Hurst parameter $H = 1 - \beta/2$. It is named after H. Hurst, who observed the following fact. Given a set of experimental data a_k ($k = 1, ..., n$), with sample mean $\overline{a}(n) = \sum_{k=1}^{n} a_k/n$ and sample variance $S^2 = (\sum_{k=1}^{n}[a_k - \overline{a}(n)])/(n-1)$, define the rescaled adjusted range (R/S) statistic as

$$\frac{R(n)}{S(n)} = \frac{1}{S(n)}[\text{Max}(0, W_1, W_2, ..., W_n) - \text{Min}(0, W_1, W_2, ..., W_n)] \tag{12.12}$$

whereby $W_k = (a_1 + a_2 + ... + a_k) - k\overline{a}(n)$. The quantities W_k measure the deviation of the process a_k from its “expected value”. $R(n)$ then measures the “record” values of this deviation. For many “naturally” occurring processes, one has $E[R(n)/S(n)] \sim n^H$ when

$n \to \infty$ with H “typically” around 0.7. If a_k is short-range- dependent process, i.e. with a correlation structure over only small time scales, one would have $H = 0.5$. The larger experimental values can be explained by assuming that a_k is self-similar.

Some “tools” to asses the self-similar nature of a given trace are:

- visual inspection – plots of samples of the process $X_k^{(m)}$ for a wide range of values of m all look similar for fractal traffic, while those for short-range-dependent traffic “flatten out” when m gets large;
- the R/S plot – plotting log$[R(n)/S(n)]$ versus log(n) for various subsets of the available data allows one to determine, by linear regression, the Hurst parameter H;
- the variance-time plot – plotting log *Var*$[X_k^{(m)}]$ versus log(m) allows one to estimate $\beta = 2(1 - H)$;
- the spectral density, which can be estimated by means of the periodogram explained previously – plotting log$[S(f)]$ versus log(f) allows an estimate for $1 - \beta$.

12.2 Call traffic models

12.2.1 Call arrival process models

Call arrivals are usually modeled by renewal processes [59]. In a renewal traffic process, the A_n are independent, identically distributed (IID), but their distribution is allowed to be general. Unfortunately, with a few exceptions, the superposition of independent renewal processes does not yield a renewal process. The ones that do however, occupy a special position in traffic theory and practice. Queuing models historically have continually assumed renewal-offered traffic.

Renewal processes, while simple analytically, have a severe modeling drawback – the autocorrelation function of $\{A_n\}$ vanishes for all nonzero lags. The importance of capturing autocorrelations stems from the role of the autocorrelation function as a statistical proxy for temporal dependence in time series. Since burstiness can be explained to a large extent by positive autocorrelation in $\{A_n\}$, renewal processes are not used to model packet arrival processes. However, for call arrival processes, the renewal model is believed to be adequate.

The *Poisson model* is the oldest traffic model, dating back to the advent of telephony and the renowned pioneering telephone engineer A.K. Erlang. A Poisson process can be characterized as a renewal process whose inter-arrival times $\{A_n\}$ are exponentially distributed with rate parameter λ: $\mathrm{P}(A_n \leq t) = 1 - \exp(-\lambda t)$. Equivalently, it is a counting process, satisfying $\mathrm{P}(N(t) = n) = \exp(-\lambda t)(\lambda t)^n/n!$, and the number of arrivals in disjoint intervals is statistically independent (a property known as independent increments).

Poisson processes enjoy some elegant analytical properties. First, the superposition of independent Poisson processes results in new Poisson process whose rate is the sum of the component rates. Second, the independent increment property renders Poisson a memoryless property. This, in turn, greatly simplifies queuing problems involving Poisson arrivals. Third, Poisson process are fairly common in traffic applications that physically comprise a large number of independent traffic streams, each of which can be quite general. The theoretical basis for this phenomenon is known as Palm's Theorem. It roughly states that under suitable but mild regularity conditions, such multiplexed streams approach a Poisson process as the number of streams grows, but the individual rates decrease so as to keep the aggregate rate constant.

12.2.2 Call holding time models

The traditional model of call holding times B_n is the (negative) exponential distribution with rate parameter μ: $\mathrm{P}(B_n \leq t) = 1 - \exp(-\mu t)$ [59]. A more recent model is the the Pareto distribution which is a heavy-tailed distribution of the form $\mathrm{P}(B_n \leq t) = 1 - \left(\frac{k}{t}\right)^{\alpha}, k > 0$. A distribution is said to be heavy tailed if Prob$[X > x] \sim cx^{\alpha}$ with $\alpha > 0$. Intuitively, a heavy-tailed holding time distribution means that if the call has not been completed for some time it becomes more and more unlikely that it will be completed soon.

12.3 Applications of call traffic models

Many services in a multi-service networks, such as voice, video-telephony, file transfer, remote login are believed to yield Poisson call arrival processes. However, an important exception from this rule is Web traffic, which comprises more than 25 % of Internet traffic. Recent results have shown the the inter-arrival process of Web connections is self-similar [22]. This can be motivated by comparing the behavior of Web user and for example a file transfer user. The Web user who starts browsing is much more likely to download another set of Web pages than to stop after just one page. In contrast, a file transfer user is not more likely to initiate new session after the first one.

Anja Feldmann has observed that the self-similar arrivals of Web connections can be accurately modeled by a renewal process with inter-arrival times following a Weibull distribution [22]. This observation is supported by the observation that the silence time of Web users are found to be heavy tailed [14].

The Weibull distribution is a generalization of the exponential distribution in the sense that a variable x has a Weilbull distribtion if $y = \left(\frac{x}{a}\right)^c$ has an exponential distribution with probability density function $p(y) = e^y$. That is, the Weibull distribution has the form $\mathrm{P}(A_n \leq t) = 1 - \exp\left[-\left(\frac{t}{a}\right)^c\right]$. As the value c decreases the probability of longer as well as shorter values increases, and the burstiness of the traffic increases. Anja Feldmann obtained a self-similar or Hurst parameter value of 0.7 in her measurement analysis of Web connections on the Internet.

Holding times for many interactive services such as telephony are accurately described by the exponential distribution. However, a notable exception are Internet sessions established at an ISP which have holding times more accurately described by a Weibull distribution or a Pareto distribtion. Another example is the holding time of Web connections which are believed to be heavy tailed [14]. The reason for the heavy tailed holding times of Web connections are due to the fact that the distribution of file sizes on the Internet appear to be heavy tailed.

12.4 Packet traffic models

12.4.1 Markov-based traffic models

Unlike renewal traffic models, Markov and Markov-renewal traffic models introduce dependence into the random sequence $\{A_n\}$ [23]. Consequently, they can potentially capture traffic burstiness, due to non-zero autocorrelations in $\{A_n\}$.

Consider a Markov process $M = \{M(t)\}_{t=0}^{\infty}$ with a discrete state space. In this case, M behaves as follows: it says in a state i for an exponentially distributed holding time with parameter λ_i which depends on i alone; it then jumps to state j with probability p_{ij}, such that the matrix $P = [p_{ij}]$ is a probability matrix. In a simple Markov traffic model, each jump of the Markov process is interpreted as a signalling an arrival, so inter-arrival times are exponential, their rate parameter depending on the state from which the jump occurred.

Markov models in slotted time can be defined for the process $\{A_n\}$ in terms of a Markov transition matrix $P = [p_{ij}]$. Here, state i corresponds to i idle slots separating successive arrivals, and p_{ij} is the probability of a j-slot separation, given that the previous one was an i-slot separation. Arrivals may be single, a batch of units or a continuous quantity. Batches may themselves be described by a Markov chain, whereas continuous-state, discrete-time Markov processes can model the (random) workload arriving synchronously at the system. In all cases, the Markovian structure introduces dependence into inter-arrival separation, batch sizes and successive workloads, respectively.

Markov-renewal models are more general than discrete-state Markov processes, yet retain a measure of simplicity and analytical tractability. A Markov renewal process $R = \{(M_n, \tau_n)\}_{n=0}^{\infty}$ is defined by a Markov chain $\{M_n\}$ and its jump times $\{\tau_n\}$, subject to the following constraint: The distribution of the pair (M_{n+1}, τ_{n+1}), of next state and inter-jump time, depends only on the current state M_n, but not on previous states nor on previous inter-jump times. Again, if we would interpret jumps (transitions) of $\{M_n\}$ as signalling arrivals, we would have dependence in the arrival process. Also, unlike the Markov process case, the inter-arrival times can be arbitrarily distributed, and these distributions depend on the state of the Markov process.

Markov-modulated processes

Markov-modulated models constitute an extremely important class of traffic models [23]. The idea is to introduce an explicit notion of state into the description of a traffic stream – an auxiliary Markov process is evolving in time and its current state controls (modulates) the probability law of the traffic mechanism.

Let $\{M(t)\}_{t=0}^{\infty}$ be a continuous time Markov process, with state space of $1, 2, ..., m$ (more complicated state spaces are possible). Now assume that while M is in state k, the probability law of traffic arrivals is completely determined by k, and this holds for every $1 \leq k \leq m$. Note that when M undergoes transition to, say, state j, then a new probability law of arrivals is modulated by the state of M.

Certainly, the modulating process can be more complicated than a Markov process (so the holding times need not to be restricted to exponential random variables), but such models are far less analytically tractable. For example, Markov Renewal processes constitute a natural generalization of Markov-modulated processes with generally-distributed inter-arrival times, but those will not be reviewed here.

Markov-modulated Poisson processes

The most commonly used Markov-modulated model is the MMPP (Markov-Modulated Poisson Process) model, which combines the simplicity of the modulating (Markov) process with that of the modulated (Poisson) process [23]. In this case, the modulation mechanism simply stipulates that in state k of M, arrivals occur according to Poisson process at rate λ_k. As the state changes, so does the rate.

MMPP models can be used in a number of ways. Consider first a single traffic source with variable state. A simple traffic model would quantize the rate into a finite number of rates and each rate would give rise to a state in some Markov modulating process. Certainly, it remains to verify that exponential holding times are an appropriate description, but the Markov transition matrix $Q = [Q_{ij}]$ of the putative M can be easily estimated from empirical data. Simply quantize the empirical data, and the estimate Q_{kj} by calculating the fraction of states that M switched from state k to state j.

As an example, consider a two-state MMPP model, where one state is an "on" state with an associated positive Poisson rate, and the other state is an "off" state with associated rate zero (such models are known as interrupted Poisson for obvious reasons). These models have been widely used to model voice sources; the "on" state corresponds to a talk spurt (when the speaker emits a sound), and the "off" state corresponds to a silence (when the speaker takes a break). This basic MMPP can be extended to aggregations of independent traffic sources, each of which is an MMPP, modulated by an individual Markov process M_i, as described above. Let $J(t) = (J_1(t), J_2(t), ..., J_r(t))$, where $J_i(t)$ is the number of active sources of traffic type i, and let $M(t) = (M_1(t), M_2(t), ..., M_r(t))$ be the corresponding vector-valued Markov process taking values on all r-dimensional vectors with non-negative integer components. The arrival rate of class i in state $(j_1, j_2, ..., j_r)$ of $M(t)$ is $j_i \lambda_i$.

Markov modulated fluid process models

The fluid paradigm dispenses with individual traffic units. Instead, it views traffic as a stream of fluid, characterized by a flow rate (e.g. bits per second), so that a traffic count is replaced by a traffic volume [23].

Fluid models are appropriate to cases where individual units are numerous relative to a chosen time scale. Put differently, an individual unit is by itself of vanishingly little significance, just as one molecule more or less in a water pipeline has but an infitesimal effect on the flow. For example, the analogy of an ATM cell to a fluid molecule is a plausible one. To further highlight the analogy, contrast an ATM cell with a much bigger transmission unit, say, a compressed high-quality video frame, which consists of the order of a thousand cells. A traffic stream of coded frames should be modeled a discrete stream of arrivals, since such frames are typically transmitted at the rate of 30 frames per second. However, a fluid model is appropriate for the constituent cells.

An important advantage of fluid models is their conceptual simplicity. But important benefits will also accrue to a simulation model of fluid traffic. To see that, consider again a broadband ATM scenario. If one is to distinguish among cells, then each of them would have to count as an event. The time granularity of event processing would be quite fine, and consequently, processing cell arrivals would consume vast CPU and possible memory resources, even on simulated time scales of minutes. A statistically meaningful simulation may often be infeasible. A fluid simulation would assume that the incoming fluid flow remains (roughly) constant over much longer time periods. Traffic fluctuations are modeled by events signalling change of flow rate. As these changes can be assumed to happen far less frequently than individual cell arrivals, one can realize enormous savings in computing. In fact, infeasible simulations of cell arrival models can be replaced by feasible simulations by fluid models of comparable accuracy. In a queuing context, it is easy to manipulate fluid buffers. Furthermore, the waiting time concept simply becomes the time is takes to serve (clear) the current buffer. Since fluid models assume a deterministic service rate, these statistics can be readily computed. Typically, though, larger traffic units (say coded frames) are of greater interest than individual cells. Modelling the larger units as discrete traffic and their transport as fluid flow will give us the best of two worlds: we can measure waiting times and enjoy significant savings on simulation computing resources.

Typical fluid models assume the sources are bursty – of the “on-off” type. While in the “off” state, traffic is switched off, whereas in the “on” state traffic arrives deterministically at constant rate λ. For analytical tractability, the durations of the “on” and “off” periods are assumed to be exponentially distributed and mutually independent. A Markov model of a set of quantized (fluid) traffic rates was investigated in [45]. Fluid models of these types can be analyzed as Markov-modulated constant rate traffic. The host of generalizations, described above for MMPP, carries over to fluid models as well, including multiple classes of sources.

12.4.2 Regression traffic models

Regression traffic models define explicitly the next random variable in the sequence by previous ones within a specified time window and a moving average of white noise [1]. Regression models are often used in simulation. There are fundamental limits for direct application of the regression models to network queuing analysis. In this section several regression models are presented.

Autoregressive models

The Autoregressive model of order p, denoted as AR(p), has the following form [1]:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + \epsilon_t \tag{12.13}$$

where ϵ_t is white noise, ϕ_j are real numbers, and X_t are prescribed correlated random variables. If ϵ_t is a white Gaussian noise with variance $\sigma^2_{\epsilon_t}$, the X_t’s will be normally distributed random variables. Let us define a lag operator B as $X_{t-1} = BX_t$, and let $\phi(B)$ be a polynomial in the operator B, defined as follows: $\phi(B) = (1 - \phi_1 B - ... - \phi_p B^p)$. The AR($p$) process can be represented as

$$\phi(B)X_t = \epsilon_t \tag{12.14}$$

The process $\{X_t\}$ is stationary if the roots of $\phi(B)$ lie outside the unit circle. The autocorrelation $r(k)$ can be computed by multiplying Equation (12.13) with X_{t-k}, taking the expectation, and dividing by the variance γ_0:

$$r(k) = \phi_1 r(k-1) + \phi_2 r(k-2) + ... + \phi_p r(k-p), k > 0 \tag{12.15}$$

Thus, the general solution is:

$$r(k) = A_1 G_1^1 + A_2 G_2^2 + ... + A_p G_p^p \tag{12.16}$$

where G_i^{-1}'s are the roots of $\phi(B)$. Therefore, the autocorrelation function of AR(p) will consist, in general, of damped exponentials, and/or damped sine waves depending on whether the roots are real or imaginary.

Since successive video frames do not vary much visually, AR models have been used to model the output bit rate of VBR encoders. The video source is approximated by a fluid flow model. In the model, the output bit rate within a frame period is constant and changes from frame to frame according to the following AR(1) model:

$$\lambda(n) = \phi\lambda(n-1) + b\epsilon(n) \tag{12.17}$$

where $\lambda(n)$ is the bit rate during frame n and $\epsilon(n)$ is Gaussian white noise. $\epsilon(n)$ is chosen such that the probability of $\lambda(n)$ being negative is very small. Since the number of bits in frame n cannot be negative the value of $\lambda(n)$ in Equation (12.17) is set to zero, whenever $\lambda(n)$ is negative. This model, cannot capture abrupt changes in the frame bit rates that occur due to scene changes or visual discontinuities. Therefore, one may model the bit rate of frames within a scene as an AR process and model the scene changes by an underlying Markov chain.

Autoregressive moving average models

An Autoregressive Moving Average model of order (p, q), denoted as ARMA(p, q), has the form [1]:

$$X_t = \phi_q X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - ...\theta_q \epsilon_{t-q} \tag{12.18}$$

which can be equivalently represented as:

$$\phi(B) = \theta(B)\epsilon_t \tag{12.19}$$

where B and $\phi(B)$ are defined as previously, and $\theta(B) = (1 - \theta_1 B - ... - \theta_q B^q$.

This is equivalent to filtering a white noise process ϵ_t by a causal linear shift time invariant filter having a rational system function with p poles and q zeros; that is,

$$H(z) = \frac{B_q(z)}{A_p(z)} = \frac{1 - \sum_{k=0}^{q} \theta_k z^{-k}}{\sum_{k=0}^{p} \phi_p z^{-k}} \tag{12.20}$$

The autocovariance γ_k of the ARMA(p, q) process can be obtained by multiplying Equation (12.18) with X_{t-k}, taking the expectation , and finding the cross-correlation between ϵ_t and X_t

$$\gamma_k = \phi_1 \gamma_{k-1} + ... + \phi_p \gamma_{k-p} - \sigma_\epsilon^2 (\theta_k h_0 + \theta_{k+1} h_1 + ... + \theta_q h_{q-k}) \tag{12.21}$$

where h_t is the impulse response of the ARMA(p, q) filter H(z).

Note that $\theta_k = 0$ for $k > q$, therefore, the auto-correlation of the process for $k > q$

$$r(k) = \phi_1 r(k-1) + \phi_2 r(k-2) + ... + \phi_p r(k-p), \text{for } k > q \tag{12.22}$$

which is the same difference equation as Equation (12.15), therefore the autocorrelation of the ARMA(p, q) decays exponentially.

An ARMA model have been used to model VBR traffic. The duration of the video frame is equally divided into m time intervals. The number of cells in the n-th time interval is modeled by the following ARMA process:

$$X_n = \phi X_{n-m} + \sum_{k=0}^{m-1} \theta_k \epsilon_{n-k} \tag{12.23}$$

Since video data will correlate at each frame due to temporal correlation, the autocorrelation function has peaks at all lags which are integer multiples of m. In the above model, the AR part is used to model the recorrelation effect and the θ_k's are used to for the correlation at other lags.

The parameter estimation of ARMA models are more involved that that of AR models, the estimation of the θ_k's require solving a set of non-linear equations or using spectral factorization techniques.

12.4.3 Models for long-range-dependent processes

Since the introduction of the notion of "self-similarity" into the field of teletraffic engineering, a number of source models exhibiting LRD and their queueuing behavior, have been analyzed by various authors. It is an open question which model is best suited for describing for example MPEG video.

Fractional Brownian Motion (FBM)

FBM is defined as a stochastic process Z_t with the following properties [50]:

- Z_t has stationary increments;
- $Z_0 = 0$ and $E[Z_t] = 0$ for all $-\infty < t < \infty$;
- $\mathit{Var}(Z_t) = |t|^{2H}$ for all $-\infty < t < \infty$;
- Z_t has continuous paths;
- Z_t is Gaussian, i.e. all its finite-dimensional marginal distributions are Gaussian.

The arrival process A_t is defined as $A_t = mt + \sqrt{am}Z_t, t \in (-\infty, \infty)$, where m is the mean rate, a is a variance coefficient and Z_t an FBM. The arrival process A_t is exactly self-similar. The parameters H and a characterize the "quality" of the traffic in contrast to the long run mean rate m which characterizes the "quantity" alone.

The aggregation of two FBM streams with mean rates m_1 and m_2, variance coefficients a_1 and a_2 and Hurst parameters H_1 and H_2 yields an new FBM stream with mean rate $m_1 + m_2$, variance coefficient $(m_1 a_1 + m_2 a_2)/(m_1 + m_2)$ and Hurst parameter Max(H_1, H_2).

ON-OFF sources with heavy tails

A superposition of many ON-OFF sources with ON (or OFF) period durations following a heavy tail distribution with infinite variances (such as the Pareto distribution) produces aggregate traffic that is self-similar [9]. A large, i.e. infinite number of such sources, may lead to an FBM process.

12.5 Application of packet traffic models

12.5.1 Voice

A packetized voice source can be accurately modeled as an ON/OFF Markov source with exponentially distributed ON and OFF period durations [52]. The generation of packets in the ON state can be modeled by a Poisson process or a constant (fluid) rate.

12.5.2 Video

Recent measurements have shown that compressed video (e.g. MPEG) traffic exhibit self-similar and behavior on a wide range of time scales [25]. It have been shown [19] that there is a knee-point of the buffer size below which the buffer distribution is dominated by the short-range correlations and above which by long-range correlations. The conclusion is that if the queue operates under conditions of low utilization and practical buffer size, the input traffic can be modeled by short-range dependent model which is generally Markovian although the input traffic has long-range dependence.

For queuing analysis purposes, the video source can be modeled by a superposition of on/off Markov fluid sources or by a superposition of two-state MMPP sources. With proper matching of parameters of the autocorrelation function, the set of MMPP sources is able to produce self-similar traffic over a certain range of time scales [2]. The FBM model is another self-similar traffic model that has been suggested for compressed video.

Regression models (AR,ARMA) can be used for simulation of compressed video..

12.5.3 Audio

From a traffic modelling perspective compressed audio (e.g. MP3) is similar to compressed video. That is, regression models (AR,ARMA) can be used for simulation, and a superposition of Markov modulated fluid or Poisson sources can be used for queuing analysis [52]. However, measurements on audio traffic have failed to show self-similarity.

12.5.4 LAN

Measurements on LAN Ethernet traffic at Bellcore in the early 90s showed that Ethernet packet traffic is self-similar [43]. The aggregate Ethernet traffic can be modeled by a FBM process or by a set of ON/OFF sources with heavy tailed ON/OFF period durations.

12.5.5 WAN

Measurements on WWW and TELNET packet traffic in the Internet conducted in the mid 90s showed that such traffic is self-similar [54]. Plausible traffic models are the FBM model or a superposition of ON/OFF sources with heavy tailed ON/OFF period durations. Similar measurements on FTP data bursts within FTP sessions has shown that the FTP data burst sizes are heavy tailed.

Chapter 13

Call admission control and routing

Call Admission Control (CAC) is used for networks which provide QoS guarantees such as circuit-switched networks, ATM networks and QOS enhanced IP networks. CAC is part of preventive traffic control which also includes traffic enforcement. The purpose of CAC is to decide for each new call request, if the call should be accepted or if it must be rejected. The decision is based on three criteria:

- the QoS requirements for network calls,
- the GoS requirements for network call classes,
- the call reward requirements.

CAC is realized by two sub functions: CAC$_{QoS}$ and CAC$_{GoS}$. The purpose of the CAC$_{QoS}$ function is to decide whether a particular path is expected to offer sufficient QoS to existing calls as well as the new call. If sufficient resources are expected, the call may be accepted on the particular path, otherwise it must be rejected. The purpose of the CAC$_{GoS}$ function is to maintain the GoS and maximize the average reward rate. Different call types will be charged differently and will thus generate a different reward for the network. The CAC$_{GoS}$ function decides if carrying a call on a certain path is economical or not. For example, assume the networks is offered one category of narrow-band calls with a small reward, and one category of wide-band calls with a large reward. In this case, the CAC$_{GoS}$ function may decide to reject narrow-band calls when the path just has free capacity to one more wide-band call.

The CAC function is carried out along with the routing function. The routing function chooses only a path which passes the CAC$_{QoS}$ test. Moreover, it is up to the CAC$_{GoS}$ function to decide whether the path recommended by routing function is consistent with GoS requirements and long-term objectives on average reward.

13.1 Routing in circuit-switched networks

In this section we describe routing in non-hierarchical circuit-switched networks. The algorithms also apply to packet-switched networks with virtual circuits such as ATM networks. Even QoS enhanced IP networks which use the concept of flows, which are the equivalents of ATM virtual circuits, may use the routing algorithms designed for circuit-switched networks. The crucial part is that the network reserves resources before call set up and that the the usage of resources is enforced throughout the lifetime of the calls.

The network is assumed to be offered traffic from K call classes. Call class j is described by:

- Origin-destination node pair
- Call arrival process parameters
- Call holding time distribution parameters
- Link s bandwidth requirement b_j^s [Mbps]
- Reward parameter $r_j \in (0, \infty)$
- Set of alternative routes W_j

The bandwidth requirement for class j on link s, b_j^s, is given by the peak bandwidth requirement in case of deterministic multiplexing, and by the *equivalent capacity* in case statistical multiplexing is employed. Note that the equivalent capacity of a call may be different on different links along the call's path. In particular, the equivalent capacity generally depends on the current mix of calls on the link, as well as link and buffer capacities.

In this section we assume the class-j call arrival process is Poisson with rate λ_j and the call holding times are exponentially distributed with mean $1/\mu_j$.

Various routing algorithms have been proposed for circuit-switched networks:

- Fixed non-alternative routing
- Fixed load sharing routing
- Sticky random routing
- Fixed alternative routing
- Least loaded routing
- Markov decision process routing

All but the first two algorithms use alternative routing. In fixed non-alternative routing there is only one path between the source and destination. In alternative routing there is a set of possible paths. Typically, the call is first offered to the direct path. If the direct path is busy, the call is offered to other (non-direct) paths. For example, in a fully connected network the direct path is the one link path and the non-direct paths are multi-link paths consisting of two or more links.

The performance of alternative routing depends on the amount of offered traffic T. In the case of Poisson traffic we get:

$$T = \sum_{j \in J} \frac{\lambda_j b_j}{\mu_j} \tag{13.1}$$

Alternative routing will improve the routing performance (reduce the overall call blocking probability) when the offered traffic T is low or moderate. When T increases alternative routing will be increasingly inefficient. Recall that alternative routed calls will use two or more links. On a particular link, direct routed calls and alternative routed calls will share the capacity. When T is high there is a high risk that the link will fill up with many alternative routed calls forcing direct calls to be rejected. Moreover, since an alternative call occupies multiple links they could “steel” the capacity for many direct calls. However, when T is low or moderate the acceptance of alternative routed calls will not severely affect the blocking probability of directly routed calls.

Different methods are used to control the sharing of direct and alternative routed calls on the links. The *trunk reservation* method is used to restrict the access of alternative routed calls when the link load is high. In particular, when the free capacity is less or equal to a threshold

t_j the link will not accept alternative routed calls. Another method is called *external blocking*. With this method only a fraction ϕ_j of the calls that are rejected on the direct path are allowed to request a non-direct path. Trunk reservation and external blocking are part of CAC$_{GoS}$. The best overall blocking performance is achieved if trunk reservation and external blocking are combined.

Another issue is the distribution of blocking probabilities, or access fairness, among the call classes. Without any special control mechanisms, calls with higher bandwidths demands (i.e. wide-band calls) will suffer higher blocking probabilities than calls with lower bandwidth demands (i.e. narrow-band calls). To level out the blocking probabilities trunk reservation can be used to protect wide-band calls from bandwidth depletion. That is, direct or alternative routed narrow-band calls are rejected when the free capacity is below or equal to a threshold s_j. Trunk reservation to level out the per-class call blocking probabilities is part of CAC$_{GoS}$.

13.1.1 Fixed non-alternative routing

In fixed non-alternative routing calls are only offered to a single path [39]. If the single path is busy the call is rejected. Trunk reservation may be used to protect wide-band calls from overload of narrow-band calls. Optimal fixed routes can be determined by solving a constrained optimization problem. In short this optimization tries to assign call flows to paths and links so that the unused capacity is minimized under flow conservation and capacity constraints.

The call blocking probability for the classes can be determined by solving the Erlang fixed point equations, which were introduced in the 1960s. The method is also called reduced load approximation. Here we describe this algorithm in the case with no trunk reservation . We assume that narrow-band calls arrive to route r as Poisson process with rate λ_r and have unit mean call holding time. The blocking for calls arriving to route r is computed as:

$$L_r = 1 - \prod_{s \in r} (1 - B_s) \tag{13.2}$$

where s denotes a link index and B_s denotes the blocking probability on link s. The B_s's are obtained by solving the set of equations by repeated substitutions:

$$\begin{aligned}\rho_s &= \sum_{r:s\in r} \lambda_r \prod_{j\in r-\{s\}} (1-B_j)\\ B_s &= E\left(\rho_s, C_s\right), s = 1,2,...,L.\end{aligned} \tag{13.3}$$

where λ_r denotes the call arrival rate to route r, C_s denotes the capacity of link s, L denotes the number of links in the network, and E denotes the Erlang blocking formula:

$$E(\rho, C) = \frac{\rho^C}{C!}\left[\sum_{n=0}^{C} \frac{\rho^n}{n!}\right]^{-1} \tag{13.4}$$

which determines the proportion of lost calls on a single link with capacity C which is offered ρ Erlang of Poisson traffic. In the repeated substitutions, the link arrival rate ρ_s is first computed for every link s. Second, the link blocking probability B_s for each link is computed. The process usually converges in a few tens of steps.

A call is only offered to a link if no links along the route r blocks the call. All routes r passing through link k contributes to the call arrival rate. Howevever, the call arrival rate to link k is "thinned" due to possible blocking on other links in the routes. This is the motivation behind the name "reduced load approximation".

13.1.2 Fixed load sharing routing

In fixed load sharing routing the call is routed according to a set of fixed load sharing probabilities which assigns probabilities for selecting different routes for a call from each class [39]. Calls are only offered to one path, and if this path is busy, the call is blocked. Also here, trunk reservation may be used to protect wide-band calls from overload of narrow-band calls. Optimal load sharing probabilities can be determined from optimization. A common approach is to choose the probabilities such that the average reward rate is maximized. The call blocking probability can be computed using the reduced load approximation. After determining L_r^j, the blocking probability of route r for class j, the network blocking probability for class j is:

$$L_j = \sum_{r\in R_j} h_{jr} L_r^j \tag{13.5}$$

where R_j denotes the set of possible routes for class j, and h_{jr} is a load sharing coefficient, i.e the probability of choosing route r for a class j call.

13.1.3 Sticky random routing

Sticky random routing provides a simple form of alternative routing [26]. The algorithm works as follows. The call is first offered to the direct path. If it is available, the call is established. Otherwise, the call is offered to a previously chosen alternative path. If the alternative path is available, the call is established. Otherwise, the call is blocked and the alternative path is reselected. The new alternative route is used by the next call that find the direct path busy.

Trunk reservation and external blocking may be used to protect direct-path calls from overflow of alternative calls. As before, trunk reservation can also be used to obtain fairness between the call classes. Optimal trunk reservation thresholds and external blocking factors can be found by evaluating the blocking and average reward rate performance for different parameter values.

The main design issue in sticky random routing is the choice of algorithm for re-selection of the alternative path. A commonly used algorithm is just to chose the path at random, according to uniform probabilities.

The blocking of call classes can be determined from a reduced load approximation.

British Telecom employed in 1996 a sticky random routing algorithm called Dynamic Alternative Routing (DAR) in the British telephone network.

13.1.4 Fixed alternative routing

Fixed alternative routing chose paths for new calls according to fixed alternative sequences [27]. Trunk reservation and external blocking, can be used as described before. The first path in the sequence to be tried is the direct path. If it is busy, the remaining paths are tried, in order, until a path that is able to accept the call is found, or there is not more paths in the sequence in which case the call is rejected.

The optimal sequence with n paths can be determined by evaluating the average reward rate of different sequence choices. To this end, the call blocking is found from a reduced load approximation. The routing table gives for each node pair p the ordered sequence of alternative paths $p(1), p(2), ..., p(M)$. Recall that in order for a call to be offered to a given path in the sequence, all prior paths in the sequence must have rejected the call. Hence, the carried call arrival rate, $\overline{\nu}_p$, for OD pair p is:

$$\overline{\nu}_p = \sum_{m=1}^{M} \lambda_p \left\{ \prod_{n=1}^{m-1} L_{p(n)} \right\} \{1 - L_{p(m)}\} \tag{13.6}$$

where $L_{p(m)}$ denotes the blocking probability on path $p(m)$, and λ_p denotes the offered call arrival rate to node pair p.

The average reward is computed as:

$$\overline{R} = \sum_p \overline{\nu}_p r_p \tag{13.7}$$

The Dynamic Nonhierachical Routing (DNHR) employed by AT&T in the US telephone network during the 1980s was based on fixed alternative routing.

13.1.5 Least loaded routing

The least loaded routing (LLR) method is the first example of state-dependent alternative routing [3]. The call is first offered to the direct path. If it is busy, an alternative path is searched for according to a state-dependent routing rule. If no such path is found the call is rejected. The state-dependent routing rule selects a path with sufficient capacity that also has the largest free capacity of its bottleneck link. The bottleneck link is the link with least free capacity along the path.

Trunk reservation and external blocking may be used for protection of direct-routed calls and wide-band calls.

Call blocking can be found from reduced load approximation. The predicted call blocking probabilities are fairly close to simulation results.

Real-Time Network Routing (RTNR) replaced DNHR in the AT&T network in 1991. RTNR is based on LLR with six load states. The load on the links is investigated in real-time for every call request.

13.1.6 Markov decision process routing

Markov decision process (MDP) routing is the second example of state-dependent alternative routing [18]. Given a model of the call traffic, MDP theory is able to compute a state-dependent routing rule which achieves *optimal* average reward rate. The traffic model typically assumes a Poisson call arrival process and exponentially distributed call holding times.

A set of state-dependent *gain functions* controls the selection of routes for new calls. The gain $g_j^k(y, \pi)$ measures, for a given call class j, the increase in long-term reward when choosing a particular path k in network state y under routing rule π. The routing rule selects the path with has the largest positive gain for the new call. If no path has a positive gain, the call is rejected.

To reduce the complexity, the network is decomposed into a set of links assumed to change state independently of other links. The state of a link is the vector with the number of accepted calls from each class as components. As calls arrive and departs the state of the link will change. The sequence of states can be described by a Markov chain with certain probabilities for jumping between adjacent states. The state transition probabilities are controlled by decision variables which specify whether the link call should be accepted or rejected in the particular state. In each state, reward is earned at a characteristic rate. When the link changes state the reward is delivered to the network.

The idea of MDP routing is simple: control the Markov chain such that it visits high income states more often than low income states. The optimal set of decisions can be computed using the Policy Iteration algorithm from Markov decision theory. One example of optimal decisions on the link level is intelligent blocking of narrow-band calls. Assume that serving a wide-band call gives larger reward than serving a narrow-band call. Now, when the link precisely has free capacity to one more wide-band call, no more narrow-band calls should be accepted, since it will be more economical to wait for a wide-band call.

No trunk reservation or external blocking is used in MDP routing – the MDP approach makes its own tradeoff between direct and alternative routed calls, and between narrow-band and wide-band calls.

The call blocking under MDP routing can be computed using an extended reduced load approximation. The accuracy compared to simulation is fairly good. The main drawback is the computational complexity which increases rapidly as the number of nodes and routes increases in the network.

The MDP routing algorithm was implemented the Bell Canada telephone network in the 1990s.

13.2 Routing in ATM networks

The PNNI (Private Network to Network Interface) is the protocol that enables the building of multi-vendor, interoperable ATM networks. PNNI is a hierarchical, dynamic link-state routing protocol for building large-scale ATM-based networks [5]. In addition, PNNI defines signalling requests to establish point-to-point or point-to-multipoint connections.

PNNI organizes switching systems into logical collections called peer group. Neighboring nodes form a peer group by exchanging their peer group identifiers (PGIDs) via Hello packets using a protocol that makes nodes known to each other. A border node has at least one link that crosses the peer group boundary. Hello protocol exchange occurs over logical links (physical link or VPC or SVCC). PNNI defines the creation and distribution of a topology database that describes the elements of the routing domain as seen by the node. This database provides all the information required to compute a route from the node to any address that is reachable in or through that routing domain. Nodes exchange database information using PTSEs (PNNI Topology State Elements). PTSEs contain topology characteristics derived from link or node state parameter information. The state parameter information consists of metrics and attributes.

The metrics include:

- Cell loss ratio (CLR)
- Maximum cell transfer delay (maxCTD)
- Cell delay variation (CDV)
- Administrative weight (AW)

The attributes include:

- Available cell rate (avCR)

- Cell rate margin (CRM) = avCR-sustainable cell rate (SCR) (optional)
- Variation factor (VF)=CRM/Stdev(SCR) (optional)
- Branching flag: Can handle point-to-multipoint traffic
- Restricted transit flag: Supports transit traffic or not

PTSEs are grouped to form PTSP (PNNI Topology State Packet) and PTSPs are flooded throughout the peer group so all nodes in one peer group will have an identical database. Each node belonging to a given peer group has complete knowledge of the group topology and the state of the related portion of the network, but keeps only a summarized information about all other groups. Every peer group has a node called peer group leader (PGL). There is at most one active PGL per peer group. The PGL will represent current peer group in the parent peer group as a single node. The PGL will also flood the PTSEs in the parent peer group to the current peer group. Apart from its specific role in aggregation and distribution of information for maintaining the PNNI hiearchy, the PGL does not have any special role in the peer group. Currently, PNNI supports 104 hierarchical levels.

Call establishment in PNNI consists of two operations: the selection of a path and the setup of the connection state at each point along that path. PNNI uses source routing for all connection set up requests. In source routing the source selects the path to the destination and the visited systems on the path obey the source's routing instruction. The path is encoded as a Designated Transit List (DTL) which is explicitly included in the connection set up request. The paths are computed on an on-demand basis and the algorithm for determining them is not standardized. However, it is envisaged that PNNI will use some form of state-dependent dynamic routing strategy.

Figure 13.1 shows an hierarchical ATM network with two levels.

13.3 Routing in IP networks

Internet consists of a huge number of individual IP subnetworks with unique network addresses (about 120,000 in 2002). To make the best effort routing in Internet manageable, the IP subnetworks are divided into a set of Autonomous Systems (ASs). An AS is a portion of the network which is under control of a single administrative unit such as a campus.

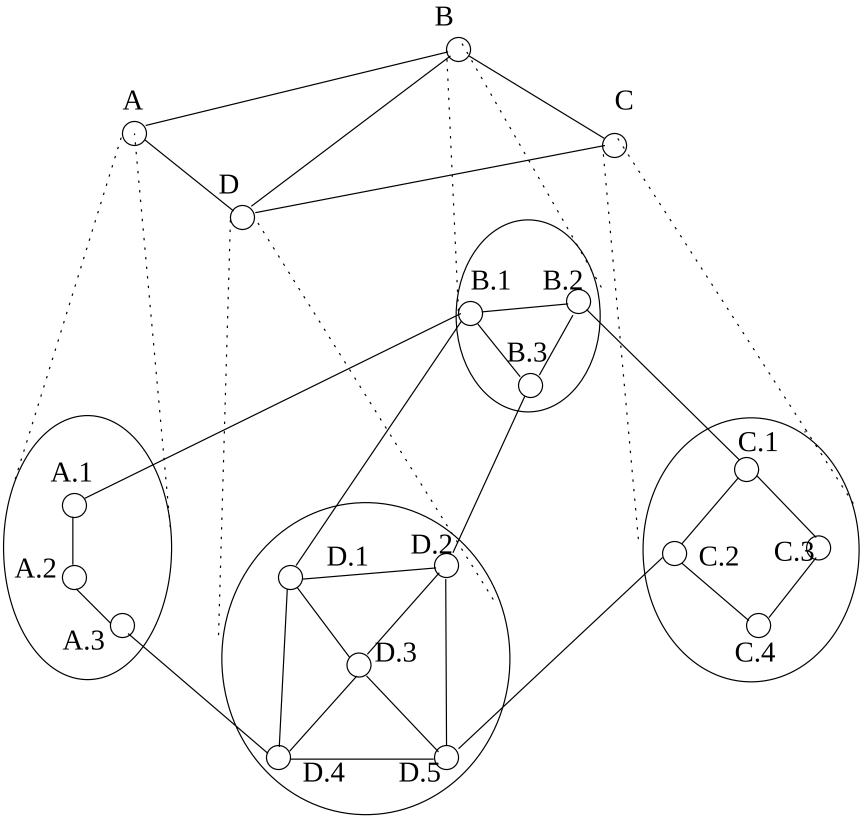


Figure 13.1: Example of a PNNI network

Today (2002) Internet has about 13,000 ASs. Special routing protocols are used within the ASs and between the ASs, called intra-domain routing and inter-domain routing protocols, respectively [55].

The intra-domain routing protocols are either based on distance vector routing or on link state routing. The protocols use some algorithm to compute *shortest path routes* between the source and destination. The context for the shortest path routing is a network with two weight associated with each link – one weight for each direction of the link. The weights measure the "distance " or cost of crossing the link. The weight could be either packet delay, bandwidth, reliability or some link cost. Figure 13.2 shows an example network depicted as a graph with links weights.

Inter-domain routing is more focused on finding loop-free paths which can reach the destinations than on finding optimal paths. The ASs use a set of policies to guide the selection of paths.

The future Internet will not only offer best effort service but also QoS service. To meet this goal Internet has to be extended with resource reservation, traffic enforcement and QoS intra- and inter-domain routing. QoS routing finds paths which meets multiple constraints on path bandwidth, delay, jitter, loss, cost etc.

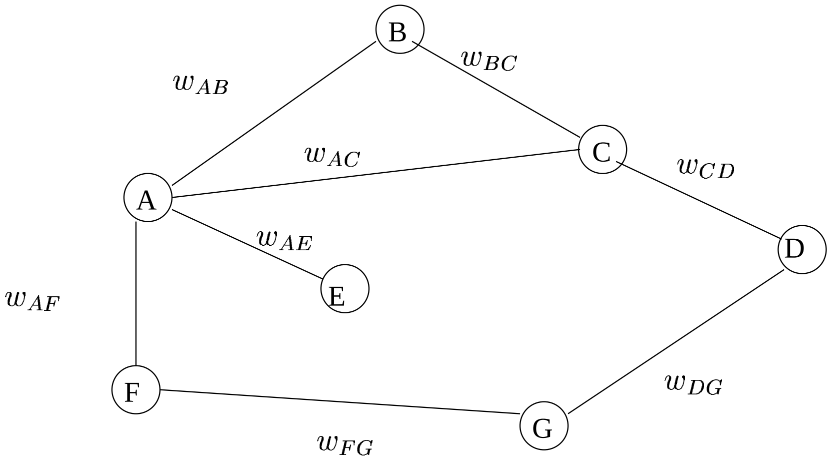


Figure 13.2: Example network graph with link weights. Only one of the two weights for each bi-directional link is shown.

13.3.1 Distance vector routing

Distance-vector routing algorithms operate by having each router maintain a table giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors.

In distance-vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the domain. This entry contains two parts; the proffered outgoing line to use for that destination, and an estimate of the cost to that destination. The starting assumption for distance vector routing is that each node knows the weight of the link to each of its directly connected neighbors. A link that is down is assigned an infinite weight.

To see how a distance-vector routing algorithm works, we consider the network example in Figure 13.2. In this example, the weight/cost w_{ij} of each bi-directional link is set to 1, so that a shortest path is simply the one with fewest hops. We can represent each node's knowledge about distances to all other nodes as a table like the one given in the table in Figure 13.3. Note that each node only knows the information in one row of the table. The global view that is represented here is not available at any single point in the network.

Information stored at node	Distance to reach node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Figure 13.3: Initial distances stored at each node (global view)

We may consider each row in the table in Figure 13.3 as a list of distances from one node to all other nodes, representing the current beliefs of that node. Initially, each node sets a

Destination	Cost	Next Hop
B	1	B
C	1	C
D	∞	–
E	1	E
F	1	E
G	∞	–

Figure 13.4: Initial routing table at node A.

weight of 1 to its directly connected neighbors and ∞ to all other nodes. Thus, A initially believes that it can reach B in one hop and that D is unreachable. The routing table stored at A reflects this set of beliefs and includes the name of the next hop that A would use to reach any reachable node. Initially, then, A's routing table would look like Table 13.4.

The next step in distance-vector routing is that every node sends a message to its directly connected neighbors containing its personal list of distances. For example, node F tells node A that it can reach node G at a cost of 1; A also knows it can reach F at a cost of 1, so it adds these costs to the the cost of reaching G by means of F. This total cost is 2 is less than the current distance of infinity, so A records that it can reach G at a cost of 2 by going through F. Similarly, A learns from C that D can be reached from C at a cost of 1; it adds this to the cost of reaching C and decides that D can be reached via C at at cost of 2, which is better than the old distance of infinity. At the same time, A learns from C that B can be reached from C at a cost of 1, so it concludes that the cost of reaching B via C is 2. Since this is worse than the current cost of reaching B, this new information is ignored.

At this point, A can update its routing table with distances and next hops for all nodes in the network. The result is shown in Figure 13.5.

A node sends routing updates to its neighbor periodically, on the order of several seconds to several minutes. A triggered update can also be sent if necessary. It occurs when a node receives an update from one of its neighbors that causes it to change one of its routes in its

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	E
G	2	F

Figure 13.5: Final routing table at node A.

routing table.

To understand what happens when a node detects a link failure, consider that happens when F detects that its link to G has failed. First, F sets its new distance to G to infinity and passes that information along to A. Since A knows that its 2-hop path to G is through F, A would also set its distance to G to infinity. However, with the next update from C, A would learn that C has a 2-hop path to G. Thus A would know that it could reach G in 3 hops through C, which is less than infinity, and so A would update its table accordingly. When it advertises this to F, node F would learn that it can reach G at a distance of 4 through A, which is less than infinity, and the system would again become stable.

Unfortunately, slightly different circumstances can prevent the network from stabilizing. Suppose, for example, that the link from A to E goes down. In the next round of updates, A advertises a distance of infinity to E, but B and C advertise a distance of 2 to E. Depending on the exact timing of events, the following might happen: Node B, upon hearing that E can be reached in 2 hops from C, concludes that it can reach E in 3 hops and advertises this to A; node A concludes that it can reach E in 4 hops and advertises this to C; node C concludes that it can reach E in 5 hops; and so on. This cycle stops only when the distances reach some number that is large enough to be considered infinite. In the meantime, none of the nodes actually knows that E is unreachable, and the routing tables for the network do not stabilize. This situation is known as the *count to infinity problem*.

One technique to improve the time to stabilize routing is called *split horizon*. The idea is that when a node sends a routing update to its neighbors, it does not send those routes it learned from each neighbor back to that neighbor. For example, if B has route (E,2,A) in its table, then it knows it must have learned this route from A and so whenever B sends a routing update to A, it does not include the route (E,2) in that update. In a stronger variation, called *split horizon with poison reverse*, B actually sends that route back to A, but it puts negative information in the route to ensure that A will not eventually use B to get to E. For example, B sends the route (E,∞) to A. The problem with both these techniques it that they only work for routing loops that involves two nodes. For larger routing loops, more drastic measures are called for.

Bellman-Ford algorithm

The *Bellman-Ford algorithm* is formally used to describe route calculation in distance-vector routing. The algorithm maintains a parameter d_i^h that contains the cumulative distance from a root node N_1 to node N_i for an h-hop path. Initially, the algorithm sets $d_1^h = 0$ for all h, sets $d_i^h = \infty$ for all $i \neq 1$, and sets $h = 1$. Next, the algorithm executes the following steps [46]:

1. Update cumulative distance for the current hop value h:

   $d_i^h = \text{Min}_j[w_{ji} + d_j^{h-1}]$ for all $i \neq 1$.

2. If there is no change in the cumulative distance counts (i.e. $d_i^h = d_i^{h-1}$), then stop; the algorithm has completed.

3. Increment the hop count: $h \leftarrow h + 1$

4. Go to step 1.

This algorithm begins at the root node, and then branches out toward all destinations in parallel fashion in increasing radius of hops away from the root node. Thus, when implemented in a distributed manner it acts as a distance-vector routing algorithm. The amount of calculation required for the Bellman-Ford algorithm is O(mL), where m is the maximum number of links the the shortest path and L is the number of links in the network graph. In the worst case, the computation required by the Bellman-Ford algorithm is between O(N^2) and

O(N^3) since $m \leq (N-1)$ and $(N-1) \leq L \leq N(N-1)$. Note that for sparsely connected networks, the required level of computation cam be considerably less than O(N^3).

As an example of operation of the Bellman-Ford algorithm consider the network in Figure 13.2 with all weights set to 1. The Bellman-Ford algorithm for node A as root node is as follows. Initially, $d_i^h = 0$ for all h and $d_i^h = \infty$ for all $i \neq$ A and $h = 1$. The algorithm proceed as follows:

1. $d_B^1 = \min_j[w_{jB} + d_j^0] = w_{AB} + d_A^0 = 1 + 0 = 1$
   $d_C^1 = \min_j[w_{jC} + d_j^0] = w_{AC} + d_A^0 = 1 + 0 = 1$
   $d_D^1 = \min_j[w_{jD} + d_j^0] = w_{CD} + d_C^0 = 1 + \infty = \infty$
   $d_E^1 = \min_j[w_{jE} + d_j^0] = w_{AE} + d_A^0 = 1 + 0 = 1$
   $d_F^1 = \min_j[w_{jF} + d_j^0] = w_{AF} + d_A^0 = 1 + 0 = 1$
   $d_G^1 = \min_j[w_{jG} + d_j^0] = w_{FG} + d_F^0 = 1 + \infty = \infty$

2. $d_B^2 = \min_j[w_{jB} + d_j^1] = w_{AB} + d_A^1 = 1 + 0 = 1$
   $d_C^2 = \min_j[w_{jC} + d_j^1] = w_{AC} + d_A^1 = 1 + 0 = 1$
   $d_D^2 = \min_j[w_{jD} + d_j^1] = w_{CD} + d_C^1 = 1 + 1 = 2$
   $d_E^2 = \min_j[w_{jE} + d_j^1] = w_{AE} + d_A^1 = 1 + 0 = 1$
   $d_F^2 = \min_j[w_{jF} + d_j^1] = w_{AF} + d_A^1 = 1 + 0 = 1$
   $d_G^2 = \min_j[w_{jG} + d_j^1] = w_{FG} + d_F^1 = 1 + 1 = 2$

3. $d_B^3 = \min_j[w_{jB} + d_j^2] = w_{AB} + d_A^2 = 1 + 0 = 1$
   $d_C^3 = \min_j[w_{jC} + d_j^2] = w_{AC} + d_A^2 = 1 + 0 = 1$
   $d_D^3 = \min_j[w_{jD} + d_j^2] = w_{CD} + d_C^2 = 1 + 1 = 2$
   $d_E^3 = \min_j[w_{jE} + d_j^2] = w_{AE} + d_A^2 = 1 + 0 = 1$
   $d_F^3 = \min_j[w_{jF} + d_j^2] = w_{AF} + d_A^2 = 1 + 0 = 1$
   $d_G^3 = \min_j[w_{jG} + d_j^2] = w_{FG} + d_F^2 = 1 + 1 = 2.$

   Now $d_i^3 = d_i^2$ for all i so the algorithm stops.

   Obviously, node B, C, E and F can be reached in one hop from node A, and node D and G can be reached in two hops.

13.3.2 Link state routing

The basic idea behind link-state protocols is very simple: Every node knows how to reach its directly connected neighbors, and if we make sure that the totality of this knowledge is disseminated to every node, then every node will have enough knowledge of the network to build a complete map of the network. Link-state protocols rely on two mechanisms: reliable dissemination of link-state information, and the calculation of routes from the sum of all accumulated link-state knowledge.

Reliable flooding is the process of making sure that all nodes participating in the routing protocol get a copy of the link-state information from all other nodes. As the term “flooding” suggests, the basic idea is for a node to send its link-state information out on all of its directly connected links, with each node that receives this information forwarding it out on all of *its* links. This process continues until the information has reached all the nodes in the network.

More precisely, each node creates an update packet, also called a link-state packet (LSP), that contains the following information:

- the ID of the node that created the LSP
- a list of directly connected neighbors of that node, with the cost of the link to each one
- a sequence number
- a time to live for this packet

The first two items are needed to enable route calculation; the last two are used to make the process of flooding the packet to all nodes reliable The sequence numbers are used in the following way. Consider a node X that receives a copy of an LSP that originated at some other node Y. Note that Y may be any other router in the same routing domain as X. X checks to see if it has already stored a copy an an LSP from Y. If not, it stores the LSP. If it already has a copy, it compares the sequence numbers; if the new LSP has a larger sequence number, it is assumed to be more recent, and that LSP is stored, replacing the old one. A smaller (or equal) sequence number would imply an LSP older (or not newer) that the one stored, so it would be discarded an no further action would be needed. If the received LSP was a newer one, X sends a copy of that LSP to all of its neighbors except the neighbor from which the LSP was just received. The fact that the LSP is not sent back to the node from which the LSP was received helps to bring an end to the flooding of an LSP. Since X passes the LSP to all of its neighbors who then turn around and do the same thing, the most recent copy of the LSP eventually reaches all nodes.

Just as in distance-vector routing, each node generates LSPs under two circumstances. Either the expiry of a periodic timer or a change in the topology can cause a node to generate a new LSP. However, the only topology-based reason for a node to generate an LSP is if one of its directly connected links or immediate neighbors has gone down.

The difference between distance-vector and link-state algorithms can be summarized as follows. In distance vector, each node talks only to its directly connected neighbors, but it tells them everything it has learned (i.e. distance to all nodes). In link state, each node talks to all other nodes, but it tells only what it knows for sure (i.e. only the state of its directly connected links).

Dijkstra algorithm

Once a given node has a copy of the LSP from every other node, it is able to compute a complete map for the topology of the network, and from this map it is able to decide the best route to each destination using the Dijkstra algorithm.

We now define Dijkstra in graph-theoretic terms. Imagine that a node takes all the LSPs it has received and constructs a graphical representation of the network, in which N denotes the set of nodes in the graph, $w(i, j)$ denotes the nonnegative cost (weight) associated with the edge between node $i, j \in N$, and $w(i, j) = \infty$ if no edge connects i and j. In the following description, we let $s \in N$ denote the root node, that is, the node executing the algorithm to find the shortest paths to all other nodes in N. Also, the algorithm maintains the following two variables: M denotes the set of nodes incorporated so far by the algorithm, and $C(n)$ denotes the cost of the path from s to node n. Given these definitions, the algorithm is defined as follows [55]:

```
M={ s }
FOR each $n$ in $N - \{s\}$
BEGIN
    $C(n) = w(s, n)$
END
```

```
WHILE ($N \neq M$)
BEGIN
    $M = M \cup \{m\}$ such that $C(m)$ is the minimum for all $m \in N - M$
    FOR each $n$ in $(N - M)$
    BEGIN
        $C(n) = \text{Min}(C(n), C(m) + w(m, n))$
    END
END
```

Basically, the algorithm works as follows. We start with M containing the root node s and then initialize the table of costs (the $C(n)$s) to other nodes using the known costs to directly connected nodes. We then look for the node m that is reachable at the lowest cost and add it to M. Finally we update the table of costs by considering the cost of reaching nodes through node m if the total cost of going from the source to m and following the link from m to n is less than the old route we had to n. This procedure is repeated until all nodes are incorporated in M.

The Dijkstra algorithm traces out the shortest path in only $N - 1$ iterations for an N node network. It requires at worst N computations in each iteration to sort the distance estimate. Hence, the worst-case level of computation is O(N^2),

As an example of operation of the Dijkstra algorithm consider the network in Figure 13.2 with all weights set to 1. The Dijkstra algorithm for node A as root node is as follows. Initially, $M = \{A\}$.

1. M={ A }

   N-M={ B, C, E, F, D, G}

   $C(B) = w_{AB} = 1$

   $C(C) = w_{AC} = 1$

   $C(E) = w_{AE} = 1$

   $C(F) = w_{AF} = 1$

   $C(D) = w_{AD} = \infty$

   $C(G) = w_{AG} = \infty$

2. $M = \{A, B, C, E, F\}$

   $N - M = \{D, G\}$

$$C(D) = \text{Min}[C(D), C(m) + w(m, D)] = C(C) + w(C, D) = 1 + 1 = 2$$
$$C(G) = \text{Min}[C(G), C(m) + w(m, G)] = C(F) + w(F, G) = 1 + 1 = 2$$

3. $M = \{A, B, C, D, E, F, G\}$
   $N - M = \emptyset$. The algorithm stops.

13.3.3 Intra-domain routing: RIP and OSPF

RIP

One of the most widely used routing protocols in IP networks is the Routing Information Protocol (RIP) which is based on distance-vector routing. RIP was employed in ARPANET until 1979 when it was replaced by a link state protocol. RIP use the number of hops as routing metric, i.e. all weights are set to 1. Valid distances are 1 through 15, with 16 representing infinity. This limits RIP to running on fairly small networks – those with no paths longer than 15 hops.

OSPF

OSPF was originally designed to support type of service (TOS) routing. OSPF maintains one graph for each type of metric (delay, throughput, reliability). Although this triples the computation needed, it allows separate routes for optimizating delay, throughput and reliability. The requirement of TOS routing was recently dropped due to lack of implementations.

OSPF allows multiple routes to the same place to be assigned the same cost and will cause traffic to be distributed evenly over those routes. OSPF has also support for authentification of routing messages.

Some link-state intradomain routing protocols – OSPF in particular – provide a means to partition a routing domain into subdomains called *areas* [55]. By adding this extra level of hierarchy, we enable single domains to grow larger without overburdening the intradomain routing protocols.

An area is a set of routers that are administratively configured to exchange link-state information with each other. There is one special area – the backbone area, also known as area 0. An example of a routing domain divided into areas is shown in Figure 13.6. Routers R1, R2 and R3 are members of the backbone area. They are also members of at least one

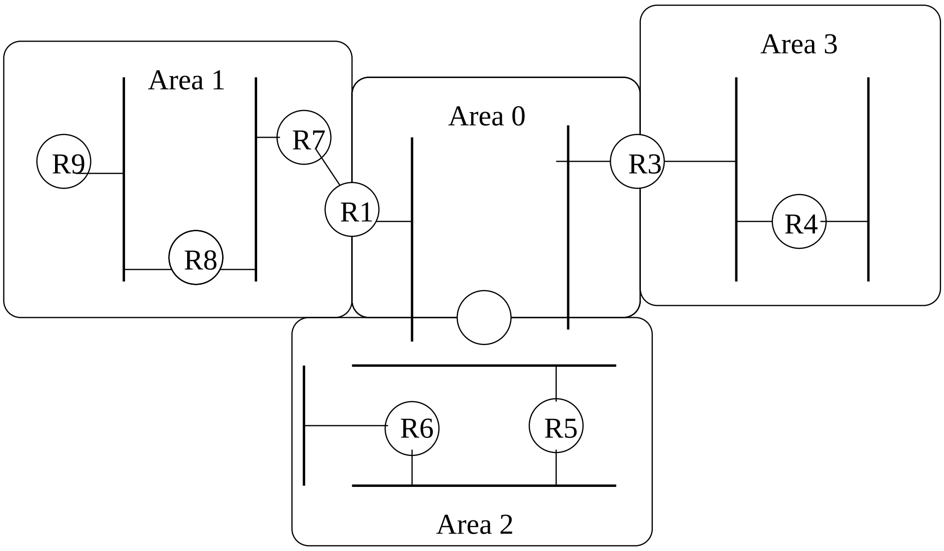


Figure 13.6: A domain divided into areas

nonbackbone area; R1 is actually a member of both area 1 and area 2. A router that is a member of both the backbone area and a nonbackbone area is an area border router. Note that these are distinct from the routers that are at the edge of the AS, whch are referred to as AS border routers for clarity.

Routing within a single area is done as previously described. All the routers in the area send link-state advertisements to each other, and thus develop a complete, consistent map of the area. However, the link-state advertisements of routers that are not area border routers do not leave the area in which they originated. This has the effect of making the flooding and route calculation process considerably more scalable. For example, router R4 in area 3 will never see a link-state advertisement from router R8 in area 1. As a consequence it will know nothing about the detailed topology of areas other than its own.

How, then, does a router in one area determine the right next hop for a packet destined to a network in another area? The answer to this become clear if we imagine the path of a packet that has to travel from one nonbackbone area to another as being split into three parts. First, it travels from its source network to the backbone area, then is crosses the backbone, then it travels from backbone to destination network. To make this work, the area border router summarize routing information that they have learned from one area and make it available in their advertisements to other areas. For example, R1 receives link-state advertisements from all the routers in area 1 and can thus determine the cost of reaching all networks in area 1.

This enables all the area 0 routers to learn to cost of all networks in area 1. The area border routers then summarize this information and advertise it to the nonbackbone areas. Thus, all routers learn how to reach all networks in the domain.

When dividing a domain into areas, the network administrator makes a tradeoff between scalability and optimality of routing. The use of areas forces all packets traveling from one area to another to go via the backbone area even if a shorter path might have been available, For example, if R4 and R5 were directly connected, packets would not flow between them because they are in different nonvbackbone areas. It turns out that the need for scalability is often more important than the need to use the absolute shortest path.

13.3.4 Inter-domain routing: BGP

The basic idea behind autonomous systems (ASs) is to provide an additional way to hierarchically aggregate routing information into a larger internet, thus improving scalability. One feature of the AS idea is that it enables some ASs to drastically reduce the amount of routing information they need to care about by using *default routes*. For example, if a corporate network is connected to the rest of the Internet by a single router, it is pretty easy for a host or router inside the AS to figure out where it should send packets that are headed for a destination outside of this AS.

Todays Internet consists of an interconnection of multiple backbone networks, and sites connected to each in arbitary ways, see Figure 13.7. Some large corporations connect directly to one or more of the backbones, while others connect to smaller, nonbackbone service providers. Many service providers exist mainly to provide service to “consumers”, and these providers must also connect to the backbone providers. Often many providers arrange to interconnect with each other or a single “peering point”. In short, it is hard to discern much structure at all in todays Internet.

Given this rough sketch of the Internet, if we define *local traffic* as traffic that originates at or terminates at nodes within an AS, and *transit traffic* as traffic that passes through an AS, we can classify ASs into three types:

- **stub AS**: as AS that has only a single connection to one other AS; such an AS will only carry local traffic.

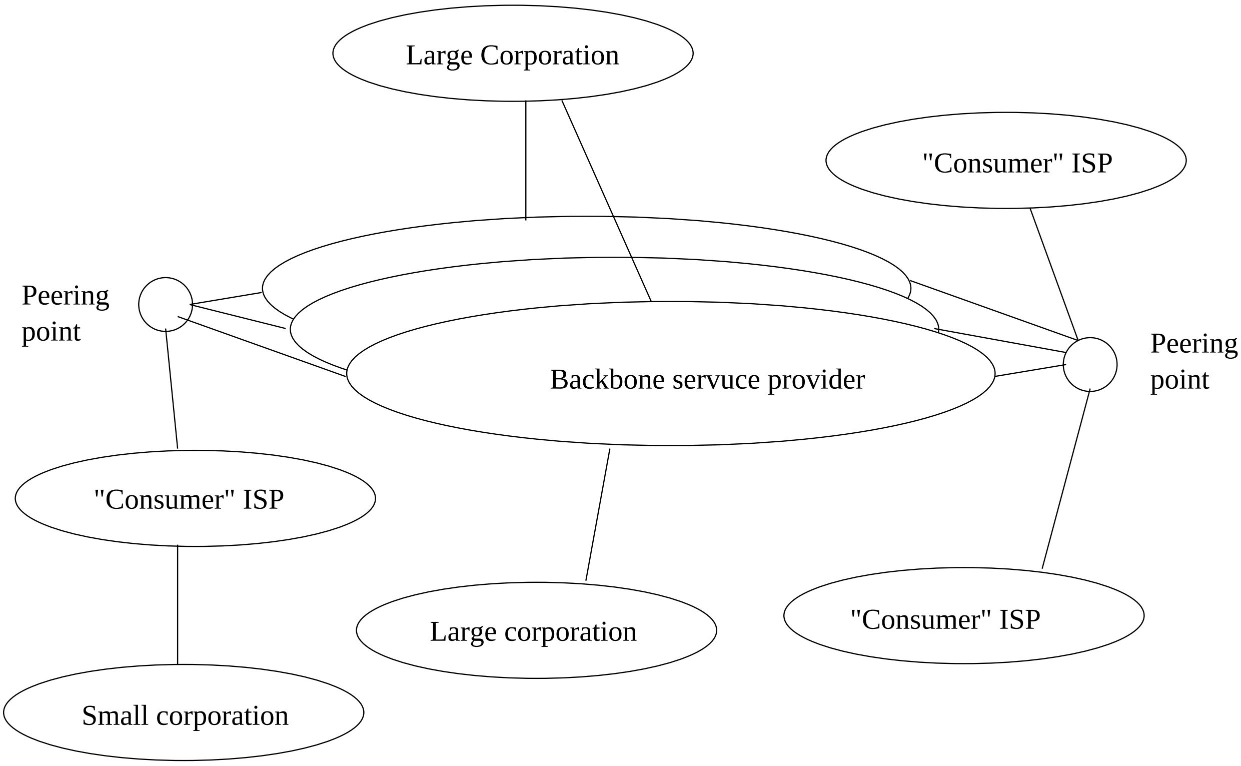


Figure 13.7: Todays backbone Internet

- **multihome AS**: an AS that has connections to more than one other AS but refuses to carry transit traffic.

- **transit AS**: an AS that has connection to more than one other As and that is designed to carry both transit and local traffic.

Whereas intra-domain routing focused on finding optimal paths based on optimizing some sort of link metric, the problem of interdomain routing turns out to be so difficult that the goals are more modest [55]. First are foremost, the goal is to find *any* path to the intended destination that is loop-free. That is, we are more concerned with reachability than optimality. Finding a path that is anywhere close to the optimal is considered a great achievement.

There are a few reasons why interdomain routing is hard. The first is simply a matter of scale. An Internet backbone router must be able to forward any packet destined anywhere in the Internet. This means having a routing table that will provide a match for any valid IP address. While CIDR has helped to control the number of distinct prefixes that are carried in the Internet's backbone routing, there is inevitably a lot of routing information to pass around

– on the order of 120,000 prefixes at the time of writing.

The second challenge in interdomain routing arises from the autonomous nature of the domains. Note that each domain may run its own interior routing protocols, and use any scheme they choose to assign metrics to paths. This means that it is impossible to calculate meaningful path costs for a path that crosses multiple ASs. As a result, interdomain routing advertises only "reachability". The concept of reachability is basically a statement that "you can reach this network through this AS.". This means that for interdomain routing it is essentially impossible to pick the optimal path.

The third challenge involves the issue of trust. Provider A might be unwilling to believe certain advertisements from provider B for fear that provider B will advertise erroneous routing information. Closely related to this issue is the need to support flexible policies in interdomain routing. For example, provider A might wish to implement policies that say "Use provider B only to reach these addresses.", "Use the path that crosses the fewest number of ASs", or "Use AS x in preference to AS y.". The goal is to specify policies that lead to "good" paths, if not to optimal ones.

The Border Gateway Protocol (BGP) is the current inter-domain routing protocol in Internet. At them time of this writing version 4 of BGP is in use. When configuring BGP, the administrator of each AS pick at least one node to be a "BGP speaker", which is essentially a spokes person for the entire AS. That BGP speaker establishes BGP sessions to other BGP speakers in other ASs. These sessions are used to exchange reachability information among ASs.

In addition to the BPG speakers, the AS has one or more border "gateways", which need not to be the same as the speakers. The border gateways are the routers through which packets enter and leave the AS.

BGP does not belong to either of the two main routing protocols (distance-vector and link-state protocols). Unlike these protocols, BGP advertises *complete paths* as an enumerated list of ASs to reach a particular network. This is necessary to enable the sorts of policy decisions described above to be in accordance with the wishes of a particular AS. It also enables routing loops to be readily detected.

All that the BGP protocol does is to specify how reachability information should be exchanged among ASs. BGP speakers obtain enough information by this exchange to calculate loop-free routes to all reachable networks, but how they choose the “best” routes is largely left to the policies of the AS.

13.3.5 QoS routing

This section describes QoS routing in the intra- and inter-domain [71, 49]. QoS routing should be combined with CAC$_{QoS}$ and CAC$_{GoS}$, resource reservation, and traffic enforcement. QoS intra-domain routing is believed to be much easier than in QoS inter-domain routing. Since QoS intra-domain routing has the responsibility of a single AS it should be possible to compute optimal paths which satisfies multiple constraints on bandwidth, delay, loss, cost etc. However, in QoS inter-domain routing it will be difficult to find optimal paths due to the reasons described in the section inter-domain routing and BGP. However, each AS should be able to answer if it can support a transit flow with given per-domain QoS requirements or not. This information can be used together with the usual set of inter-domain routing policies to determine reachable inter-domain paths.

The intra-domain QoS routing scheme should:

- inter operate with CAC$_{QoS}$ and CAC$_{GoS}$ functions
- optimize resource usage
- support best-effort flows
- support multicast QoS flows

The inter-domain QoS routing should:

- inter operate with CAC$_{QoS}$ and CAC$_{GoS}$ functions
- determine whether a destination is reachable
- map routing policies (e.g. monetary cost, usage and administrative factors) to flow metrics
- avoid routing loops
- optionally determine multiple paths to the destination, based on service classes

We now formally define the QoS intra-domain routing problem with two independent metrics which take on non-negative real numbers from the set $\mathbf{R}^+$

Definition. *Multi-constrained path problem* (MCP): Given a directed graph $G = (V, E)$, a source vertex s, a destination vertex t, two weight functions $w_1 : E \to \mathbf{R}^+$ and $w_2 : E \to \mathbf{R}^+$, two constants $c_1 \in \mathbf{R}^+$ and $c_1 \in \mathbf{R}^+$; the problem, denoted MCP$(G, s, t, w_1, w_2, c_1, c_2)$, is to find a path p from s to t such that $w_1(p) \leq c_1$ and $w_w(p) \leq c_2$ if such path exists.

A path p which satisfies $w_1(p) \leq c_1$ and $w_2(p) \leq c_2$ is called a *solution* for MCP$(G, s, t, w_1, w_2, c_1, c_2)$, We assume both weight functions are *additive* – the weight of a path is equal to the summation of the weights of all edges on the path.

It is well known that the above MCP problem is NP complete. The complexity of a NP complete problem is such that the time to find a solution increases exponentially with the size of problem. This means that the time complexity of the MCP problem increases exponentially in the number of network nodes.

However, if all metrics except one take on bounded integer values, then the problem are solvable in polynomial time by running an extended Dijkstra or extended Bellman-Ford algorithm.

If all metrics are dependent on a common metric, the problem may also be solved in polynomial time. For example, the worst-case delay and jitter are functions of bandwidth in networks using the Weighted Fair Queuing (WFQ) bandwidth scheduling. The delay-jitter-constrained routing problem is solvable in polynomial time in such networks.

Finally, the problem of finding a least-delay path that has a required bandwidth are solvable in polynomial time. The problem can be solved by a shortest path algorithm on the graph where the links violating the bandwidth constraint have been removed.

Chapter 14

Congestion control

14.1 Congestion control in ATM networks

14.1.1 Problem definition

The Available Bit Rate (ABR) service category is aimed at data applications which are delay, jitter and throughput tolerant but loss sensitive [38]. The ATM network makes no guarantees on delay and jitter for ABR connections but promises to support a certain Minimum Cell Rate (MCR) and to minimize the cell loss. Besides guaranteeing a minimum throughput, the allocation of any excess bandwidth should be fair.

The QoS of ABR connections is the concern of congestion control which is a reactive traffic control mechanism. The ABR congestion control is classified by its feedback mechanism. There exists two main types of congestion control mechanisms: hop-by-hop credit based congestion control and end-to-end rate-based congestion control. Both methods aims at achieving high network utilization by allowing ABR traffic to fully utilized leftover bandwidth not used by higher priority traffic.

Hop-by-hop credit-based congestion control works as follows. The virtual circuit (VC) from the source end-system (SES) to the destination end-system (DES) is composed of sequence of source-destination pairs interconnected by direct links. Before forwarding any data cell over the link the source needs to receive credits for the VC from the receiver. At various times, the receiver sends credits to the sender indicating availability of buffer space for receiving data cell on the VC. After receiving credits, the sender is eligible to forward some number of data cells on the VC to the receiver according to the received credit information. Each time the sender forwards a data cell of a VC, it decrements its current credit balance for the VC by one.

End-to-end rate based congestion control works as follows. ABR SESs adjust their transmission rates dynamically between a pre-determined MCR and Peak Cell Rate (PCR), based on the amount of network bandwidth left unused by higher priority service categories (CBR, rt-VBR and nrt-VBR). The rate adjustment is done using a closed-loop feedback mechanism, using Resource Management (RM) cells. RM cells convey control information to the SESs about the state of the network, such as congestion state and bandwidth availability.

Forward RM (FRM) cells are generated by the SESs and inserted into the outgoing data stream, see Figure 14.1. One FRM cells are sent periodically – one FRM cell after every N_{rm} data cell (e.g. N_{rm}=32). On their way to the DES and back from the DES to the SES, RM cells are processed by the switches. When an RM cell arrives at the DES, the destination changes the direction bit (DIR) in the cell and returns it to the SES. RM cells traveling from DES to the SES are called Backward RM (BRM) cells. BRM cells bring updated network state information to the SESs.

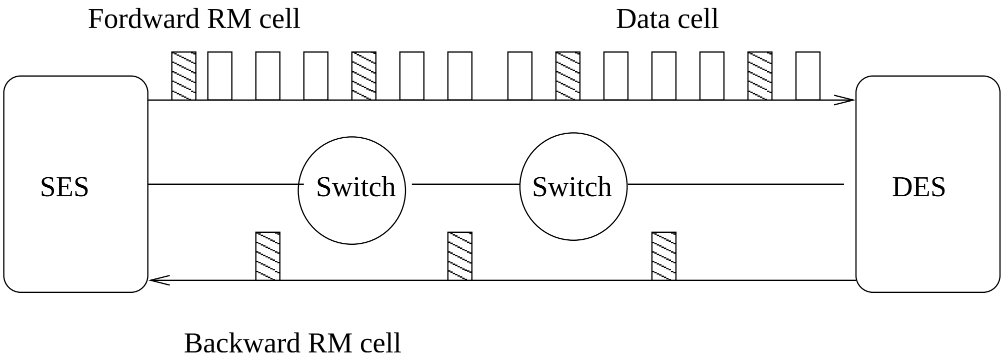


Figure 14.1: ABR rate based flow control

The rate at which an ABR source is allowed to schedule cells for transmission is denoted by ACR (Allowed Cell Rate). The ACR is initially set to the Initial Cell Rate (ICR) and is always bounded between MCR and PCR. The Current Cell Rate (CCR) value in the FRM cell is set to the ACR, and the Explicit Rate (ER) value is set to the PCR. The SES dynamically adjusts the ACR to the congestion information it receives via the BRM cells. A switch can convey information about congestion status or desired rate via RM cells in four ways.

First, a congested switch may set the Explicit Forward Congestion Indication (EFCI) bit of the Payload Type (PT) field in the ATM header of each data cell. The DES, when receiving the EFCI bit set in the data cell, sets the Congestion Indication (CI) bit of the next RM cell it sends to the SES indicating that some node along the VC is congested.

Second, a congested switch which receives a FRM cell with a larger CCR value than it can support, may reduce the Explicit Rate (ER) parameter value to the level desired. Besides continuing the RM cell forward, the switch may at that moment also generate an BRM cell with CI=1 and ER to whatever value it can support. Finally, it sets a particular, new indicator called Backward Notification (BN) bit, to indicate to the SES that this RM cell has been returned by a switch prior to the DES. This combined mechanism is commonly referred to as Backward Explicit Congestion Notification (BECN).

Third, a congested switch can directly set the CI bit of passing FRM or BRM cells. In addition, it can set the No Increase (NI) bit the FRM or BRM cell. The use of a second bit allows indicating to the SES that the rate should be kept unchanged. This additional information limits the oscillation of the transmission rate.

Forth, a switch can calculate a desired rate which is currently can support, The desired rate is written to the ER field in passing FRM or BRM cells. However, a switch is only allowed to decrease the value of the ER field, it may never increase it.

The SES, upon reception of a BRM, updates its ACR as follows:

NI	**CI**	**Action**
0	0	ACR ← Max[MCR,Min[ER,PCR,ACR+RIF*PCR]]
0	1	ACR ← Max[MCR,Min[ER,ACR*(1-RDF)]]
1	0	ACR ← Max[MCR,Min[MCR,Min(ER,ACR)]]
1	1	ACR ← Max[MCR,Min[ER,ACR*(1-RDF)]]

where RIF stands for Rate Increase Factor (set default to 1/16) and RDF stands for Rate Decrease Factor (set default to 1/16).

Segmentation of the rate control loop is an important option provided by the rate-based framework. It is achieved by closing the control loop at intermediate switches, which then need to act as virtual source/virtual destinations. A virtual source typically must maintain separate queues for each connection, a virtual destination must maintain information for each connection. Experiments have shown the control loop segmentation increases the queues within the network, but is does not raise the delays significantly.

The rate-based approach was selected as a standard by the ATM Forum in 1994. The main reason for selecting the rate-based approach was that although credit-based congestion control permits development of low-cost network adapter cards and delivers high performance in LAN environments, it also requires excessively large buffers or exceedingly sophisticated buffering algorithms for WAN environments. Rate-based congestion control, on the other hand, provides a more efficient solution for WAN environments, making possible and affordable to develop high-capacity WAN switches at lower costs. Rate-based schemes are also very flexible and permit a variety of implementations. This allows for product differentiation, while maintaining compatibility with the standard.

14.1.2 Congestion detection

We describe three different schemes for congestion detection. The first scheme, which is commonly used, monitors the queue length. When the queue length exceeds an upper threshold Q_H congestion is detected, if it subsides a lower threshold Q_L congestion is regarded as terminated. The use of two thresholds instead of one reduces the speed of oscillations and therefore stabilizes the network state. The main advantage of the queue-based scheme is it low complexity, because the absolute queue length can be monitored by a single counter.

The second scheme monitors the virtual queue length Q^v_{ACR} defined by:

$$Q^v_{ACR} = Q\left(\frac{\sum ACR - LCR}{LCR} + 1\right) = Q\frac{\sum ACR}{LCR} \tag{14.1}$$

where Q denotes the instantaneous queue length, and LCR denotes the bandwidth available to the ABR class. As usual, we use two thresholds, Q_H and Q_L, to detect when the multiplexer enters and leaves the congestion state. With this scheme the upper threshold is exceeded more quickly if the aggregated arrival rate is high, even when the there are queued less than Q_H cells in the buffer. This behavior compensates for the fact that the number of cells queued additionally during feedback delay the larger the higher the arrival rate is. The same holds in an opposite manner for the lower threshold Q_L.

The third detection scheme considers the fact that it is more likely to approach a threshold when the queue length increases or decreases fast than if it varies slowly. Therefore, the growth of the queue length can be used to detect congestion. We define the virtual queue length as:

$$Q^v_{dQ} = Q\left(\frac{dQ/dt}{LCR} + 1\right) \tag{14.2}$$

Again, congestion is monitored according to the queue length thresholds Q_H and Q_L. The virtual measure Q^v_{dQ} compensates the influence of the growth, which affects the number of cells to be queued during feedback delay. A disadvantage of this scheme is the monitoring of the growth of the queue length. It requires a considerable computational effort of the frequency of updates is high.

14.1.3 Fairness criteria

An important objective in the allocation of bandwidth to ABR connections is *fairness*. Two main fairness criteria have been considered in the literature: max-min fairness and proportional fairness. The former criteria has been chosen by ATM forum and it is adopted by many ER allocation algorithms.

The max-min allocation is defined as follows [38]. Given a configuration with n contending sources, suppose the i^{th} source is allocated a bandwidth x_i. The allocation vector $\{x_1, x_2, ..., x_n\}$ is feasible if all link load levels are less than or equal to 100 %. Given an allocation vector, the source that is getting the least allocation is, in some sense, the “unhappiest source”. We need to find the feasible vectors that gives the maximum allocation to this unhappiest source. Now we remove this “unhappiest source” and reduce the problem to that of the remaining $n-1$ sources operating on a network with reduced link capacities. Again, we find the unhappiest source among these $n-1$ sources, give that source the max- minimum allocation and reduce the problem by one source. We repeat this process until all sources have been allocated the maximum they can get.

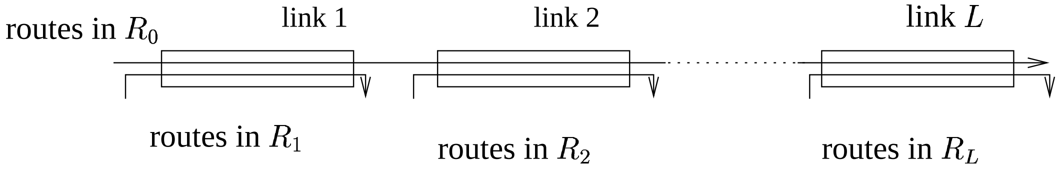


Figure 14.2: Linear network example

As an example, consider the network in Figure 14.2. Assume route set R_s contains x_s connections. The max-min allocation is as follows:

$$\lambda_r = \begin{cases} \frac{1}{x_0 + max_{s\geq 1} x_s} & \text{for } r \in R_0 \\ \frac{1}{x_s} 1 - \frac{x_0}{x_0 + max_{s\geq 1} x_s} & \text{for } r \in R_s, s \geq 1, x_s > 0 \end{cases} \tag{14.3}$$

The appropriateness of max-min fairness as a bandwidth sharing objective has recently been questioned by Kelly who has introduced the alternative notion of *proportional fairness* [42]. Rate allocations λ_r are proportionally fair if they maximize $\sum_R \text{log}\lambda_r$ under the capacity constraints. The objective may be interpreted as being to maximize the overall utility of rate allocations assuming each route has a logarithmic function.

Again, in the case of finitely many links and routes, the vector of proportionally fair rates λ_r is unique. It may be characterized as follows. The aggregate of proportional rate changes with respect to the optimum of any other feasible allocation λ'_r is negative, i.e.

$$\sum_R \frac{\lambda'_r - \lambda_r}{\lambda_r} \leq 0 \tag{14.4}$$

Consider how this rate allocation works in the case of the linear network in Figure 14.2. The network consists of L links each with unit capacity. The connections in set R_0 traverses all the L links. Connections in set R_i traverses only link i.First it is clear that all routes in the same set R_i must have the same allocation. Let γ_i be the allocation to routes in set R_i for $0 \leq i \leq L$. We necessarily have $x_0\gamma_0 + x_i\gamma_i = 1$ for $0 \leq i \leq L$: this sum is the capacity used at link i and must therefore be less or equal to one; however, for any rate allocation such that this sum is less than one, γ_i can be increased without violating the capacity constraints and this results in an increase in the objective function to be maximized. It follows that to determine the optimal rate allocation we must find the value γ_0 which maximizes

$$x_0\text{log}(\gamma_0) + \sum_{i=1}^{L} x_i\text{log}\left(\frac{1 - x_0\gamma_0}{x_i}\right) \tag{14.5}$$

Differentiating, we have at the optimum

$$\frac{x_0}{\gamma_0} = \sum_{i=1}^{L} \frac{x_i x_0}{1 - x_0\gamma_0} \tag{14.6}$$

giving

$$\gamma_0 = \frac{1}{x_0 + \sum_{x=1}^{L} x_i} \tag{14.7}$$

In the particular case where $x_i = 1$ for $0 \leq i \leq L$, we deduce the allocation $\lambda_0 = 1/(L+1)$ and $\lambda_r = L/(L+1)$ for $r \neq 0$. This corresponds to an overall throughput of $L - (L-1)/(L+1)$. It is clear from this example that proportional fairness penalizes long routes more severly than max-min fairness in the interest of greater overall throughput.

Researchers argue what kind of fairness is achieved by binary feedback schemes. As described in the previous section, binary feedback results in additive increase in the absence of congestion indication, and multiplicative decrease when congestion is indicated. Kelly originally argued for that additive increase/multiplicative decrease give rise to proportional fairness. However, recent results have shown that when sources receive feedback proportionally to their sending rates (as in the ABR control case), and the round trips times are equal, the fairness provided, called F_A fairness, is different. For example, in the linear network example, additive increase/multiplicative decrease would allocate higher rates to sources which received a small rate allocation from proportional fairness, but less to these sources than max-min fairness. In event of small link capacity, F_A fairness approximates proportional fairness. For large link capacity, F_A fairness varies between max-min and proportional fairness.

14.1.4 Explicit rate calculation: the MIT scheme

The MIT scheme consists of each source sending an RM cell every n-th data cell [38]. The RM cell contains the VC's current cell rate (CCR) and a "desired rate". The switches monitor all VC's rates and computes a "fair share". Any VC's whose desired rate is less than the fair share is granted the desired rate. If a VC's desired rate is more than the fair share, the desired rate field is reduced to the fair share and a "reduced bit" is set in the RM cell. The destinations return the RM cell back to the source, which then adjusts its rate to that indicated in the RM cell. If the reduced bit is clear, the source could demand a higher desired rate in the next RM cell. If the bit is set, the source use the current rate as the desired rate in the next RM cell.

The fair share is computed using an iterative procedure as follows. Initially, the fair share is set at the link bandwidth divided by the number of active VCs. All VCs, whose rates are less than the fair share are called "underloading VCs". If the number of underloading VCs increases at any iteration, the fair share is recomputed as follows:

$$FairShare = \frac{LinkBandwidth - \sum Bandwidth\ of\ Underloading\ VCs}{Number\ of\ VCs\ \text{-}\ Number\ of\ Underloading\ VCs} \tag{14.8}$$

The iteration is the repeated until the number of underloading VCs and the fair shares does not change. It can be shown that two iterations are sufficient for this procedure to converge. It can also be shown that the MIT scheme achieve max-min fairness in $4k$ round trips where k is the number of bottlenecks.

The MIT scheme has computational complexity in the order n of the number of VCs. This has motivated the search for other schemes with less complexity such as the ERICA scheme which has complexity O(1).

14.1.5 Explicit rate calculation: the ERICA scheme

The Explicit Rate Indication for Congestion Avoidance (ERICA) scheme works as follows [20]. The switch periodically monitors the load on each link and determines a load factor, z, the available capacity, and the number of currently active VCs. The load factor is computed as follows:

$$z \leftarrow \frac{ABR\ Input\ Rate}{ABR\ Capacity} \tag{14.9}$$

where

$$ABR\ Capacity \leftarrow Target\ Utilization \times Link\ Bandwidth - CBR\ Usage\ \text{-}\ VBR\ Usage \tag{14.10}$$

The input rate and output link ABR capacity are measured over an interval called switch measurement interval. The above steps are executed at the end of the switch measurement interval. Target utilization is a parameter which is set to a fraction (close to, but less than 100 %). The load factor, z, is an indicator of the congestion level of the link. The optimal operating point is at an overload value equal to one. The fair share of each VC is computed as follows:

$$FairShare \leftarrow \frac{ABR\ capacity}{Number\ of\ Active\ Connections} \tag{14.11}$$

The switch allows each connection sending a rate below the *FairShare* to rise to *FairShare*. If the connection does not use its *FairShare*, then the switch fairly allocated the remaining capacity to connections which can use it. For this purpose, the switch calculates the quantity:

$$\mathit{VCShare} \leftarrow \frac{\mathit{CCR}}{z} \tag{14.12}$$

If all VCs changed their rate to their *VCShare* values the, in the next cycle, the switch would experience unit overload ($z = 1$). *VCShare* aims at bringing the system to an efficient operating point, which may not necessarily be fair. A combination of the *VCShare* and *FairShare* quantities is used to rapidly reach optimal operation as follows:

$$\mathit{ER\ Calculated} \leftarrow \text{Max}\ (\mathit{FairShare, VCShare)} \tag{14.13}$$

This calculated ER value cannot be greater than the *ABR Capacity* which has been measured earlier. Hence, we have:

$$\mathit{ER\ Calculated} \leftarrow \text{Min}(\mathit{ER\ Calculated, ABR\ Capacity)} \tag{14.14}$$

To ensure the the bottleneck ER reaches the SES, each switch computes the minimum of the ER it has calculated as above and the ER value in the RM cell, and indicates this value in the ER field of the RM cell.

14.2 Congestion control in IP networks

14.2.1 TCP congestion control

Congestion control in TCP controls the sender window which determines how many segments can be send in a burst. The algorithm is due to Van Jacobsen [37]. The sender window during transmission phase is controlled according to

Sender-window=Min(receiver-window, congestion-window)

where *receiver-window* specifies how much free buffer space the receiver currently has. *Congestion-window* is set according to feedback on congestion in the network. From the beginning the source starts with *congestion-window* that equals one. After this, the *congestion-*

window is incremented by one for each positive acknowledgement (ACK) which returns to the source. This yields an exponential increase of the *congestion window* called "*slow start*". The exponential increase continues until there is a timeout for the oldest unacknowledged segment or the receiver runs out of memory space. Timeout indicates congestion in a router along the path to the destination. When a timeout occurs, the slow start phase is restarted with *congestion-window* that equals one. After this, the *congestion-window* is increased (exponentially) until a threshold, which is set to half the congestion-window you had at timeout. After the threshold is reached, the *congestion-window* is increased linearly, i.e with one after every acknowledgement burst of segments. This linear phase is called *congestion avoidance*. The linear increase continues until you have a timeout or until the *receiver-window* is reached. Figure 14.3 shows the evolution of the *congestion window* of source first starting in the slow start phase and then entering the congestion avoidance phase.

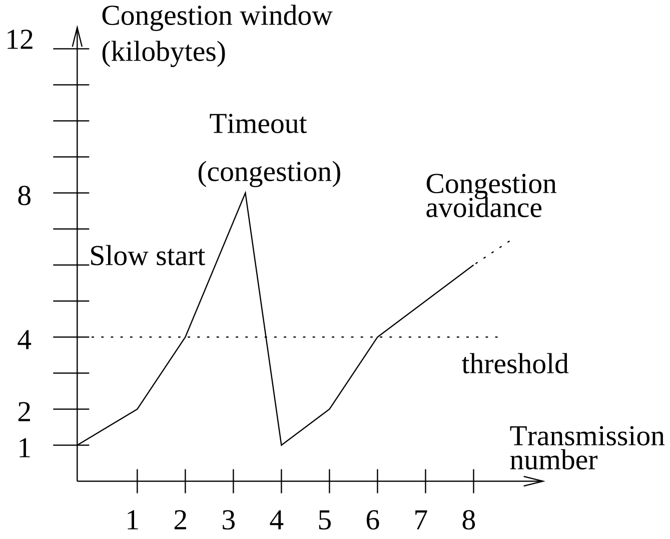


Figure 14.3: TCP congestion control

Fast Retransmit and Fast Recovery are two improvements of TCP congestion control that increases the throughput. Fast Retransmitt allows the source to retransmitt a segment when three duplicate ACKs (each requesting a segment with the same sequence number) are received instead of waiting until the retransmission timer expires. The Fast Recovery mechanism is also activated when three duplicate ACKs are received. Instead of reducing the congestion window to 1 and running slow start, the congestion window is set to half its current value and the congestion avoidance phase is entered. The throughput is improved since the source will not reduce its congestion window to one, as happens when slow start is initiated.

In TCP, the multiplicative decrease factor (1/2) is the same for all connections, but the additive increase factor is roughly one segment per round trip time (RTT), and this does not provide a uniform increase in the rates of TCP connections with different RTTs. As observed by many researchers, connections with long RTTs open their windows more slowly after reacting to congestion compared to those with short RTTs. If a mixture of short and long RTT connections share a bottleneck link, severe unfairness is likely as the short RTT connections grab the available bandwidth well before the long RTT connections have a chance. To remedy this problem, Sally Floyd has proposed that long RTT connections should use a faster rate of increase during their window growth phase.

Previous results have shown that for equal round-trip times TCP appeared to provide proportional fairness. Recent results, however, have shown that in the event of rate negative feedback and equal round trip times, TCP distributes rates more closely in accordance with the F_A fairness, i.e. the fairness measure realized by binary feedback schemes in ABR flow control.

14.2.2 Random Early detection

The objective of Random Early Detection (RED) is to improve the throughput of TCP-friendly connections. The RED mechanism drops packets according to the value of the smoothed queue length. The smoothed queue length, $\overline{Q}$, is obtained as an exponential weighted moving average :

$$\overline{Q} = (1-\alpha)\overline{Q} + \alpha Q \tag{14.15}$$

where Q denotes the instantaneous queue length. When the smoothed queue length is below T_{min} no dropping occur. Between T_{min} and T_{max} dropping occur at a certain probability according the RED dropping probability curve, see Figure 14.4. Above T_{max} all packets will be dropped. By dropping packets before congestion becomes severe, TCP will trigger Fast Recovery and enter the congestion avoidance phase. Since TCP avoids entering the slow start phase, which first closes the congestion window completely, the throughput is improved.

RED is claimed to provide several benefits, in particular 1) decrease the end-to-end delay and increase the throughput for both responsive (TCP) and non-responsive real-time traffic (UDP), 2) remove higher loss bias against bursty traffic observed with the Tail Drop mechanism, which drops incoming packets when the router runs out of buffer space.

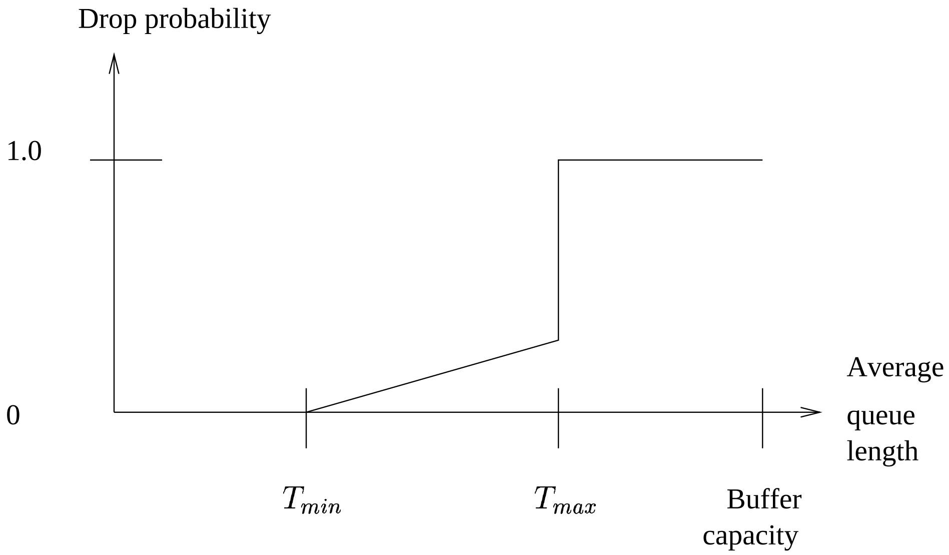


Figure 14.4: RED drop probability curve

14.2.3 Explicit Congestion Notification

Explicit Congestion Notification (ECN) is an extension proposed to RED which marks a packet instead of dropping it. ECN marks IP packets before congestion actually occurs, which is useful for protocols like TCP that are sensitive to even a single packet loss.

The ECN capable transport (ECT) bit is set in all IP packets sent from an ECN capable host so that the routers can see that ECN is supported. Packets encountering congestion are marked by the router using the Congestion Experienced (CE) bit in the IP packet when the average queue size is between T_{min} and T_{max}. The marking is controlled by a probability proportional to the average queue size following the procedure used in RED. Upon receipt of a congestion marked packet, the TCP receiver informs the sender (by setting the ECH-echo flag in TCP header of the subsequent ACKs) about incipient congestion which will in turn trigger the Fast Recovery and the congestion avoidance phase at the sender. The sender sets the Congestion Window Reduced (CWR) flag in the next TCP segment it sends to the receiver so that the receiver knows that actions have been taken.

Experiments with ECN has shown that for large TCP transfers aggressive ECN gives higher goodput (throughput experienced by the user) than RED.

Chapter 15

Call level performance evaluation

In this section we evaluate the performance of a call queuing system and a call loss system [59]. We assume both systems are offered a homogeneous Poisson call stream with rate $\lambda[\text{s}^{-1}]$. The call service (holding) time is assumed to be exponentially distributed with mean $1/\mu$ [s]. Each call is assumed to require 1 unit of capacity (bandwidth). In the queuing system we have one server with unit capacity. In the loss system we have C parallel servers, each with unit capacity.

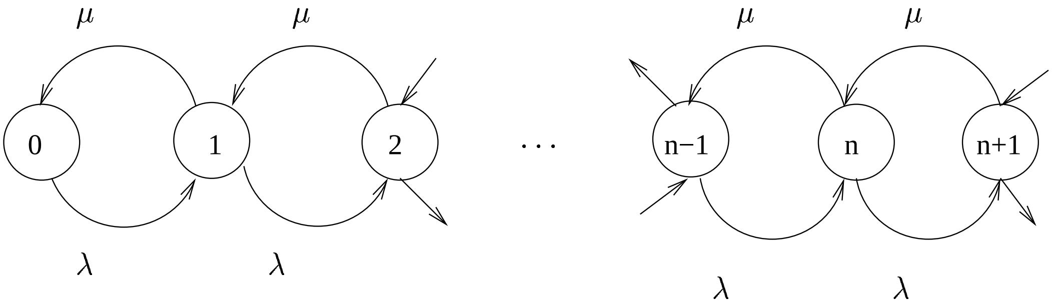


Figure 15.1: State transition diagram for infinite call queuing system.

15.1 M/M/1 queuing model

With the assumptions on exponential service time distribution and Poisson call arrivals we observe that

- we only need to look at the transition between time t and $t + \Delta t$ for $\Delta t \to 0$.

- if there are n calls in the queue at time $t+\Delta t$, then there could have be only $n-1, n$ or $n+1$ calls at time t

With these assumptions we can write:

$$\begin{aligned} p_n(t+\Delta t) &= p_n[(1-\mu t)(1-\lambda t)+\mu\Delta t\lambda\Delta t] \\ &+p_{n+1}[\mu\Delta t(1-\lambda\Delta t)]+p_{n-1}[\lambda\Delta t(1-\mu\Delta t)] \text{ if } n>0 \\ p_0(t+\Delta t) &= p_0(t)[(1-\lambda\Delta t)+\mu\Delta t\lambda\Delta t]+p_1(t)[\mu\Delta t(1-\lambda\Delta t)] \end{aligned} \tag{15.1}$$

According to Taylor series we have

$$p_n(t+\Delta t) = p_n(t)+\frac{\mathrm{d}}{\mathrm{dt}}p_n(t)\Delta t \tag{15.2}$$

Combining the two equations above and letting $\Delta t \to 0$ we have

$$\frac{\mathrm{d}}{\mathrm{dt}}p_n(t) = -(\lambda+\mu)p_n(t)+\lambda p_{n-1}(t)+\mu p_{n+1}(t) \tag{15.3}$$

$$\frac{\mathrm{d}}{\mathrm{dt}}p_0(t) = -\lambda p_0(t)+\mu p_1(t) \tag{15.4}$$

These are the differential-difference equations which describe the state of the system as function of time. We are interested in steady state behavior so we set $\frac{\mathrm{d}}{\mathrm{dt}}p_n(t) = 0$ for all n, which yields

$$0 = -(\lambda+\mu)\pi_n+\lambda\pi_{n-1}+\mu\pi_{n+1} \tag{15.5}$$

$$0 = -\lambda\pi_0+\mu\pi_1 \tag{15.6}$$

From these equations we have

$$\pi_1 = \frac{\lambda}{\mu}\pi_0 \tag{15.7}$$

$$\pi_2 = \frac{(\lambda+\mu)}{\mu}\pi_1-\frac{\lambda}{\mu}\pi_0 = \left(\frac{\lambda}{\mu}\right)^2\pi_0 \tag{15.8}$$

$$\dots \tag{15.9}$$

$$\pi_n = \frac{(\lambda+\mu)}{\mu}\pi_{n-1}-\frac{\lambda}{\mu}\pi_{n-2} = \left(\frac{\lambda}{\mu}\right)^n\pi_0 \tag{15.10}$$

All probabilities have to sum up to one:

$$\sum_{n=0}^{\infty} \pi_n = 1 \tag{15.11}$$

Therefore:

$$\pi_0 = \frac{1}{1 + \sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n} = \frac{1}{\frac{1}{1-\frac{\lambda}{\mu}}} = 1 - \frac{\lambda}{\mu} \tag{15.12}$$

Let $\rho = \frac{\lambda}{\mu}$. Inserting the expression for π_0 in the expression for π_n we get:

$$\pi_0 = 1 - \rho \tag{15.13}$$

$$\pi_n = \rho^n (1 - \rho) \tag{15.14}$$

The throughput is defined as the rate which calls depart from the queue. We have:

$$\text{Throughput} = \mu \text{Prob}[> 0 \text{ calls in the system}] = \tag{15.15}$$

$$\mu(1 - \text{Prob}[\ 0 \text{ calls in the system}]) = \tag{15.16}$$

$$\mu(1 - \pi_0) = \mu(1 - (1 - \rho)) = \lambda \tag{15.17}$$

The average number of calls in the system, $\overline{L}$, is given by

$$\overline{L} = \sum_{n=0}^{\infty} n\pi_n = (1 - \rho) \sum_{n=0}^{\infty} \rho^n = \frac{(1-\rho)\rho}{(1-\rho)^2} = \frac{\rho}{1-\rho} \tag{15.18}$$

The average time in the system, $\overline{W}$, is obtained from Little's formula:

$$\overline{W} = \overline{L}/\lambda = \frac{\rho}{1-\rho} \frac{1}{\lambda} \tag{15.19}$$

The average number of calls in the queue, $\overline{L}_q$, is given by:

$$\overline{L}_q = L - (1 \cdot \text{Prob[Server not empty]}) = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho} \tag{15.20}$$

The average time in the queue, $\overline{W}_q$, is given by Little's formula:

$$\overline{W}_w = \overline{L}_q/\lambda = \frac{\rho^2}{1-\rho} \frac{1}{\lambda} \tag{15.21}$$

15.2 M/M/C/* Loss model

In this section we evaluate the call blocking probability of system with C parallel servers but with no queue (loss system). The state of the system is denoted by n and represents the number of busy servers.

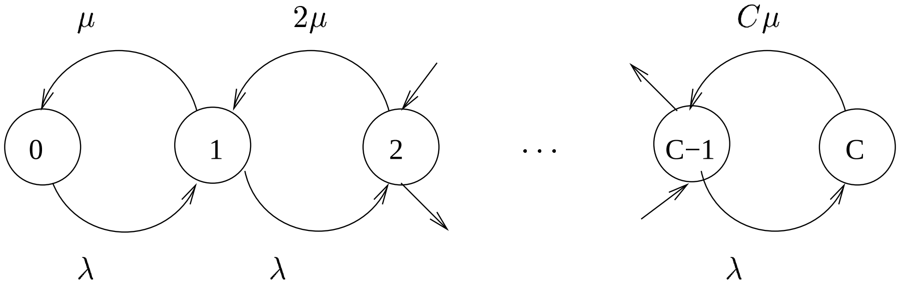


Figure 15.2: State transition diagram for call loss system.

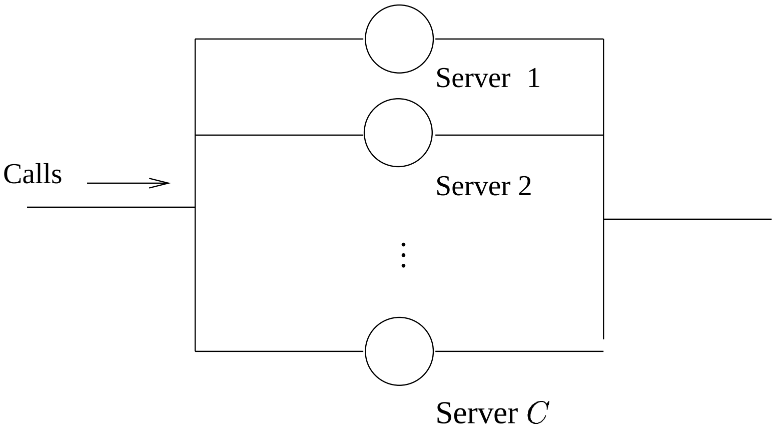


Figure 15.3: Call loss system.consists of C parallel servers

By the same way as in the previous section we obtain the differential equations:

$$\frac{\mathrm{d}}{\mathrm{dt}} p_n(t) = -(\lambda + n\mu)p_n(t) + \lambda p_{n-1}(t) + (n+1)\mu p_{n+1}(t) \tag{15.22}$$

$$\frac{\mathrm{d}}{\mathrm{dt}} p_0(t) = -\lambda p_0(t) + \mu p_1(t) \tag{15.23}$$

We are interested in steady state behavior so we set $\frac{\mathrm{d}}{\mathrm{dt}} p_n(t) = 0$ for all n, which yields

$$0 = -(\lambda + n\mu)\pi_n + \lambda\pi_{n-1} + (n+1)\mu\pi_{n+1} \tag{15.24}$$

$$0 = -\lambda\pi_0 + \mu\pi_1 \tag{15.25}$$

Hence, the probabilities π_n are

$$\pi_1 = \frac{\lambda}{\mu}\pi_0 \tag{15.26}$$

$$\pi_2 = \frac{(\lambda + \mu)\frac{\lambda}{\mu}\pi_0 - \lambda\pi_0}{2\mu^2} = \frac{\lambda^2}{2\mu}\pi_0 \tag{15.27}$$

$$\pi_3 = \frac{(\lambda + 2\mu)\frac{\lambda^2}{2\mu^2}\pi_0 - \lambda\frac{\lambda}{\mu}\pi_0}{3\mu} = \frac{\lambda^3}{6\mu^3}\pi_0 \tag{15.28}$$

$$\pi_4 = \frac{(\lambda + 3\mu)\frac{\lambda^3}{6\mu^3}\pi_0 - \lambda\frac{\lambda^2}{\mu^2}\pi_0}{4\mu} = \frac{\lambda^4}{4!\mu^4}\pi_0 \tag{15.29}$$

$$\ldots \tag{15.30}$$

$$\pi_n = \left[\prod_{i=1}^{n} \frac{\lambda}{i\mu}\right]\pi_0 = \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n \pi_0 \tag{15.31}$$

Normalization of the probabilities, $\sum_{n=0}^{C} \pi_n = 1$, yields

$$\sum_{n=0}^{C} \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n \pi_0 = 1 \tag{15.32}$$

resulting in the empty system probability π_0

$$\pi_0 = \frac{1}{\sum_{n=0}^{C} \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n} \tag{15.33}$$

A call which find a system with C busy servers must be rejected. Hence, the call blocking probability, $E(C, \rho)$, where $\rho = \lambda/\mu$ is

$$E(C, \rho) = \frac{\frac{1}{C!}\rho^C}{\sum_{n=0}^{C} \frac{1}{n!}\rho^n} \tag{15.34}$$

which also is known as the Erlang blocking formula.

Chapter 16

Packet level performance evaluation

16.1 General considerations

A packet network is a complex network of queues. The behavior of a packet stream, e.g. the packet inter-arrival times, will typically change as the packets cross the network. The reason is that at each switch/router the packets share the output multiplexer with packets from other calls. The mixing of traffic streams can result in clumping and dispersion of packets since packets may have to wait in the queue for their turn to be transmitted. A multiplexer inside the network will be offered traffic streams that have passed zero or more multiplexing steps before arriving to this multiplexer. However, to consider the "sharing effects" experienced by each call is believed to be too complex. Instead, each switch/router in the network is analyzed as if it is offered fresh user traffic which only has passed the policing function. The packet traffic on successive links are assumed to be independent. When a new call is accepted to the network only the traffic on the path of the new call is assumed to be changed.

Switching conflicts can give rise to *packet scale congestion* and *burst scale congestion*. These types of congestion are due to overload on different time scales. Packet scale congestion are due to the random fluctuations on a short time scale due to variable packet inter-arrival times. The output link can only transmit one packet at a time, and if many packets simultaneously want access to the same output link, all but one packet must wait in a queue. Burst scale congestion occurs at a larger time scale when the total input rate to the multiplexer exceeds the output capacity for some time period. In this case, the buffer occupancy level will increase until the buffer is saturated or the total input rate decreases. Switching conflicts that arise when the buffer is saturated can not be resolved, forcing arriving packets to be dropped.

The probability of packet loss due to packet scale congestion can be reduced by using a small central or output buffer which can hold in the order of 100 packets per output link. The buffer requirements are proportional to the number of input links. To obtain a low probability of packet loss due to burst scale congestion much larger buffers may be required. The buffer requirement is in this case counted in hundreds of packets. See Figure 16.1 which shows the general characteristics of a buffer overflow distribution.

The fluid flow queuing model captures the behavior of burst scale congestion. Hence, performance measures derived from the fluid flow queuing model will be accurate in the case of large buffers. For smaller buffer sizes, the queue analysis should be performed assuming Markov modulated Poisson process (MMPP) packet arrivals. The MMPP queuing model captures the behavior of both packet and burst scale congestion.

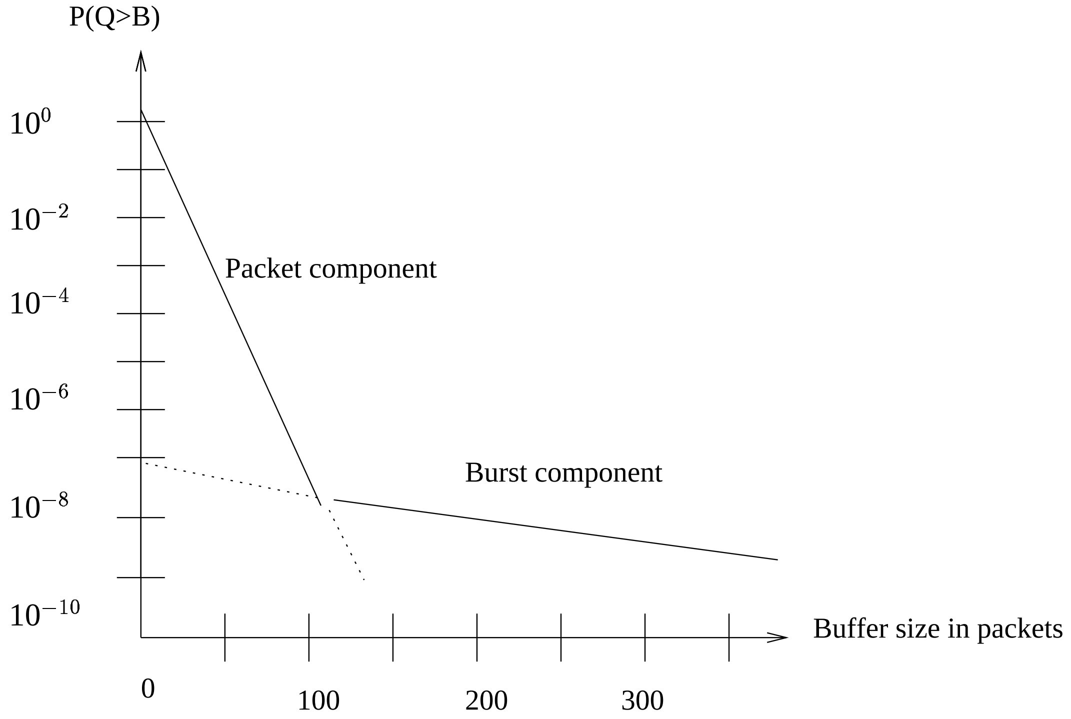


Figure 16.1: Buffer overflow distribution

In general, performance evaluation faces an *accuracy-simplicity dilemma*. High accuracy is desired to to admit correct and efficient call admission control. High accuracy enables call admission control which is neither too optimistic nor too pessimistic. Low complexity (high simplicity) is required since call admission control must operate in real time and provide admission decisions in in the order of tens of milliseconds. Hence, the designer of call admission control must make a tradeoff between accuracy and simplicity.

16.2 Queue management

16.2.1 Buffering strategies

Consider an ATM switch or a IP router with multiple input and output links. Without any special mechanisms, switching conflicts would arise when packets arriving at different input ports simultaneously want access to the same output link. To resolve the switching conflicts buffers are used to queue packets waiting for transmission. The buffers can be placed at three places in a switch/router: at the inputs, at the outputs or centrally [57].

Input queuing is least efficient; it suffers from Head Of Line (HOL) blocking. Lets assume input port i and j has a packet destined for output link p. Further, assume that in queue j there is a second packet destined for output link q for which there is currently no packet waiting in the other input queues. HOL blocking manifests itself by the fact that while the packet first in queue j waits for transmission, it also forces the packets behind it to wait. That is, the packet destined for output link q must wait although there would be no switching conflicts for that packet.

The central queuing solution behaves exactly as the output queuing solution. This means that the mean waiting time is exactly that of the output queuing system. However, since multiples queues are combined in a single physical memory, the major advantage of the central queuing system is reflected in the number of packets to be stored in the central memory (central queue). A more complicated control logic is required to ensure that the single central memory performs the First-In-First-Out (FIFO) discipline to all outlets. The memory size can be computed as the convolution of N individual output queues.

Indeed, since the buffer memory will be shared, a more effective use of the memory can be made. It can be shown that with the size of the central memory and the sum of the output buffer memory sizes being equal, the central queuing yields lower packet loss probability than output queuing.

16.2.2 First-in-First-Out queuing

In the First-In-First-Out (FIFO) queuing discipline packets are served according to the order at which they arrive. That is, the server operates on a First-Come-First-Served (FCFS) basis. FIFO/FCFS provides statistical sharing among calls. Each call receives a statistical (long-term) fraction of the server capacity. On the short term, no guarantees on service are given, since the likelihood of obtaining service depends on the momentary behavior of other sources using the same multiplexer. In particular, if too many sources send at their peak rates the buffer will overflow and packets will have to be dropped. The FCFS discipline is work conserving, which means that the server is never idle when it has work to do.

The FCFS discipline may also change the time spacing between successive packets leaving the switch/router. The obvious reason is that packets may have to wait in the queue before being served. Packets may suffer from clumping or dispersion. In clumping the time distance between two successive packet is reduced, and in dispersion the time distance is increased. The packet clumping/dispersion introduces delay variability which might be critical for real-time services.

To smooth out delay variability, the receiver can use a playout buffer. Packets which arrive at the receiver is placed in buffer which is served in a periodic manner. Packets have time stamps which indicate the time when they were transmitted from the source. The receiver reads out a packet from the buffer a fixed time after the packet was sent from the source. Note that the smoothing ability, i.e. the size of the buffer, is limited by the maximum tolerated delay.

16.2.3 Priority queuing

In (time) Priority Queuing (PQ) separate FIFO queues are used for each call class,which may contain one or more calls. The queues have different priorities which control the order in which queues are served. A low priority queue is only served if there are no packets in the higher priority queues. Starvation of low priority packets is therefore possible if the packet arrival rates of higher priority call classes are too high. However, by proper CAC all calls should be guaranteed a certain statistical, long-term fraction of the server capacity. The PQ discipline is work conserving. Like FIFO, packet streams may suffer from delay variability.

16.2.4 Weighted fair queuing

In Weighted Fair Queuing (WFQ) each call class has its own FIFO queue [53]. Each queue is guaranteed to obtain a certain fraction of the bandwidth (given by the weight of queue) on both the short-term and long-term time scale. When no packets arrive within a call class the excess capacity is shared fairly, according to the weights, among active call classes. The WFQ discipline schedules packets according to the time the packets would have been sent by bit-by-bit round robin. Hence, if all queues contain only one packet each, and all weights are equal, the packets are transmitted in increasing packet size order. When a packet is transmitted the packet behind it becomes first in the queue. This new packet joins the bit-by-bit service emulation and if the packet is short enough it may be transmitted before other longer packets which already were first in their queue when the short packet reached the head of its queue. The WFQ discipline is work conserving. Due to the “isolation” between call classes the packet delay variability under WFQ becomes lower than under FIFO or PQ.

16.3 Buffer management

In the simplest form of buffer management all classes completely share the buffer capacity. On the other hand, space priority queuing assigns buffer space to packets according to their priority. Two examples of such schemes are the Push out scheme and the Partial buffer sharing scheme [52].

16.3.1 Push out scheme

The push out scheme operates only when the buffer is full. When a low priority packet arrives at a full buffer, it is dropped. When a high priority packet arrives at a full buffer and a lower priority packet is in the buffer, the lower priority packet is “pushed out”, and the arriving high priority packet is added to the buffer. Simulation experiments show that when two priority classes contribute equally to the load, the packet loss for the higher priority class is dramatically reduced while the packet loss for the lower priority class increases slightly.

16.3.2 Partial buffer sharing scheme

The partial buffer sharing scheme reserves a fixed number of buffer spaces for high priority packets. Packets from all classes share the buffer space until the buffer content reaches a threshold. If the number of packets in the buffer exceeds the threshold then only higher priority packets will be accepted (as long as there is space). Simulation results indicate that partial buffer sharing performs only slightly worse than the push out approach.

16.4 Performance evaluation of a single multiplexer

In this section we perform queuing analysis of a single multiplexer which performs queuing centrally or at the output port. We will study only the FIFO queuing discipline with complete buffer sharing. For sake of simplicity, we omit the analysis of the PQ and WFQ disciplines. However, we note that worst-case end-to-end delay bounds under the WFQ discipline are tighter than worst case end-to-end delay bounds under the FIFO or PQ discipline. Worst case delay bounds are required by hard-real time applications. Soft real-time applications can rely on statistical delay bounds. Indeed, statistical end-to-end delay bounds for the WFQ displine have been proposed in the literature.

In this section we analyze the performance for three multiplex scenarios:

- a single multiplexer with a FIFO queue offered traffic from a set of heterogeneous Markov modulated rate (fluid) sources
- a single bufferless multiplexer offered traffic from a set of heterogeneous Markov modulated rate (fluid) sources
- a single multiplexer with a FIFO queue offered Fractional Brownian Motion (FBM) traffic

We omit the case of MMPP sources due to the complexity of the queuing analysis. The bufferless multiplexer case is included since it allows computationally efficient estimation of the packet loss ratio.

16.4.1 Fluid flow queuing model

Assumptions and notation

We assume the FIFO multiplexer is offered traffic from c classes of ON/OFF sources. An ON/OFF source alternates between an active ON state, when it transmits information at a peak rate, and a silent OFF state. The durations of the ON and OFF periods are assumed to be exponentially distributed.

A traffic class i is described by four parameters:

- number of sources: N_i
- average duration of the ON state: $1/b_i$ [s]
- average duration of the OFF state: $1/a_i$ [s]
- peak rate in the ON state: f_i [Mbps]

The intensity (rate) of transitions from the ON (OFF) state to the OFF (ON) state for a class i source is $b_i(a_i)[\mathrm{s}^{-1}]$. The mean rate for class i is given by $m_i = f_i\frac{a_i}{a_i+b_i}$ [Mbps].

The traffic descriptor (peak rate, mean rate, maximum burst size) can be enforced by two leaky buckets or two token buckets, one supervising the peak rate and one supervising the mean rate and the maximum burst size. It is widely believed, though the author have seen no formal mathematical proof, that the worst case output from a leaky bucket or token bucket with respect to queue build up is a periodic ON/OFF process in which the source is transmitting at peak as long as allowed and then turns silent until it is able to transmit a maximum burst at peak again. From numerical experience it is known that more variable ON and OFF period durations lead to larger queuing build up. Therefore the cell loss ratio in the exponential distributed ON/OFF queue will be an upper bound on the cell loss probability in the periodic ON/OFF queue. Hence, the ON → OFF intensity b_i should be set to $b_i = f_i/\mathit{MBS}$. The OFF → ON intensity a_i should be set to $a_i = \frac{b_i m_i}{f_i - m_i}$.

The data transmitted by the $\sum N_i$ sources is received by a finite buffer of size B Mbits with a maximum output rate of C Mbps. The buffer is modelled as a fluid reservoir with a hole in the bottom and arriving information is modelled as a fluid running into the reservoir. If f_{in} is the input rate, and x is the buffer content, then the change in buffer content is given by [36]

$$dx/dt = \begin{cases} 0 & \text{when } x = 0 \text{ and } f_{in} < C \\ f_{in} - C & \text{when } \begin{cases} x = 0 \text{ and } f_{in} > C \\ 0 < x < B \\ x = B \text{ and } f_{in} < C \end{cases} \\ 0 & \text{when } x = B \text{ and } f_{in} > C \end{cases} \tag{16.1}$$

The average input rate is $\sum_{i=1}^{c} N_i f_i a_i/(a_i + b_i)$ and the load on the output is then given by

$$\rho = \frac{\sum_{i=1}^{c} N_i f_i \frac{a_i}{a_i+b_i}}{C} \tag{16.2}$$

We assume the average input rate to be smaller than the output capacity, i.e. $\rho < 1$.

Let k_i denote the number of active sources in class i, and $\mathbf{k} = (k_1, ..., k_c)$ the state vector which the sources are in. Let

$$S := \{\mathbf{k} = (k_1, ..., k_c) : 0 \le k_i \le N_i, i = 1, ..., c\} \tag{16.3}$$

denote the state space for the sources. S is of cardinality $(N_1 + 1) \cdots (N_c + 1)$.

Vectors and matrices will appear in bold letter such that they can be distinguished from numbers, functions etc.

Let $\mathbf{f} = (f_1, ..., f_c)$ denotes the peak rate vector for the sources and by the scalar product of $\mathbf{k}$ and $\mathbf{f}$ we mean:

$$f_{in} = \mathbf{k} \cdot \mathbf{f} = \sum_{i=1}^{c} k_i f_i \tag{16.4}$$

which is the total input rate in state $\mathbf{k}$.

The assumption that the duration of the ON (OFF) state is exponential ensures that the transitions between state in S are determined by a Markov chain, and this together with the fluid assumption enables us to find the equilibrium buffer distribution as solution to a set of first order differential equations.

A set of differential equations

For $\mathbf{k}$ in S, $t > 0, 0 < x < B$, let $P_{\mathbf{k}}(t, x)$ denote the probability that at time t, the sources are in state $\mathbf{k}$ and the buffer content does not exceed x. Due to the exponential assumption, at the next small time interval $(t, t + \Delta t)$ a source from class i will switch from the OFF state to the ON state with probability $(N_j - k_j)a_j\Delta t + o(\Delta t)$, and a source from class i will switch from the ON state to the OFF state with probability $k_j b_j \Delta t + o(\Delta t)$. The probability of two or more changes is $o(\Delta t)$. These considerations yields the following expression:

$$\begin{aligned} P_{\mathbf{k}}(t + \Delta t, x) = & \sum_{i=1}^{c}(N_i - k_i + 1)a_i\Delta t P_{(k_1,\ldots,k_i-1,\ldots,k_c)}(t, x - \Delta x) \\ & \sum_{i=1}^{c}(k_i + 1)b_i\Delta t P_{(k_1,\ldots,k_i+1,\ldots,k_c)}(t, x - \Delta x) \\ & (1 - \sum_{i=1}^{c}(N_i - k_i)a_i\Delta t + k_i b_i \Delta t)P_{\mathbf{k}}(t, x - \Delta x) \end{aligned} \tag{16.5}$$

where $\Delta x = (\sum_{i=1}^{c} f_i k_i - C)\Delta t$ and where a function $g(t)$ is written $o(t)$ whenever $g(t)/t \to 0$ for $t \to 0$.

Isolating terms containing $P_{\mathbf{k}}$ on the left side, dividing by Δt and taking the limit $\Delta t \to 0$, yields

$$\begin{aligned} \frac{\partial P_{\mathbf{k}}}{\partial t}(t, x) + (\sum_{i=1}^{c} f_i k_i - C)\frac{\partial P_{\mathbf{k}}}{\partial x}(t, x) = & -\sum_{i=1}^{c}((N_i - k_i)a_i + k_i b_i)P_{\mathbf{k}}(t, x) \\ & + \sum_{i=1}^{c}(N_i - k_i + 1)a_i P_{(k_1,\ldots,k_i-1,\ldots,k_c)}(t, x) \\ & + \sum_{i=1}^{c}(k_i + 1)b_i P_{(k_1,\ldots,k_i+1,\ldots,k_c)}(t, x) \end{aligned} \tag{16.6}$$

We are only interested in the time independent equilibrium probabilities

$F_{\mathbf{k}}(x)$:= equilibrium probability that the sources are in state **k** and the buffer content does not exceed x

and we therefore put $\partial P_{\mathbf{k}}/\partial t = 0$ and obtain

$$\begin{aligned} (\sum_{i=1}^{c} f_i k_i - C)\frac{\partial F_{\mathbf{k}}}{\partial x}(x) = & -\sum_{i=1}^{c}(N_i - k_i)a_i + k_i b_i)F_{\mathbf{k}}(x) \\ & + \sum_{i=1}^{c}(N_i - k_i + 1)a_i F_{(k_1,\ldots,k_i-1,\ldots,k_c)}(x) \\ & + \sum_{i=1}^{c}(k_i + 1)b_i F_{(k_1,\ldots,k_i+1,\ldots,k_c)}(x) \end{aligned} \tag{16.7}$$

where $F_{\mathbf{k}} := 0$ when k_i is not in $\{0, 1, ..., N_i\}$.

Equation (16.7) can be written as a $(N_1 + 1) \cdots (N_c + 1)$ dimensional matrix differential equation

$$\mathbf{D}\frac{d\mathbf{F}}{dx} = \mathbf{MF}(x) \tag{16.8}$$

where $\mathbf{D}$ is a diagonal matrix with entry $(\mathbf{k}, \mathbf{k})$ equal to

$$d_{\mathbf{k}} = (\sum_{i=1}^{c} f_i k_i - C) \tag{16.9}$$

and where entry $(\mathbf{k}, \mathbf{n})$ in $\mathbf{M}$ looks as follows:

$$\begin{aligned} m_{(\mathbf{k},\mathbf{k})} &= -\sum_{i=1}^{c}(N_i - k_i)a_i + k_i b_i, \quad \text{for } \mathbf{k} \in S \\ m_{(\mathbf{k},k_1,...,k_i-1,...,k_c)} &= (N_i - k_i + 1)a_i, \quad \text{for } \mathbf{k} \in S \\ m_{(\mathbf{k},k_1,...,k_i+1,...,k_c)} &= (k_i + 1)b_i, \quad \text{for } \mathbf{k} \in S \end{aligned} \tag{16.10}$$

In both approaches to be presented it is necessary to be able to invert the matrix $\mathbf{D}$. To ensure this, the following technical assumption is needed:

$$C \in \left\{ \sum_{i=1}^{c} f_i k_i | k_i \text{ is an integer} \right\} \tag{16.11}$$

Equation (16.11) is a mild assumption and in the case where all the f_i's are multiples of f_1 the condition (16.11) is fulfilled whenever C is not a multiple integer of f_1.

The (vector) solution to the matrix differential equation is:

$$\mathbf{F}(x) := \text{exp}(\mathbf{D}^{-1}\mathbf{M}x)\mathbf{f}_0 \tag{16.12}$$

where $\mathbf{F}(x) = \{F_{\mathbf{k}}(x)\}_{\mathbf{k}\in S}$ and where the initial vector $\mathbf{f}_0$ is found from initial conditions.

The initial condition

Let $u_{\mathbf{k}}$ denote the probability of the buffer being held at its maximum B, and the sources being in state $\mathbf{k}$. If $q_{\mathbf{k}}$ denotes the overall probability of the sources being in state $\mathbf{k}$, then the following expression is valid for $u_{\mathbf{k}}$

$$u_{\mathbf{k}} = q_{\mathbf{k}} - \text{lim}_{x\to B} F_{\mathbf{k}}(x) \tag{16.13}$$

where $q_{\mathbf{k}}$is to be calculated as:

$$q_{\mathbf{k}} = \prod_{i=1}^{c} \frac{\binom{N_i}{k_i}\left(\frac{a_i}{b_i}\right)^{k_i}}{(1+\frac{a_i}{b_i})^{N_i}} \tag{16.14}$$

Using this notation the initial condition is easy to formulate. When the input rate is larger than the output rate the buffer cannot stay empty. Therefore:

$$F_{\mathbf{k}}(0) = 0 \text{ when } \sum_{i=1}^{c} k_i f_i > C \tag{16.15}$$

When the input rate is smaller than the output rate the buffer cannot stay at its maximum. Therefore:

$$0 = u_{\mathbf{k}} = q_{\mathbf{k}} - \text{lim}_{x\to B} F_{\mathbf{k}}(x) \text{ when } \sum_{i=1}^{c} k_i f_i < C \tag{16.16}$$

The solution

At is was seen in the previous section, the equilibrium buffer distribution can be derived when the exponential of the matrix $\mathbf{D}^{-1}\mathbf{M}$ is found.

Suppose that the matrix $\mathbf{D}^{-1}\mathbf{M}$ has basis vectors $(\phi_{\mathbf{k}})_{\mathbf{k}\in S}$, and let $z(\mathbf{k})$ be the eigenvalue associated with $\phi_{\mathbf{k}}$. Since the dimension of $\mathbf{D}^{-1}\mathbf{M}$ is equal to $(N_1+1)\cdots(N_c+1)$, it is possible to use S as index set. With this notation the solution will have the appearance:

$$\mathbf{F}(x) = \sum_{\mathbf{k}\in S} a_{\mathbf{k}} \text{exp}(z(\mathbf{k})x)\phi_{\mathbf{k}} \tag{16.17}$$

where the coefficients $a_{\mathbf{k}}$ must be found by means of the initial condition which in the eigenvalue context reads:

$$\begin{aligned} 0 = F_{\mathbf{n}}(0) &= \sum_{i=1}^{c} a_{\mathbf{k}}(\phi_{\mathbf{k}})_{\mathbf{n}} \quad \text{when } \mathbf{n}\cdot\mathbf{f} > C \\ q_{\mathbf{n}} = \text{lim}_{x\to B} F_{\mathbf{n}}(x) &= \sum_{i=1}^{c} a_{\mathbf{k}} \text{exp}(z(\mathbf{k})x)(\phi_{\mathbf{k}})_{\mathbf{n}} \quad \text{when } \mathbf{n}\cdot\mathbf{f} < C \end{aligned} \tag{16.18}$$

where $(\phi_{\mathbf{k}})_{\mathbf{n}}$ denotes the $\mathbf{n}$-th component of the $\mathbf{k}$-th eigenvector.

In the infinite buffer case with identical sources, the initial condition is different, and it can be formulated such that the corresponding coefficient matrix is a Van der Monde matrix, and an analytical solution exist. However, in the finite buffer case with a more general input stream, no such approach seems possible, and the equation must be solved numerically.

The eigenvalue for state $\mathbf{k}$ can be found by solving the algebraic equation $f(z(\mathbf{k})) = g(z(\mathbf{k}))$ where

$$f(z(\mathbf{k})) = z(\mathbf{k})(C - \sum_{i=1}^{c} \frac{N_i}{2} f_i) - \sum_{i=1}^{c} \frac{N_i}{2}(a_i + b_i) \tag{16.19}$$

$$g(z(\mathbf{k})) = \sum_{i=1}^{c}(k_i - \frac{N_i}{2})\sqrt{(z(\mathbf{k})f_i + b_i - a_i)^2 + 4a_i b_i} \tag{16.20}$$

The eigenvector $\phi_{\mathbf{k}}$ corresponding to the eigenvalue $z(\mathbf{k})$ is given as the coefficients in the following polynomial in c variables:

$$p_{\mathbf{k}} = \prod_{i=1}^{c}(x_i - r_i(z))_i^k (x_i - s_i(z))^{N_i - k_i} \tag{16.21}$$

where

$$r_i(z) = \frac{-(zf_i + b_i - a_i) + \sqrt{(zf_i + b_i - a_i)^2 + 4a_i b_i}}{2a_i} \tag{16.22}$$

$$s_i(z) = \frac{-(zf_i + b_i - a_i) - \sqrt{(zf_i + b_i - a_i)^2 + 4a_i b_i}}{2a_i} \tag{16.23}$$

The size of the state space for the fluid flow model is $N_s = \prod_{i=1}^{c}(N_i + 1)$. As far as computational complexity is concerned, is is dominated by the calculation of the coefficients $a_{\mathbf{k}}$ from the boundary conditions. The coefficients are found by solving a set of N_s linear equations. This system of equations can be solved by Gauss elimination or, which can be more efficient, by some iterative method. In any case, the associated complexity is in the order of $O(N_s^3)$. Hence, the complexity of the fluid flow model increases very fast with increasing number of classes c and increasing class sizes N_i. Therefore, the fluid flow model is only considered to be useful as a reference model, and not as a basis for implementation of call admission control in real multi-service networks.

Performance formulas

In this subsection we present formulas for the buffer overflow probability, packet loss ratio, and mean waiting time in the queue.

The equilibrium distribution can be written as

$$\mathbf{F}(x) = \sum_{\{\mathbf{k}|\mathbf{k}\cdot\mathbf{f}>C\}} a_{\mathbf{k}} \text{exp}(z(\mathbf{k})x)\phi_{\mathbf{k}} + \mathbf{q} \tag{16.24}$$

where $\mathbf{q}$ is the equilibrium probability distribution.

The buffer overflow probability $G(x) = \text{P}(Q > x)$ can be written as

$$\begin{aligned} G(x) = 1 - \sum_{\mathbf{k}\in S} F_{\mathbf{k}}(x) = 1 - \sum_{\mathbf{k}\in S} q_{\mathbf{k}} - \sum_{\mathbf{k}\in S} \sum_{\{\mathbf{n}|\mathbf{n}\cdot\mathbf{f}>C\}} a_{\mathbf{n}} \text{exp}(z(\mathbf{n})x)(\phi_{\mathbf{n}})_{\mathbf{k}} \\ = -\sum_{\mathbf{k}\in S} \sum_{\{\mathbf{n}|\mathbf{n}\cdot\mathbf{f}>C\}} a_{\mathbf{n}} \text{exp}(z(\mathbf{n})x)(\phi_{\mathbf{n}})_{\mathbf{k}} \end{aligned} \tag{16.25}$$

For large buffers the following asymptotic approximation is valid:

$$G(x) \sim -a_{\mathbf{0}} \text{exp}(z(\mathbf{0})x) \sum_{\mathbf{k}\in S} (\phi_{\mathbf{0}})_{\mathbf{k}} \tag{16.26}$$

where $z(\mathbf{0})$ is the largest negative eigenvalue and $\phi_{\mathbf{0}}$ its associated eigenvector.

The overall loss ratio p_{loss} is defined as the fraction lost information to offered information. Loss can only take place when the buffer is at maximum and the input rate is larger than the output rate. Therefore we get:

$$p_{loss} = \frac{\sum_{\{\mathbf{k}|\mathbf{k}\cdot\mathbf{f}>C\}} (\mathbf{k}\cdot\mathbf{f} - C) u_{\mathbf{k}}}{\sum_{i=1}^{c} N_i f_i \frac{a_i}{a_i+b_i}} \tag{16.27}$$

The buffer overflow probability is an upper bound of the packet loss ratio.

The distribution of queuing delay, $H(t) = \text{P}(T \le t)$, is given by:

$$H(t) = \sum_{\mathbf{k}\in S} F_{\mathbf{k}}(Ct) \sum_{i=1}^{c} \frac{k_i}{\rho_i N_i} \frac{N_i}{N} = \sum_{\mathbf{k}\in S} F_{\mathbf{k}}(Ct) \sum_{i=1}^{c} \frac{k_i}{\rho_i N} \tag{16.28}$$

where $N = \sum_{i=1}^{c} N_i$ denotes the total number of sources, and $\rho_i = \frac{a_i}{a_i+b_i}$ denotes the probability that an class i source is in the active state.

This expression (16.28) can be used to determine the the $1-\alpha$ quantile, $D_{queueing}(\alpha)$, of the queuing delay distribution. The delay quantile can be found by searching for the value of t which yields $H(t) = 1 - \alpha$.

The mean waiting time in the queue, μ, is obtained as

$$\mu = \int_0^{B/C} t \frac{\mathrm{d}H(t)}{\mathrm{d}t} \mathrm{d}t \tag{16.29}$$

The variance of the waiting time in the queue, σ^2, is obtained as

$$\sigma^2 = \int_0^{B/C} (t-\mu)^2 \frac{\mathrm{d}H(t)}{\mathrm{d}t} \mathrm{d}t \tag{16.30}$$

16.4.2 Bufferless fluid flow model

Exact solution

The exact packet loss ratio for a superposition of heterogeneous ON/OFF Markov fluid sources offered to a bufferless multiplexer is:

$$p_{loss} = \frac{\sum_{\{\mathbf{k}|\mathbf{k}\cdot\mathbf{f}>C\}} (\mathbf{k}\cdot\mathbf{f} - C) q_{\mathbf{k}}}{\sum_{i=1}^{c} N_i f_i \frac{a_i}{a_i+b_i}} \tag{16.31}$$

where $q_{\mathbf{k}}$ is the equilibrium probability of being in state :

$$q_{\mathbf{k}} = \prod_{i=1}^{c} \frac{\binom{N_i}{k_i}\left(\frac{a_i}{b_i}\right)^{k_i}}{\left(1+\frac{a_i}{b_i}\right)^{N_i}} \tag{16.32}$$

The computational complexity of the exact method may be estimated by [48]

$$O\left[(c+N)\prod_{i=1}^{c} N_i\right] \text{ where } N = \sum_{i=1}^{c} N_i \tag{16.33}$$

Hence, the computational complexity increases geometrically with the size of the classes. Also this method is considered to be too complex to serve as a basis for call admission control in real multi-service networks.

Integrated Chernoff bound approximation

The cell loss ratio is defined as the mean loss rate, L, to mean offered rate, M: $p_{loss} = L/M$. The mean rate M is given by

$$M = \sum_{i=1}^{c} f_i \rho_i \tag{16.34}$$

where $\rho = \frac{a_i}{a_i+b_i}$ denotes the activity factor for class i.

In the following, we will consider an efficient approximative method for computing L [48]. Let T be a sum of the peak rate f_i of individual sources:

$$T = \sum_{i=1}^{c} N_i f_i \tag{16.35}$$

If $T \leq C$, L is immediately zero by definition. If not, we may transform L into an integral from C to T of the complementary distribution function, $F_c(x) = \mathrm{P}(X > x)$, for a stochastic variable X representing the input rate.

$$L = -\int_C^T (x - C)\mathrm{d}F_c(x) = \int_C^T F_c(x)\mathrm{d}x \tag{16.36}$$

where $F_c(x)$ obeys the Chernoff bound:

$$F_c(x) = P(X > C) = \int_C^\infty f(x)\mathrm{d}x \leq \int_0^\infty e^{s(x-C)} f(x)\mathrm{d}x = e^{-sC} \int_C^\infty e^{sx} f(x)\mathrm{d}x = \frac{E[e^{sX}]}{e^{sC}} \tag{16.37}$$

Hence, we have

$$L < E[e^{sX}] \int_C^T e^{-sx}\mathrm{d}x < \frac{E[e^{sX}]}{se^{sC}} \tag{16.38}$$

where we have replaced T with infinity to obtain the final inequality. Let $f(s)$ denote the right hand side of Equation (16.38), which is a parametric upper bound of L.

$$f(s) = \frac{E[e^{sX}]}{se^{sC}} = \frac{E[e^{s(X-C)}]}{s} \tag{16.39}$$

If the input rate is larger than C, then $f(s)$ has the minimum value at $s = s^*$ that is the root of the equation

$$\frac{\mathrm{d}\,\log E[e^{sX}]}{\mathrm{d}s} - \frac{1}{s} - C = 0 \tag{16.40}$$

Proof: The second derivative of $f(s)$ is always positive for $s > 0$, and hence $f(s)$ is a convex function for $s > 0$. From Equation (16.39), $\lim_{s\to 0} f(s) = \infty$, and if $T > C$, then $\lim_{s\to\infty} f(s) = \infty$. Function $f(s)$ thus has the minimum value with respect to s. s^* minimizing $f(s)$ is the root of $\mathrm{d}f(s)/\mathrm{d}s = 0$. Applying Equation (16.39) yields Equation (16.40).

If $T > C$, then we use the minimum value of $f(s)$ as the best approximation of L. Thus we have

$$p_{loss} = \frac{\text{Min}_{s>0} f(s)}{M} = \frac{f(s^*)}{M} \tag{16.41}$$

This approximation is referred to as the Integrated Chernoff Bound (ICB).

The moment generating function $E[e^{sX}]$ is obtained as

$$E[e^{sX}] = \prod_{i=1}^{c} E[e^{sX_i}] = \prod_{i=1}^{c} [(1-\rho_i) + \rho_i e^{sf_i}]^{N_i} \tag{16.42}$$

In conclusion, we have

$$L = \frac{\prod_{i=1}^{c} [(1-\rho_i) + \rho_i e^{s^* f_i}]^{N_i}}{s^* e^{s^* C}} \tag{16.43}$$

where s^* is the root of the following equation:

$$\sum_{i=1}^{c} \frac{N_i \rho_i f_i e^{sf_i}}{\rho_i (e^{sf_i} - 1) + 1} - \frac{1}{s} - C = 0 \tag{16.44}$$

This nonlinear equation can be solved by Newton's method. The above two formulas reduces the p_{loss} computational complexity to order $O(c)$. The ICB method has sufficiently low complexity to serve as the basis for call admission control in real multi-service networks.

An alternative to the ICB method is the Integrated Large Deviation (ILD) method, which provides slightly tighter cell loss ratio bounds while having the same computational complexity.

FBM queuing model

In this section we derive a formula for the buffer overflow probability $\text{P}(Q > x)$ for a FIFO multiplexer offered FBM traffic with mean rate m, variance coefficient a and Hurst parameter H. The amount of traffic arriving to the multiplexer in the interval $[0, t)$ is given by $A_t = mt + \sqrt{am} Z_t$, where Z_t is a fractional brownian motion process.

We have the trivial lower bound [50]

$$\text{P}(Q_t > x) \geq \text{Max}_{t \geq 0} \text{P}(A_t > Ct + x) \tag{16.45}$$

Which can be expressed as

$$\begin{aligned}
&\mathrm{Max}_{t\geq 0}\mathrm{P}(A_t > Ct + x) = \mathrm{Max}_{t\geq 0}\mathrm{P}(mt + \sqrt{am}Z_t > Ct + x) = \\
&\mathrm{Max}_{t\geq 0}\mathrm{P}(Z_t > \frac{(C-m)t + x}{\sqrt{am}}) = \left(\begin{array}{l} E[Z_t] = 0 \\ E[Z_t^2] = |t|^{2H} = t^{2H} \end{array} \right) = \\
&\mathrm{Max}_{t\geq 0}\overline{\Phi}(\frac{(C-m)t + x}{\sqrt{am}t^H}) = \left(\text{Max when } t = \frac{Hx}{(1-H)(C-m)} \right) = \\
&\overline{\Phi}\left[\frac{(C-m)\frac{Hx}{(1-H)(C-m)} + x}{\sqrt{am}\left[\frac{Hx}{(1-H)(C-m)}\right]^H} \right] = \overline{\Phi}\left[\frac{x^{1-H}(C-m)^H}{\sqrt{am}\kappa H} \right] \qquad (16.46)
\end{aligned}$$

where $\kappa(H) = H^H(1-H)^{1-H}$ and $\overline{\Phi}(y)$ is the residual distribution function of the standard Gaussian distribution:

$$\overline{\Phi}(y) \sim \exp(-y^2/2) \qquad (16.47)$$

which yields the result

$$\mathrm{P}(Q > x) \sim \exp\left(-\frac{(C-m)^{2H}}{2\kappa(H)^2 am} x^{2-2H} \right) \qquad (16.48)$$

Hence, the buffer overflow probability decays like a Weibull distribution. For Poisson traffic $H = 0$ and the distribution decays like an exponential distribution. For self-similar traffic $0.5 < H < 1$ and the distribution decays slower than an exponential distribution. We observe that for self-similar traffic, the buffer overflow distribution decays in the same way as the autocorrelation $r(k)$ function decays with the time lag k. In the section on the fluid flow queuing model we saw that the buffer overflow probability for Markov fluid traffic decays exponentially with the buffer size. This has implications on the dimensioning of buffers. Apparently, compared to Poisson traffic and Markov fluid traffic, self-similar traffic requires the use of larger buffers to obtain a certain buffer overflow probability.

The *statistical multiplexing gain* is defined as the ratio of number of sources admitted under statistical multiplexing to the number of sources admitted under deterministic multiplexing (peak rate allocation). Theoretical as well as simulation results indicate that there is a statistical multiplexing gain for FBM sources but it is not as large as for Markov sources. Generally, the gain increases as the same bandwidth is shared among more and more sources (assuming the per-source fraction of the link capacity decreases) while keeping the overall load constant.

16.5 End-to-end performance measures

In this section we present formulas for the end-to-end packet loss probability (ratio), end-to-end buffer overflow probability, end-to-end mean delay, end-to-end $(1-\alpha)$ quantile of delay, and end-to-end packet delay variation.

Let ϵ_i denote the packet loss probability or the buffer overflow probability in node i. Assume the end-to-end path consists of N nodes. If we assume the buffer occupancy on successive links to be independent, we can compute the end-to-end packet loss probability or buffer overflow probability as:

$$\epsilon_{path} = 1 - \prod_{i=1}^{N}(1-\epsilon_i) \tag{16.49}$$

The end-to-end mean packet delay, D, is computed as:

$$D = D_{min} + D_{queueing} \tag{16.50}$$

where D_{min} denotes the minimum end-to-end packet delay given by [52]

$$\begin{aligned} D_{min} = D_{packetization} + D_{propagation} + D_{transmission} + \\ D_{reassembly} + \sum_{i=1}^{N} D_{switching,i} \end{aligned} \tag{16.51}$$

and $D_{queueing}$ denotes the sum of queuing delay experienced along the path:

$$D_{queueing} = \sum_{i=1}^{N} \mu_i \tag{16.52}$$

The packetization delay is the time it takes to accumulate enough bits to fill a packet. The propagation delay occurs due to the speed of light in the transmission medium and depends on the distance between the source and destination. Transmission delay depends on the speed (capacity, bandwidth) of the link and becomes neglible as the transmission speeds increases. The reassembly delay occurs occurs when several packets of a frame (e.g. AAL frame) are collected before they are passed to the upper protocol layers. The switching delay is the total time it takes for a packet to traverse the switch/router. The queuing delay are due to the fact the some packets must wait in a queue to resolve switching conflicts.

The switch/router employs either either cut-through switching or store-and-forward switching. Cut-through switching means that the switch/router starts to forward bits from the packet as soon as it arrives. Store-and-forward switching means that the router/switch waits for the complete packet to arrive before it starts to forward it. In the first case $D_{transmission}$ is given by the transmission delay at the source. In the second case $D_{transmission}$ is given by the sum of transmission delay at the source and the transmission delay at each store-and-forward switch/router.

Theoretically, the probability density function of the end-to-end packet delay can be obtained by taking the convolution of individual packet delay density functions in the switches/routers. But due to signalling constraints, this is not a feasible method to derive the end-to-end $(1-\alpha)$ quantile of delay,

The end-to-end $(1-\alpha)$ quantile of queuing delay, $D_{queueing}(\alpha)$ can be computed using an asymptotic method:

$$\begin{aligned} D_{queueing}(\alpha) &= \sum_{i=1}^{N} \mu_i + \sqrt{\sum_{i=1}^{N} \sigma_i^2} \times Q^{-1}(\alpha) + \\ \text{Max}_{1 \leq i \leq N} &\{D_{queueing,i}(\alpha) - \mu_i + \sigma_i \times Q^{-1}(\alpha)\} \end{aligned} \tag{16.53}$$

where $Q^{-1}(\alpha)$ is the $(1-\alpha)$ quantile of the standard normal distribution, μ_i is the mean and σ_i^2 is the variance of the delay at node i, and $D_{queueing,i}(\alpha)$ is the $(1-\alpha)$ quantile of the queuing delay at at node i. The term $\{D_{queueing,i}(\alpha) - (\mu_i - \sigma_i \times Q_{-1}(\alpha)\}$ is a heuristic compensation term defined as the difference between the actual $(1-\alpha)$ delay quantile and the asymptotic method estimate of the delay quantile.

The end-to-end $(1-\alpha)$ delay is obtained by adding the fixed minimum delay, D_{min}, to the end-to-end $(1-\alpha)$ queuing delay:

$$D(\alpha) = D_{min} + D_{queueing}(\alpha) \tag{16.54}$$

Note that if a playout buffer is used, the playback time point is normally set to $D(\alpha)$.

The effective end-to-end delay D_{eff} experienced by the application user is given by:

$$D_{eff} = D' + D_{coding} + D_{upper-protocols} \tag{16.55}$$

where $D' = D$ or $D' = D(\alpha)$. The coding delay includes any the time to convert a nondigital signal to digital bit patterns. For example, an analog signal generated by voice, audio or video source is sampled and digitized before transmission in the network. The coding delay also includes any delay of the application compression/decompression algorithm (e.g. JPEG, MPEG, MP3). The upper-protocol delay is due to protocol processing at higher layers (e.g. transport layer, application layer).

The end-to-end peak-to-peak packet delay variation (PDV) can also be obtained by the asymptotic method:

$$pdv(\alpha) = \sum_{i=1}^{N} \mu_i + \sqrt{\sum_{i=1}^{N} \sigma_i^2} \times Q^{-1}(\alpha) + \mathrm{Max}_{1\leq i\leq N}\{pdv_i(\alpha) - (\mu_i + \sigma_i \times Q^{-1}(\alpha))\}$$

where $pdv_i(\alpha) = D_{queueing,i}(\alpha)$ is the PDV at node i.

Chapter 17

Equivalent bandwidth

In this section we evaluate the bandwidth requirements of a FIFO multiplexer with complete buffer sharing. The traffic is assumed to be generated by a superposition of general sources from c classes. Traffic class i is described by the number of sources in the class N_i, the peak rate f_i [Mbps] and the mean rate m_i [Mbps]. Note that we do not necessarily impose any correlation structure on the traffic process.

The bandwidth C_{equ} is the minimum aggregate bandwidth that should be allocated to calls such that the overflow probability is less than a target overflow probability, ϵ. That is, we have

$$C_{equ} := \inf\{C : \text{P(overflow)} \leq \epsilon\} \tag{17.1}$$

since P(overflow) is a function of C, the server capacity.

The bandwidth requirement C_{equ} fulfills the criteria:

$$\sum_{i=1}^{c} N_i m_i \leq C_{equ} \leq \sum_{i=1}^{c} N_i f_i \tag{17.2}$$

Hence, the equivalent bandwidth is between the aggregate mean rate and the aggregate peak rate.

17.1 Chernoff equivalent bandwidth

Let X denote a stochastic variable describing the rate of aggregate traffic at an arbitrary instant made up by sources at rate X_i ($X = \sum_{i=1}^{N} X_i$). The the probability of resource overload (congestion probability) on a transmission link with capacity C can be written $\mathrm{P}(X > C)$ and can be interpreted as the fraction of time when the traffic offered to a link exceeds the capacity of the link. The congestion probability can be estimated by Chernoff bound [61]:

$$\mathrm{P}(X > C) = \int_C^\infty f(x)\mathrm{d}x \leq \int_C^\infty e^{s(x-C)} f(x)\mathrm{d}x = e^{-sC} \int_C^\infty e^{sx} f(x)\mathrm{d}x = \frac{E[e^{sX}]}{e^{sC}} \quad (17.3)$$

We define the scaled logarithmic moment generating function as

$$\alpha(s) := \frac{1}{s}\mathrm{log}E[e^{sX}] \quad (17.4)$$

The Chernoff bound can now be written as

$$\mathrm{P}(X > C) \leq e^{s\alpha(s)-sC} \quad (17.5)$$

The tightest Chernoff bound is obtained when the parameter s has the value:

$$s_C := \mathrm{arg\ inf}_{s>0}\{s\alpha(s) - sC\} \quad (17.6)$$

We seek the minimal service capacity such that $\mathrm{P}(X > C) \leq \epsilon := e^{-\gamma}$. In the context of Chernoff bound this condition translates to

$$e^{\mathrm{inf}_{s>0}\{s\alpha(s)-sC\}} \leq e^{-\gamma} = \epsilon \quad (17.7)$$

Thus we obtain the equivalent capacity from the dual optimization

$$C_{equ} := \mathrm{inf}\{C : \mathrm{inf}_{s>0}\{s\alpha(s) - sC\} \leq -\gamma\} \quad (17.8)$$

Since the function $s\alpha(s) - sC$ is monotonously decreasing in the variable C, and thus $s_{C_{equ}}\alpha(s_{C_{equ}}) - s_{C_{equ}}C_{equ} = -\gamma$ holds, one can easily verify that a direct method for evaluating the equivalent capacity can be arrived at:

$$C_{equ} = \mathrm{inf}_{s>0}\left\{\alpha(s) + \frac{\gamma}{s}\right\} \quad (17.9)$$

The optimizing value in (17.5) now depends on the arrival rate X and on the QoS constraint γ:

$$s_\gamma = \text{arg inf}_{s>0} \left\{ \alpha(s) + \frac{\gamma}{s} \right\} \tag{17.10}$$

In order to determine $\alpha(s)$ we need to determine the moment generating function $E[e^{sX}]$. This can be done by direct traffic measurements or analytically, by an ON/OFF source model assumption as in the Section on Integrated Chernoff Bound.

The moment generating function can be obtained by $E[e^{sX}] = R$, where R is given by an exponential weighted moving average:

$$R_n = (1-\beta)R_{n-1} + \beta e^{sX_n/T} \tag{17.11}$$

where β is the weight of the exponential moving average, and $X_n = X[nT, nT + t)$ denotes the number of bits arriving to the multiplexer in the interval $[nT, nT + t)$. Here we assume the measurements occurs periodically with period T Milli seconds. At every measurement instant, the number of bits arriving during t Milli seconds is counted. Of course $t \leq T$.

17.2 Large buffer asymptotic equivalent bandwidth

The large buffer asymptotic is concerned with the estimation of buffer overflow probability when the buffer size get very large [16]. Consider a single-server queuing system with buffer size B and service rate C. Let Q denote the queue length in the system. The large buffer asymptotic states that the decay rate of the logarithm of the buffer overflow probability is for Markov sources asymptotically linear in B, as B increases to infinity:

$$\text{lim}_{B\to\infty} \frac{1}{B} \text{log P}(Q > B) = -s_{\infty,C} \tag{17.12}$$

where $s_{\infty,C}$ is the rate function of the tail probability. It is computed as

$$s_{\infty,C} = \text{sup}\{s > 0 : \alpha(s) \leq C\} \tag{17.13}$$

and is parameterized by the total amount of arriving work through $\alpha(s)$ and by the server capacity C.

Kelly [40] has defined the *effective bandwidth* function as

$$\alpha(s,t) = \frac{1}{st}\text{log } E(e^{sX[0,t)}) \tag{17.14}$$

In the large buffer asymptotic case this function is reduced to

$$\alpha(s) = \text{lim}_{t\to\infty}\frac{1}{st}\text{log}E[e^{X[0,t)}] \tag{17.15}$$

due to the assumption of the infinite buffer.

Once the decay rate is determined from optimization defined in (17.13), the buffer overflow probability can be computed according (17.12) for large buffers

$$\text{P}(Q > B) \approx e^{-Bs_{\infty,C}} \tag{17.16}$$

For the computation of the equivalent capacity an optimization is needed with respect to C. However, in this case the optimization can be eliminated analytically:

$$C_{equ} := \text{inf}\{C : -Bs_{\infty,C} \leq -\gamma\} = \alpha(s_{\infty,C_{equ}}) = \alpha(\frac{\gamma}{B}) \tag{17.17}$$

In the particular case of the large buffer asymptotic we get $s_{opt} = s_{\infty,C_{equ}} = \frac{\gamma}{B}$, $t_{opt} = \infty$, $\epsilon = e^{-\gamma}$. Note that the case of large buffer asymptotic is the only example where the equivalent capacity equals the effective bandwidth function taken at the corresponding operating point of the system.

The effective bandwidth $\alpha(s_{opt})$ for $s = \frac{\gamma}{B}$ is obtained from measurements using an exponential weighted moving average of the workload process W_n.

$$W_n = (1-\beta)W_{n-1} + \beta e^{sX_n} \tag{17.18}$$

where β is the weight of the exponential moving average, and $X_n = X[nT, nT + t)$ denotes the number of bits arriving to the multiplexer in the interval $[nT, nT + t)$. Although the parameter t is infinity in the Large buffer asymptotics case it can be approximated with a sufficiently large value. The effective bandwidth is given by $\alpha(s_{opt}) = \frac{1}{s_{opt}}\text{log}W$.

The method of large buffer asymptotic incorporates the effect of multiplexing only partially. It captures the multiplexing gain that arises from the statistical properties of individual sources queued in a large buffer, but fails to reflect the economies of scale due to the superposition of many sources.

In the case of self-similar sources, the overflow probability decays like a Weibull distribution:

$$P(Q > B) \approx e^{-s_{\infty,\gamma} B^{2(1-H)}} \tag{17.19}$$

The Weibull decay behavior yields a new formula for the equivalent capacity which we not not consider here.

17.3 Many sources asymptotic equivalent bandwidth

The method of many sources asymptotic holds the promise of taking full advantage of statistical multiplexing gain. It encompasses all the statistical mechanisms that the large buffer asymptotic builds on and takes a step further by exploiting the gain arising from multiplexing of large number of calls. Therefore, it can be expected that the approximation of equivalent capacity delivered by the many sources asymptotic is the most precise of all in the the large deviation context. The following presentation is valid for general traffic sources, including Markov sources and self-similar sources.

Let the stochastic process $X[0,t)$ denote the total amount of bits arriving in the interval $[0,t)$ from N independent sources at a buffered communication link with buffer size B and transmission capacity C. The probability of buffer overflow in this finite-size buffer queuing system can be deduced from the proportion of time over which the the queue length $Q(N,C)$ is above level B in a queue of infinite buffer. The many sources asymptotic states that the decay rate of the logarithm of the corresponding overflow probability is asymptotically linear in the number of sources N in a system where the per-source buffer $b = \frac{B}{N}$ and the per-source capacity $c = \frac{C}{N}$ are kept constant [13, 61]:

$$\text{lim}_{N\to\infty} \frac{1}{N} \text{log P}(Q(N,cN) > bN) = \text{sup}_{t>0} \text{inf}_{s>0} \{st\alpha_v(s,t) - s(b+ct)\} := -I \tag{17.20}$$

Here I is called the asymptotic rate function that depends on the per-source system parameters and the scaled arrival process. The term $\alpha_v(s,t)$ describes the effective bandwidth of a virtual source, the third scaling constant of the system besides b and c. For the aggregate arrival process $X[0,t)$ made up of N sources the effective bandwidth of a virtual source is given by $\alpha_v(s,t) = \frac{\alpha(s,t)}{N}$, where $\alpha(s,t)$ is the effective bandwidth of $X[0,t)$ from (17.14).

Equation (17.20) practically means that for N large the probability of overflow can be approximated by

$$\mathrm{P}(Q(N,C)) \approx e^{-NI} \tag{17.21}$$

where the exponent can be computed as $-NI = \sup_{t>0}\inf_{s>0} J(s,t)$ with

$$J(s,t) := st\alpha(s,t) - s(B+Ct) \tag{17.22}$$

Introducing the optimizing values

$$\begin{aligned} s_{B,C}(t) &:= \arg\,\inf_{s>0} J(s,t) \\ t_{B,C} &:= \arg\,\sup_{t>0} J(s_{B,C}(t),t) \end{aligned} \tag{17.23}$$

the corresponding decay rate $-NI$ can be expressed as $J(s_{B,C}(t_{B,C}), t_{B,C})$. The extremising values $t_{B,C}$ and $s_{B,C}(t_{B,C})$ (which also depend on the properties of the system parameters B, C and the statistical properties of $X[0,t)$) are commonly termed as the critical time and space scales, respectively. Intuitively, the critical time scale is the most probable time interval over which overflow occurs in the multiplexing system. In other words, it is the most likely length of the busy period prior to overflow. Although other busy periods also contributes to the total overflow, large deviation theory take into account only the most probable one because that is the dominant one in the asymptotic sense. The critical space parameter on the other hand captures the statistical behavior of the arrival process (the amount of achievable statistical multiplexing gain and the burstiness). Critical space parameter values close to zero describe sources that can benefit from statistical multiplexing, while larger values infer a higher bandwidth requirement.

We now show how to obtain the equivalent capacity for the many sources asymptotic method. Starting from the inequality set up for the buffer overflow probability $\mathrm{P}(Q(N,C)) \leq e^{-\gamma}$, the minimum service rate has to be determined for which the QoS constraint ($\epsilon = e^{-\gamma}$) is still satisfied. The equivalent capacity is given by

$$C_{equ} := \inf\{C : \sup_{t>0}\inf_{s>0} J(s,t) \leq -\gamma\} \tag{17.24}$$

Similarly to the bufferless case, considering that the function $J(s,t)$ is monotonously decreasing in C, the equality $J(s_{B,C}(t_{B,C}),t_{B,C}) = -\gamma$ can be observed and hence the equivalent capacity is obtained through a double optimization only:

$$C_{equ} = \mathrm{sup}_{t>0}\mathrm{inf}_{s>0}K(s,t) := K(s_{B,\gamma}(t_{B,\gamma}),t_{B,\gamma}), \tag{17.25}$$

where

$$\begin{aligned} s_{B,\gamma}(t) &:= \text{arg inf}_{s>0}K(s,t) \\ t_{B,\gamma} &:= \text{arg sup}_{t>0}K(s_{B,\gamma}(t),t) \end{aligned} \tag{17.26}$$

and the $K(s,t)$ function is defined as

$$K(s,t) := \alpha(s,t) + \frac{\gamma}{st} - \frac{B}{t} \tag{17.27}$$

and

The effective bandwidth $\alpha(s_{B,\gamma},t_{B,\gamma})$ can be obtained from measurements of the workload process W_n exactly as was shown for the large buffer asymptotic method.

17.4 Fluid flow equivalent bandwidth

As shown in the section on the fluid flow queuing model the following equation is valid:

$$\begin{aligned} & z(\mathbf{k})(C - \sum_{i=1}^{c}\frac{N_i}{2}f_i) - \sum_{i=1}^{c}\frac{N_i}{2}(a_i+b_i) = \\ & \sum_{i=1}^{c}(k_i - \frac{N_i}{2})\sqrt{(z(\mathbf{k})f_i+b_i-a_i)^2+4a_ib_i} \end{aligned} \tag{17.28}$$

Solving for the capacity C we obtain

$$C = \sum_{i=1}^{c}\frac{(z(\mathbf{k})f_i+a_i+b_i)N_i + (2k_i-N_i)\sqrt{(z(\mathbf{k})f_i+b_i-a_i)^2+4a_ib_i}}{2z(\mathbf{k})} \tag{17.29}$$

We also saw that for large buffers the following asymptotic approximation is valid:

$$G(x) = \mathrm{P}(Q > x) \sim -a_{\mathbf{0}}\mathrm{exp}(z(\mathbf{0})x) \sum_{\mathbf{k}\in S} (\phi_{\mathbf{0}})_{\mathbf{k}} = \beta \mathrm{exp}(z(\mathbf{0})x) \tag{17.30}$$

where $z(\mathbf{0})$ is the largest negative eigenvalue and $\phi_{\mathbf{0}}$ its associated eigenvector.

Recall that the equivalent capacity C_{equ} is the smallest capacity for which P(overflow) $\leq \epsilon$. To apply formula (17.32) we need the largest negative eigenvalue for the queuing system which has capacity equal to C_{equ} corresponding to a buffer overflow probability of ϵ. Hence, by approximating the coefficient β with one we have that $e^{z(\mathbf{0})x} = \epsilon = e^{-\gamma}$, which yields the largest negative eigenvalue [28]

$$z(\mathbf{0}) = -\gamma/x \tag{17.31}$$

where $\epsilon = e^{-\gamma}$.

The fluid flow equivalent capacity can now be written as

$$C_{eqv,fluid} = \sum_{i=1}^{c} \frac{(z(\mathbf{0})f_i + a_i + b_i)N_i - N_i\sqrt{(z(\mathbf{0})f_i + b_i - a_i)^2 + 4a_ib_i}}{2z(\mathbf{0})} \tag{17.32}$$

where $z(\mathbf{0})$ is given by formula (17.31).

The fluid flow equivalent capacity formula tends to overestimate the required bandwidth when many sources with relatively long burst periods are multiplexed. Fortunately, in this case the required bandwidth can be obtained by a stationary approximation [28].

In the stationary approximation source from class i is characterized by their mean rate m_i and their standard deviation σ_i. In the case of ON/OFF sources we have $\sigma_i^2 = m_i(f_i - m_i)$. With $N = \sum_{i=1}^{c} N_i$ calls, the problem is determining the total bandwidth required by the N calls, C_0, such that probability of instantaneous aggregate bit rate exceeding C_0 is less than a given value ϵ.

Let X be a stochastic variable denoting the aggregate bit rate of the c classes. Then the problem is to determine the value of C_0 such that Pr$(X > C_0) \leq \epsilon$. Assuming that the aggregate bit rate distribution is Gaussian, we have

$$\begin{aligned} \mathrm{Pr}(X > C_0) = \mathrm{Pr}((X - m)/\sigma > (C_0 - m)/\sigma) \approx \\ \mathrm{Pr}(X_{01} > (C_0 - m)/\sigma) = \mathrm{Pr}(X_{m\sigma} > C_0) \end{aligned} \tag{17.33}$$

where $m = \sum_{i=1}^{c} N_i m_i$ and $\sigma^2 = \sum_{i=1}^{c} N_i \sigma_i^2$, and X_{01} and $X_{m\sigma}$ are Gaussian stochastic variables with mean and standard deviation (0,1) and (m, σ), respectively.

The congestion probability can be written

$$\text{Pr}(X > C_0) \approx \text{Pr}(X > m + \alpha\sigma) = \frac{1}{\sqrt{2\pi}} \int_{\frac{C_0 - m}{\sigma}}^{\infty} \text{exp}\left(-\frac{y^2}{2}\right) \text{d}y \approx \epsilon \tag{17.34}$$

Hence, $C_0 = m + \alpha\sigma$. The parameter α is the inverse of the Gaussian distribution. Various formulas have been developed for obtain its value approximately. An accurate approximation is

$$\alpha = \sqrt{2\text{log}(1/\epsilon) - \text{log}(2\pi)} \tag{17.35}$$

For most practical cases of interest, C_0, is an upper bound of the actual bandwidth required to have a buffer overflow probability of less than or equal to ϵ. This is mainly due to the fact the the buffer size is not considered and the in reality the aggregate bit rate is allowed to exceed the C_0 for some period of time until the buffer becomes full, thereby absorbing some of the inaccuracy introduced with this method.

The equivalent capacity under the fluid flow approximation and the stationary approximation can be combined as

$$C_{equ} = \text{Min}(C_{equ,fluid}, C_{equ,stationary}) \tag{17.36}$$

where $C_{equ,stationary}$ refers to the value of C_0.

17.5 FBM equivalent bandwidth

In the section of performance evaluation of a FIFO multiplexer loaded with FBM traffic we obtained the result

$$\text{P}(Q > x) \sim \text{exp}\left(-\frac{(C - m)^{2H}}{2\kappa(H)^2 am} x^{2-2H}\right) \tag{17.37}$$

By setting $\text{P}(Q > x) = \epsilon$ and solving for C we get the FBM equivalent capacity [50]

$$C_{equ} = m + (\kappa(H)\sqrt{-2\text{log}\ (\epsilon)})^{\frac{1}{H}} a^{\frac{1}{2H}} x^{\frac{-1-H}{H}} m^{\frac{1}{2H}} \tag{17.38}$$

where $\kappa(H) = H^H(1-H)^{1-H}$. This result is in accordance with the result of Kelly who applied the effective bandwidth definition (17.14).

Chapter 18

Network dimensioning

18.1 Dimensioning in circuit-switched networks

18.1.1 General considerations

In this section we describe dimensioning in circuit-switched networks. The algorithms also apply to packet-switched networks with virtual circuits such as ATM networks. Even QoS enhanced IP networks which use the concept of flows, which are the equivalents of ATM virtual circuits, may use the dimensioning algorithms designed for circuit-switched networks.

A set of virtual networks can be overlayed on the physical network to simplify resource management. A given virtual network is typically targeted at a specific QoS class. For example, in ATM networks, separate virtual networks can be maintained for CBR, VBR, ABR and UBR services, respectively. The topology of the virtual networks may be different from the physical network topology. Each virtual network link may be carried by one or several physical links. Each physical link is in general shared between multiple virtual networks. The bandwidth allocation at the call layer may be based on complete sharing (CS), complete partitioning (CP) or partial sharing (PS). The bandwidth allocation at the packet layer may be based on deterministic multiplexing or statistical multiplexing.

Resource allocation to virtual and physical networks is referred to as virtual/physical link capacity dimensioning [21, 39]. The first step in network dimensioning is the design of the topological structure of the network, i.e. where to place the nodes and how to interconnect them. In most cases the location of the nodes is given for political or historical reasons. So only the structure of the interconnection graph has to be determined. This can be done through methods of topological optimization and graph theory. By performing this step, connectivity and reliability constraints and link costs have to be regarded. The link cost information is simply a fixed node interconnection cost per unit length, depending on the used technology. For ATM in most cases the result of the topological design phase will lead to a partly meshed backbone network structure. The second step of network dimensioning determines, given the network topology, traffic demand and GoS requirements, the capacities (bandwidths) of the physical and virtual network links.

18.1.2 Problem formulation

Let us consider a fixed infrastructure network with N nodes and P physical links. The capacity of physical link $m \in M = \{1, \cdots, P\}$ is denoted C_m. A set of virtual networks overlay the physical network. The total number of virtual links over all virtual networks is denoted V. The capacity of virtual link $s \in S = \{1, \cdots, V\}$ is denoted by L_s.

The system is assumed to be offered traffic from K call classes. Call class $j \in J = \{1, \cdots, K\}$ is described by:

- Virtual network identifier;
- Origin-destination node pair;
- Poisson call arrival process with rate λ_j;
- Exponentially distributed call holding time with mean $1/\mu_j$;
- Link s bandwidth requirement b_j^s [Mbps];
- Revenue parameter $w_j \in (0, \infty)$;
- Set of alternative routes W_j;

The objective used in dimensioning of the virtual and physical networks can be of several types. Two common examples are maximization of the average revenue and minimization of total network link cost. The first type objective of objective function can in case of virtual network dimensioning be written as:

$$W(L,\Upsilon) = \sum_{j \in J} \lambda_j w_j [1 - B_j(L,\Upsilon)] \tag{18.1}$$

where $L = (L_1, \cdots, L_V)$ denotes the vector of virtual link capacities and Υ denotes a set of CAC and routing parameters. For example, in MDP routing Υ refers to the set of reward parameters, and in LLR routing Υ refers to the set of trunk reservation parameters or external blocking factors. Note that although the CAC and routing strategy is specified, its optimal parameters may be a function of network flow distributions and link dimensions which are not known in advance. Thus, in general, the CAC and routing parameters, Υ, should also be treated as optimization variables.

The second type of objective function can in case of virtual network dimensioning be written as:

$$c(L,\Upsilon) = \sum_{s \in S} c_s(L_s,\Upsilon) \tag{18.2}$$

where c_s denotes a cost function for link s.

The GoS constraints are either expressed in a absolute or relative manner. An absolute GoS constraint specifies that the absolute call blocking probability for a given call category and OD pair should be less than a given value:

$$B_j(L,\Upsilon) \leq B_j^c \tag{18.3}$$

In this case, the dimensioning is carried out for the set of virtual networks, but need not to be carried out for the physical network. Instead, each physical link capacity is given by the sum of capacities of virtual links sharing the link.

The relative GoS constraints specify for each call class, the ratio α_j between the call blocking probability of class j and the maximum call blocking probability over all classes.

$$\frac{B_j(L,\Upsilon)}{B_j^{max}(L,\Upsilon)} = \alpha_j \tag{18.4}$$

In case of relative GoS constraints, the physical network should be dimensioned first using absolute GoS constraints. Second, the set of virtual network link capacities should be determined, using the set of relative GoS constraints together with a bandwidth constraint.

The sum of the capacities of virtual links sharing a given physical link m should not exceed the physical link capacity:

$$\sum_{s \in m} L_s \leq C_m \tag{18.5}$$

18.1.3 Grade of Service models

In general the network dimensioning procedures require network performance models since the network design should meet the call blocking probability constraints. The fixed point equations is based on statistical link independence resulting in decomposition of the network model into a set of link loading functions, f_l, and link performance functions, f_p:

$$\mathbf{A}^s = f_l(\mathbf{\Pi}, \pi) \tag{18.6}$$

$$\Pi^s = f_p(\mathbf{A}^s, L^s) \tag{18.7}$$

where $\mathbf{A}^s = [A^s_j, j \in J]$ denotes the offered traffic to link s, Π^s denotes the set of required performance characteristics of link s (e.g. state probabilities or blocking state probabilities), π denotes the routing policy, and L^s denote the link capacity of link s. The fixed point equations can be solved using repeated substitutions.

Most dimensioning procedures assume statistical link independence resulting in a set of link performance functions coupled by means of link loading functions and possibly link dimensioning functions of link capacities are not used as optimization variables. The link dimensioning function is usually defined by the inverse of the link performance function

$$L^s = f_d(\mathbf{A}^s, \Pi^s), s \in S \tag{18.8}$$

18.1.4 Optimization framework

The problem of finding vector of link capacities where $L = (L_1, \cdots, L_V)$ that optimizes the non-linear objective function under the given set of non-linear constraints can be solved using Sequential Quadratic Programming (SQP).

The basic idea of SQP is to model the nonlinear programming problem (NLP) at a given approximate solution, say x^k, by a quadratic programming subproblem, and then to use the solution to the subproblem to construct a better approximation x^{k+1}. This process is iterated to create a sequence of approximations that, it is hoped for, will converge to the solution $x*$. The nonlinear constraints are replaced by a linear first order Taylor series approximation and the nonlinear objective is replaced by a second order Taylor series approximation augmented by second order information from the constraints. Perhaps the key to understanding the performance and theory of SQP is that fact that, with an appropriate choice of quadratic subproblem, the method can be viewed as the natural extension of Newton and quasi-Newton methods to the constrained optimization setting.

18.2 Dimensioning in IP networks

The dimensioning problem in best effort IP networks can be stated as "Dimension the link capacities such that the mean delay of a packet is minimized given an upper bound of the total cost of the system"

The parameters of the system are as follows. There are P links, $i = 1, \cdots, P$ in the network. The distribution of the packets is assumed to be exponential with mean size $1/\mu$ [bits]. The output buffers of the routers are modeled as M/M/1 queues:

λ_i = packet arrival rate on link i

C_i = capacity of link i

d_i = specific bandwidth price of link i (Euro/bps)

$1/\mu$ = mean size of packet [bits]

Λ = total rate of packets arriving to the network

The average sending time of a packet on link i is $1/(\mu C_i)$ which implies that the capacity of the link is μC_i [packets/s]. The mean sojourn time of a packet on link i (queuing+transmission) is

$$T_i = \frac{1}{\mu C_i - \lambda_i} = \frac{1}{\mu C_i} \frac{1}{1 - \lambda_i / \mu C_i} \tag{18.9}$$

One wishes to minimize the average time T an arriving packet spends in the network:

$$T = \frac{1}{\Lambda} \sum_i \lambda_i T_i \tag{18.10}$$

According to Little's result $\lambda_i T_i$ is the average number of packets (in queue + being transmitted) in buffer i, and $\sum \lambda_i T_i$ is the average number of packets in the whole network.

The constraint for the optimization is that the total cost may not exceed a given limit D:

$$\sum_i d_i C_i \leq D \tag{18.11}$$

It is clear that the delay is minimized if all the money is spent to increase the link capacities. Thus the inequality constraint can be replaced by an equality constraint:

$$\sum_i d_i C_i = D \tag{18.12}$$

The task is to find capacities C_i such that under this constraint T is minimized. The constraint of the minimization problem can be taken into account by the method of Lagrangian multiplier. To the objective function we add the function defining the constraint multiplied by a coefficient β which so far is undetermined. The function to be minimized is thus

$$G = \frac{1}{\Lambda} \sum_i \frac{\lambda_i}{\mu C_i - \lambda_i} + \beta \left(\sum_i d_i C_i - D \right) \tag{18.13}$$

Taken the derivative of G with respect to C_i we get

$$\frac{\partial G}{\partial C_i} = \frac{1}{\Lambda} \frac{-\lambda_i \mu}{(\mu C_i - \lambda_i)^2} + \beta d_i = 0, i = 1, \cdots, P \tag{18.14}$$

$$\Rightarrow (\mu C_i - \lambda)^2 = \frac{\lambda_i \mu}{\Lambda \beta d_i} \tag{18.15}$$

$$\Rightarrow C_i = \frac{\lambda_i}{\mu_i} + \frac{1}{\sqrt{\Lambda \beta \mu}} \sqrt{\frac{\lambda_i}{d_i}} \tag{18.16}$$

Now solve β from the requirement

$$D = \sum_i d_i C_i = \sum_i d_i \frac{\lambda_i}{\mu_i} + \frac{1}{\sqrt{\Lambda \beta \mu}} \sum \sqrt{\lambda_i d_i} \tag{18.17}$$

$$\Rightarrow \frac{1}{\sqrt{\Lambda \beta \mu}} = \frac{D - \sum_i d_i \frac{\lambda_i}{\mu}}{\sum_i \sqrt{\lambda_i d_i}} \tag{18.18}$$

We note that λ_i / μ is the mean bit rate on link i; at least this capacity is needed on link i in order to carry the load. Correspondingly, $\sum_i d_i \frac{\lambda_i}{\mu}$ is the minimum cost of the system.

Denote

$$D_e = D - \sum_i d_i \frac{\lambda_i}{\mu} \tag{18.19}$$

This is the excess money left over for optimization, when the minimum capacities have been allocated. In terms of D_e the constraint gives

$$\frac{1}{\sqrt{\Lambda \beta \mu}} = \frac{D_e}{\sum_i \sqrt{\lambda_i d_i}} \tag{18.20}$$

Substitute this back to the expression for the optimal C_i:

$$C_i = \frac{\lambda_i}{\mu} + D_e \frac{\sqrt{\lambda_i / d_i}}{\sum_{j=1}^{P} \sqrt{\lambda_j d_j}} \tag{18.21}$$

That is, the link capacity exceeds the minimum value λ_i/μ by an amount which is proportional to $\sqrt{\lambda_i/d_i}$. The mean sojourn time in the optimized network is

$$T_{min} = \frac{1}{\Lambda \mu D_e} \left(\sum_{i=1}^{P} \sqrt{\lambda_i d_i} \right)^2 \tag{18.22}$$

In the special case when all the links have the same specific cost, one can set $d_i = 1$ which gives $D = C$, i.e. the total available capacity. Further denote

$$\rho = \frac{1}{C} \sum_{i=1}^{L} \frac{\lambda_i}{\mu} \tag{18.23}$$

Then the formulas take a simpler form

$$C_i = \frac{\lambda_i}{\mu} + (1-\rho) C \frac{\sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}} \tag{18.24}$$

$$T_{min} = \frac{\left(\sum_i \sqrt{\lambda_i}\right)^2}{(1-\rho)\Lambda C \mu} \tag{18.25}$$

Chapter 19

Charging and pricing in multi-service networks

19.1 Charging and pricing

Charging, pricing, accounting and billing are crucial features of telecommunication services. How should the network provider design tariffs for the range of services offered? This is partly a marketing decision – tariffs must be attractive to customers – but network providers are also concerned with efficiency and cost-recovery. Charging schemes should encourage efficient use of the network and should generate revenue in a fair way according to the relative usage of customers.

In multi-service networks, tariffs might depend on a number of parameters defining the traffic and QoS characteristics of a call, in order that charges should reflect network resource usage. The way that a customer uses the network depends on the tariffs and also on how the customers values each type of call (the customer's *utility*, in the language of economics). The interplay between tariffs, network resource usage, and customer incentives is a fertile area for economic and mathematical models.

Multi-service networks need to include facilities for charging, pricing, accounting, and billing [67]. In this context, *charging* designates the evaluation of costs for the call. The cost is calculated based on some characteristics of the call, according to a *charging scheme*, which in turn is part of a tariffing policy. Pricing is the process of assigning a price (expressed in monetary units) to a specific service. This process combine technical considerations, such as the amount of resources used for a service, and economical considerations, such as applying tariffing theory and marketing methods. *Accounting* involves gathering information necessary so that the total charge can be itemized against tariffs and usage measurements. *Billing* involves collecting charge information over a given period and communicating this to the customer in the form of a bill. Another important concept is *advise of charge*, where a customer can be given on request the charge for a specific call (whether intended, ongoing or just completed).

A tariffing policy may include both call charges and subscription charges. We will say that charging is *usage-based* when the call charges are included. Usage-based charges may also include subscription charges that are not related to usage.

19.2 Incentive compatibility

The role of charging is not only to cover the costs of service provision and generate income for the service provider, but also to influence the way customers use network services. This happens as each individual customer reacts to tariff and seeks to minimize charges. The tariff structure should provide the right incentives for users to use network resources efficiently. This is the key idea of *incentive compatibility*. Tariffs should guide customers to select services and use the network in ways that are good for overall network performance.

Tariffs which are not incentive compatible give wrong signals and lead users to use the network in very inefficient ways. One example of this is the Internet which faces intense congestion problems due to its ineffective pricing structure, which is based primarily on flat-rate pricing. Under flat-rate pricing, charges depend only on the rate of the access pipe which connects customers to Internet service providers. Such a pricing scheme provides no incentive for users to use less bandwidth than the rate of their access pipe. Furthermore, flat-rate pricing does not enable users to adequately reveal their preferences for network usage. All users are treated the same, event though different users might place different values on the same service. Both of these limitations result in a congested network where resources are not used according to the actual needs of customers.

Usage-based charging is necessary for incentive compatibility, and economic theory suggest that usage-based charging will be employed where there is perfect competition. This is important in view of the worldwide process of deregulation which is increasing the competitive nature of the telecommunication market. However charging schemes will also be determined by marketing and strategic decisions, customer preferences, and the cost and complexity of implementing and operating these schemes.

19.3 The difficulty of charging in multi-service networks

The amount of network resources used by a call and the QoS experienced by the user depend on the statistical properties of the traffic generated by the call. Within the telecommunications and computer industries it is possible to discern two extreme approaches to this issue. One (impractical) approach is to expect the user to provide the network with a full statistical characterization of traffic, in advance, which is then policed by the network. Another approach stresses the difficulty for a user of providing any information on traffic characteristics, and expects the network to cope nevertheless. The correct balance will necessarily involve trade-offs between the user's uncertainty about traffic characteristics and the network's ability to statistically multiplex calls in an efficient manner. A desirable property of a charging scheme would be to encourage the cooperative sharing of information and characterization effort between the user and the network. This can be realized if tariffs encourage cost-minimizing users to make a more accurate characterization of some statistical properties of their traffic. This information can then be used by the network to multiplex user calls more efficiently.

The traffic contract parameters alone will not accurately determine resource usage. This is because the contract only defines the user traffic to lie within specific ranges determined by the traffic description. Since the user will not in general need to produce the worst case input allowed within the contract, charging according to this worst case traffic would not give the right incentives. From the above discussion it becomes clear that in order to create tariffs with the right incentive properties we need to combine traffic contract parameters with actual measurements.

19.4 The cost of charging

An important requirement of a charging scheme is that it is efficiently implementable. By this we mean that the information required by the charging scheme should be easy to obtain and to manipulate. It is well know that accounting and billing are major parts of the total cost of telephone networks. Hence, a prerequisite for a realistic charging scheme is low cost of implementation and operation. Current technology allows sophisticated traffic measurements to be done in hardware, which greatly expands the spectrum of charging functions which are feasible to implement. However detailed statistics are costly to manipulate and store, hence there is a trade-off between amount of statistical information gathered (which would allow more accurate characterization of a user's traffic) and the cost of gathering, storing and manipulating such information.

19.5 Multi-service network costs

The cost of multi-service networks consists of the following [67]:

- The incremental cost of send an extra packet. In the absence of congestion this is essentially zero;
- The congestion costs, or social costs of delaying other users' packets. As above, this cost is zero when there is no congestion;
- The fixed costs of the network infrastructure (e.g. routers, communication lines, maintenance, and management;
- The incremental costs of connection to the network. This involves the cost of the access lines and customer premises equipment needed to connect to the network. This cost represent the largest portion of the total cost for an organization to connect to the multi-service network;
- The cost of expanding the capacity of the network.

Fixed costs constitute the major percentage of the total costs. On the other hand, the marginal or incremental costs are non-zero only in the presence of network congestion. The observation leads to the proposition than under normal (i.e. uncongested) operation, the network charges should include only fixed charges. However, in the presence of congestion, charges should also include a non-zero usage charge which depends on the level of congestion and the magnitude of the users' contribution to it. This effect of congestion on prices can be expressed either with dynamically adjusted prices or time-of-day sensitive prices.

19.6 Work on Internet charging

Bohn et al. [7] present a scheme to differentiate user traffic based on the precedence field in the IPv4 header. Specifically, end users set the precedence field depending on the level of service they require. Intermediate routers maintain more than one queue for packets with different precedence values and implement a priority service discipline rather than a simple FIFO discipline. Packets with a higher precedence are placed in a queue with higher priority, hence experience better service (lower delay) in periods of congestion. A quota system can be used to discourage users from always selecting high precedence. If quotas are related to momentary units the resulting scheme generates charges based on (precedence) priority level *independently* of the level of congestion. An advantage of this proposal is that it can be gradually implemented in parts of the Internet where congestion is a problem.

MacKie-Mason and Varian [44] propose a "smart market" approach to charging. According to this approach, each packet contains a "bid" which indicates how much a user is willing to pay for the transmission of the packet. Routers queue packets in order of decreasing bids, hence packets with a higher bid experience less delay. In case of congestion, the packets offering the lowest bids are discarded first and accepted packets are tariffed at a rate determined by the highest bid among the rejected packets. The cost of carrying each packet is thus related to the marginal value (represented by the bid) of the traffic which is squeezed out.

Odlyzko [51] presents an approach to charging in the Internet called Paris Metro Charging, due to its resemblance to the pricing structure of the Paris Metro. The basic idea is to partition the Internet into several logical networks, each with separate and non-sharable resources. The (fixed) price of bandwidth would be different for each logical network, and the expectation would be that higher priced networks would be loss congested than lower priced networks. Hence, the scheme allows a user who requires better performance to switch to a higher priced, and less congested logical network.

Clark [11] takes a different approach to charging in the Internet. The main objective of the approach is to discriminate users at times of congestion. The scheme allows different users to obtain a different share of the capacity at times of congestion, by purchasing a profile or *expected capacity*. One possible way to express user profiles is the two parameters of a token bucket or leaky bucket. Packets are marked as *in* or *out* depending on whether they are within the user's profile or in excess of it. In the absence of congestion all packets (both those marked with *in* and those marked with *out*) receive the same service. On the other hand, in the presence of congestion, routers preferentially drop packets marked *out* since those packets have exceeded the profile purchased by the user. A user's *expected capacity* represents the capacity the user *expects* to be available to him, and provides a method of allowing users to obtain different shares if the network capacity in periods of congestion. An important advantage of this scheme is that internally the network switches are required to implement a simple scheme where, in periods of congestion, packets marked *out* are preferentially dropped. Buying expected capacity rather than peak rate of the access link allows a customer to vary his profile depending on his demand without changing or upgrading his access link. At the same time, the network provider can better dimension its resources bas on expected capacity that it sells, rather than on the sum of the peak rates of the customers' access links.

Rather than present a specific charging scheme, Shenker et al. [64] focus on structural and architectural issues such as the local control of pricing policies, multicast charging and receiver charging. In the proposed architecture, charges are determined locally at the access point, hence the name *edge pricing*. This is important because more detailed pricing schemes can be implemented and tested at the edge of the network, while maintaining a simple and generic network core. Rather than try to compute the congestion costs, which the authors argue are inherently inaccessible, they propose to approximate them with prices that depend on the QoS, the time-of-day, and the expected path from the source to the destination. In the case of multicast traffic, there is a potentially large and varying group of destination users. One approach for accumulating accounting information to the source would be to have receivers periodically send *accounting messages*. While traveling to the source, each node (router) would add the cost of its downstream link to the accounting message. Branching nodes would add the sum of the costs of all downstream links. Hence, when the accounting packets reach the source they will contain the accounting information for the whole multicast tree. In case of receiver charging, charges can be computed at the "exit" point (the receivers access point). For multicast connections each receiver can be assigned a fraction of the total charge.

19.7 The *abc* charging scheme

The *abc* scheme is a general method for charging guaranteed or elastic services [41, 67, 66]. The charging mechanism comprises a subscription charge and a per-call charge. The subscription charge can be related to many different aspects of the service being purchased, including access rate, the range of services available and the QoS. The per-call charge takes the form

$$aT + bV + c$$

where T and V are the measured duration and the volume of the call, and a, b, c are tariff parameters applying to the call. The tariff parameters a, b, c are agreed between the user and network at the start of the call, dependent on the traffic and service contract. They are static parameters (they are changed by the network only infrequently, typically at intervals measured in months). The tariff parameters have a simple interpretation for the user: a is the charge for duration (for example in euros per second), b is the charge for volume (for example in euros per Mbit). c is a minimum charge for a call.

19.7.1 *abc* scheme for guaranteed services

Let $\overline{\alpha}(m, \mathbf{h})$ be an upper bound for the greatest effective bandwidth possible subject to the traffic source $X(t)$ being constrained to have the mean rate m and to satisfy traffic contract parameters $\mathbf{h}$. An important property of $\overline{\alpha}(m, \mathbf{h})$ is that it is concave in m. This upper bound is an appropriate basis for charging, but in order to construct usable tariffs we must find suitable approximations that simplify the formula. Assume that the source is policed by leaky buckets with parameters (ρ_k, β_k) for $k \in K$, which are part of the traffic contract $\mathbf{h}$, and let $H(t) := \mathrm{Min}_{k \in K}\{\rho_k t + \beta_k\}$. Then a bound, which we shall call the *simple bound*, is given by

$$\overline{\alpha}(m, \mathbf{h}) \leq \frac{1}{st}\log\left[1 + \frac{tm}{H(t)}(e^{sH(t)} - 1)\right] \tag{19.1}$$

For a source whose peak rate f only is policed by a single leaky bucket with parameters $(f, 0)$, we have $H(t) = ft$, and the above bound reduces to

$$\overline{\alpha}(m, \mathbf{h}) \leq \frac{1}{\theta}\log\left[1 + \frac{m}{f}(e^{\theta f} - 1)\right] \tag{19.2}$$

where $\theta = st$. The above approximation is appropriate for the case where buffers in the network are small. We refer to this bound as the *on-off bound*.

Next we describe a charging scheme based on the simple bound (19.1) which is linear in measurement of time and volume. The user is offered traffic corresponding to tangents to this effective bandwidth function. A tangent at the point m has the form $g(m, \mathbf{h}; M) = a(m, \mathbf{h}) + b(m, \mathbf{h})M$, where the coefficients are given by

$$\begin{aligned} b(m, \mathbf{h}) &= \frac{e^{sH(t)} - 1}{s[H(t) + mt(e^{sH(t)} - 1)]} \\ a(m, \mathbf{h}) &= \overline{\alpha}(m, \mathbf{h}) - mb(m, \mathbf{h}) \end{aligned} \tag{19.3}$$

The user is then charged at at rate $a(m, \mathbf{h}) + b(m, \mathbf{h})M$ per unit time, where M is the actual mean rate of the call. This gives a charge of $a(m, \mathbf{h})T + b(m, \mathbf{h})V$ for a call of duration T and volume V. The network may additionally make a fixed charge $c(m, \mathbf{h})$ for each call, which represents the cost to the network, in switching and signalling resources, of establishing the call. The user then sees a charge $aT + bV + c$ arising from the selected tariff (a, b, c), comprising a duration charge, a volume charge, and a per-call charge.

For a given traffic contract the user may be offered several choices of tariff, corresponding to distinct tangents to the effective bandwidth curve. The user's choice should depend on his estimate of the mean rate of the call. A user with a low expected mean rate should choose a tariff with small duration charge a, whereas a user with high expected mean rate should choose a tariff with small volume charge b. In order to minimize the expected charge the user should choose the tariff corresponding most closely to the expected mean rate of the call. The user's choice of tariff thus conveys information to the network which could be used in call admission control. This charging scheme thus provides appropriate incentives to the user to choose the best tariff and to constrain the duration and volume of the call.

19.7.2 *abc* scheme for elastic services

The *abc* scheme can also be applied to ABR charging. The essence of the scheme is that traffic up to the minimum cell rate (MCR) is charged at one rate, while traffic above the MCR is charged at a lower rate. If a resource within the network has spare capacity beyond that required for high-priority and MCR traffic, then it may be shared amongst the ABR calls in proportion to their MCRs. Thus the choice of MCR by a user buys a share of spare capacity, as well as providing a minimum cell rate.

More precisely, we suppose that there is a charge of a times the chosen MCR per unit time, and additionally a charge b per unit volume, where b may be zero. This is precisely equivalent to different charges for volume above and below the MCR under the assumption that the cell rate does not fall below the MCR – in other words, the contracted MCR defines a minimum level for volume charging. Thus traffic above the chooses MCR is charged at a lower rate, possibly a substantially lower rate. To illustrate the properties of this scheme, consider a typical ABR call, such as file transfer of a given size. By choice of MCR a user can obtain an upper bound on the time taken to transfer the file, although the user would expect a much faster transfer if the network were lightly loaded. Note the important feature that both the time taken to transfer the file *and* the total charge to transfer the file will be larger when the network is congested, since the higher charge a applies to a larger volume of the transfer. User may of course complain that they are charged more for a slower service, but this is the key characteristic of any incentive-compatible scheme designed to ease congestion. At times of congestion the user can speed up the file transfer by increasing the chosen MCR for the call, i.e. each user is able to act according to its own trade-off between delay and cost.

19.8 Charging for elastic services

In this sub section we consider the problem of combined charging and rate allocation for elastic services such as ABR.

We consider a network with a set J of links, C_j being the finite capacity of link j, for

$j \in J$. A user of this network is associated with a route r, a non-empty subset of J, and we denotes by R the set of routes (users). We define the 0-1 matrix $A = (A_{jr}, j \in J, r \in R)$ so that $A_{jr} = 1$ if $j \in r$, and $A_{jr} = 0$ otherwise. A user r has an associated effective rate x_r, with implied utility $U_r(x_r)$. We assume that the utility functions are increasing, strictly concave and continuously differentiable functions of x_r over $x_r \geq 0$, and that they are also additive.

The theory of effective bandwidths suggests that a reasonable approximation for the set of feasible assignments of $x = (x_r, r \in R)$ is the set characterized by the constraints $Ax \leq C, x \geq 0$ where C is the vector of link capacities.

The optimal way to assign effective rates in the system is the solution of the problem [**?**]

SYSTEM(U, A, C):

$$\text{Max} \sum_{r \in R} U_r(x_r), \text{ such that } Ax \leq C, x \geq 0 \tag{19.4}$$

While the optimization problem is mathematically tractable, it involves utility functions that are unlikely to be known by the network. Hence we will seek its solution using an interesting decomposition, where the users are only required to know their utility functions. The information between the users and the network involves prices.

Suppose that the user r may choose an amount w_r to pay per unit time to the network, and by doing that he will receive a permission to send with effective flow x_r, where the ratio $w_r/x_r = \lambda_r$, the "price" of the unit flow, is given. Then user r in order to maximize his benefit, needs to solve the following optimization problem

USER$_r(U_r; \lambda_r)$:

$$\text{Max } U_r\left(\frac{w_r}{\lambda_r}\right) - w_r \text{ over } w_r \geq 0 \tag{19.5}$$

Suppose now the the network knows the vector of amounts $(w_r, r \in R)$, and attempts to maximize the logarithmic utility function $\sum_r w_r \log x_r$

NETWORK$(A, C; w)$:

$$\text{Max} \sum_r w_r \log x_r \text{ such that } Ax \leq C, x \geq 0 \tag{19.6}$$

Kelly has shown there exist vectors λ, w, x satisfying $w_r = \lambda_r x_r, r \in R$, such that w solves (19.5), x solves (19.6), and further x is the solution of (19.4). This implies that if the users independently of the network optimize their benefits by solving (19.5) and communicate to the network the amounts w, the network solves (19.6), and communicates to the users the flows x, then the system will have as an equilibrium the optimal operating point of (19.4).

Practical solutions the network problem (19.6) is subject to current research efforts. One attempt is due to Kelly who has suggested an iterative solution of the problem. We refer the interested reader to reference [67].

Appendix A

LAN protocols

IEEE has issued several LAN standards: Ethernet, Token bus and Token ring. The standards specify the physical layer and the data link layer. The data link layer is composed of two sub layers: the lower Medium Access Control (MAC) sub layer and upper Logical Link Control (LLC) sub layer. The MAC sub layer are unique for each LAN standard but the LLC sub layer is the same.

A.1 Ethernet

Ethernet is packet-oriented LAN standard developed in the mid-1970s [55, 68, 70]. Ethernet does only provide non real-time service. Ethernet MAC is based on Carrier Sense, Multiple Access with Collision Detect (CSMA/CD). As the name CSMA indicate, the Ethernet is a multiple-access network, meaning that a set of stations send and receive over a shared link. The "carrier sense" in CSMA/CD means that all the stations can detect between a busy and an idle link state, and "collision detect" means that a station listens as it transmits and can therefore detect when a frame it is transmitting has interfered (collided) with a frame transmitted by another station. Normal Ethernet run over 10-Mbps links. Fast Ethernet run over 100-Mbps links. Gigabit Ethernet run over 1000 Mbps links. They are all included in the IEEE 802.3 standard.

An Ethernet segment is implemented on a coaxial cable of up to 500 m connecting up to 1024 hosts. The host's Ethernet adaptor is connected to a transceiver which is connected to the Ethernet using a tap. Multiple Ethernet segments can be joined using *repeaters*. An

Ethernet *hub* is a multi-way repeater allowing multiple segments to share collision domain. An Ethernet *switch* is a device that allows forwarding of specific frames between segments without a common collision domain.

Bytes	7	1	2 or 6	2 or 6	2	0−1500	0−46	4
	Preamble	SFD	Destination address	Source address	Length/ Type	LLC data	Pad	Checksum

Figure A.1: The Ethernet frame format

The *Preamble* allows the receiver to synchronize with the signal. The *SFD* or *Start Frame Delimiter* identifies the start of the frame. The destination and source *Address* are identified with a 48-bit address. The packet *Length/Type* gives the length of LLC data fields in bytes, or the Ethernet Type field depending on whether the frame conforms to the IEEE 802.3 standard or the earlier Ethernet specification. The *LLC Data* field contains 46 to 1500 bytes of data. The *Pad* field contains bytes to assure that the frame is long enough. Finally each frame includes a 32 bit *Checksum*.

When an adaptor has a frame to send and the link is busy, it waits for the line to go idle and then transmits immediately. Since Ethernet provides no centralized control it is possible that two (or more) adaptors begin transmitting at the same time which results in a frame collision. At the moment an adaptor detects its frame is colliding with another, if first makes sure to transmit a 32 bit jamming sequence and then stops the transmission. Thereafter the adaptor waits a certain amount of time and tries again. Each time is tries but fails, the adaptor doubles the amount of time it waits before it tries again. Adaptors typically retry up to 16 times. This re trial algorithm is known as *exponential back-off*.

A.2 Token Bus

Token bus is a packet-oriented LAN standard called IEEE 802.4 [55, 68, 70]. Its provides, in contrast to Ethernet, real-time services for use in e.g. factory automation. To obtain a good fault tolerance Token bus has a physical bus topology. The stations attached to the bus form a logical ring. The MAC sub layer is based on a special control frame called *token* circulating in the idle logical ring. When a station wants to send frames, it captures the token and may thereafter send frames during a small period called the token holding time (e.g. 10 ms).

Traffic is divided into four priority classes. The stations has a queue for each priority level. The queue with highest priority is served first in FIFO order. Only when all the queues with higher priority are empty, a lower priority queue gets served in FIFO order. The highest priority queue is used for real-time traffic. By appropriate setting of the timers of the token passing algorithm, the real time requirements of the highest priority class can be met.

The protocol automatically handles stations which enter and leave the logical ring, station failure, multiple or missing tokens.

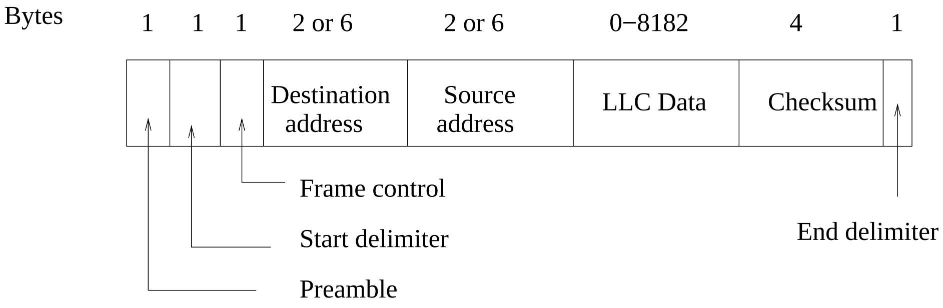


Figure A.2: The Token bus frame format

The *Preamble* is used to synchronize the receivers clock. The *Starting delimiter* and *Ending delimiter* marks the frame boundaries. The *Frame control* field is used to distinguish data frames from control frames. For data frames, it carries the frame's priority. For control frames the field is used to specify the frame types, e.g. token frame or maintenance frame. The *Destination address* and *Source address* fields are the same as for Ethernet. The *LLC data* field may be up to 8182 bytes long. The *Checksum* field is the same as for Ethernet.

A.3 Token Ring

Token ring is a packet-oriented LAN standard called IEEE 802.5 [55, 68, 70]. Stations are connected point-to-point over copper media in a ring topology. Transmission on the ring runs at 4 Mbps or 16 Mbps. Token ring may have up to 250 stations per ring.

The standard token ring protocol does not provide real-time service. However, the standard protocol can be modified to give real-time service.

Each station receives frames from its upstream neighbor and forwards them to its downstream neighbor. The MAC sub layer is based on token management as Token bus. When a station has something to send it waits for and grabs the token. The token is re-transmitted by the token-holding station either immediately after the last frame has been sent or after the frame has gone all the way round the ring. Frames that have circulated one round in the ring is removed by its sending station. The media access algorithm is fair in the sense that as the token circulates the ring, each station gets a chance to transmit. Stations are served in a round-robin fashion.

A special station is appointed monitor station. The monitor can insert additional delay into the ring in order make the ring have enough storage capacity to hold one token. The monitor also detects and re-generates missing tokens, checks for corrupted or orphaned frames, and detects dead stations.

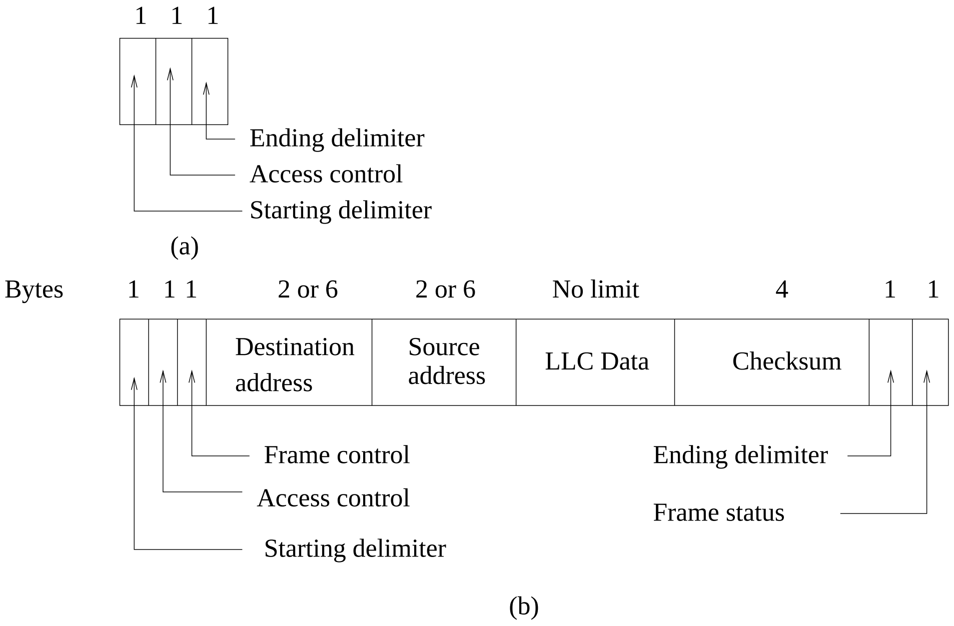


Figure A.3: (a) Token format. (b) Data frame format

The *Staring delimiter* and *Ending delimiter* mark the beginning and ending of the frame. The *Access control* byte contains the token bit, and also the *Monitor bit, Priority bits,* and *Reservation bits*. The monitor bits are used to detect orphan frames. The priority bits are used to distinguish important data from less important. The reservation bits are used for token reservation purposes. The *Frame control* byte distinguishes data frames from various possible control frames. The *Destination address* and *Source address* fields are the same as in Ethernet and Token bus. The *LLC data* field can be arbitrary long. The *Checksum* field is also the same as for Ethernet and Token bus. The *Frame status* byte contains status information on whether a frame as arrived to a certain station, and if the station's interface has copied the frame to the station.

A.4 LLC

The *Logical Link Control* (LLC) is standardized IEEE protocol in the upper part of the data link layer in IEEE 802 LANs and MANs [55, 68, 70]. The LLC sub layer is assumed to run over the MAC sub layer. The objective is to provide service to the network layer which is independent of the actual MAC protocol used. The service options include unreliable datagram service, acknowledge datagram service, and reliable connection-oriented service. The LLC header is based on the older HDLC protocol. Different frame formats are used for different LLC service types. The LLC sub layers on top of two different MAC sub layers (e.g. Ethernet and Token ring) can be connected using a *bridge*.

Appendix B

MAN protocols

This section describes two MAN standards: FDDI and DQDB. The standards specify the physical and data link layers.

B.1 FDDI

Fiber Distributed Data Interface (FDDI) is a connectionless packet-oriented MAN standard [55, 68, 70]. FDDI is similar to Token ring. However, FDDI network consists of a dual ring – two independent rings that transmit data in opposite directions at 100 Mbps. The second ring is not used during normal operation but instead comes into play only if the primary ring fails. In this case, a complete ring is set up by connecting the primary and secondary ring at the two stations closest to the failing station. The FDDI standard limits a single network to at most 500 stations (hosts). with a maximum distance of 2km between any pair of stations. Overall, the network is limited to a total of 200 km of fiber, which means that, because of the dual nature of the ring, the total amount of cable connecting all stations is limited to 100 km. The media of the ring can be fiber, coax or twisted pair.

The rules governing the token holding times is little more complex than for Token ring. The token holding time is defined the same way as for Token ring. An additional feature of FDDI is to limit the token rotation time (TRT), that is, to ensure that a given station has the opportunity to transmit within certain amount of time. Each stations measures the time between two arrivals of the token, called the measured TRT. If this value is less than the limit, called the target TRT, the token is early and the station is allowed to hold the token for the difference between target TRT and the measured TRT.

FDDI defines two classes of traffic: synchronous and asynchronous. Synchronous traffic is delay sensitive. Asynchronous traffic is throughput sensitive. When a stations receives the token it is always allowed to send synchronous data, without regard to whether the token is early or late. In contrast, the station can only send asynchronous data when the token is early. The total amount of synchronous data that can be sent during one token rotation is bounded by one target TRT. This means that in the worst case, the stations with asynchronous traffic first use up one target TRT worth of time, and the the stations with synchronous traffic consume another target TRT worth of time. The stations negotiate through a bidding process to determine the target TRT which is short enough for all stations.

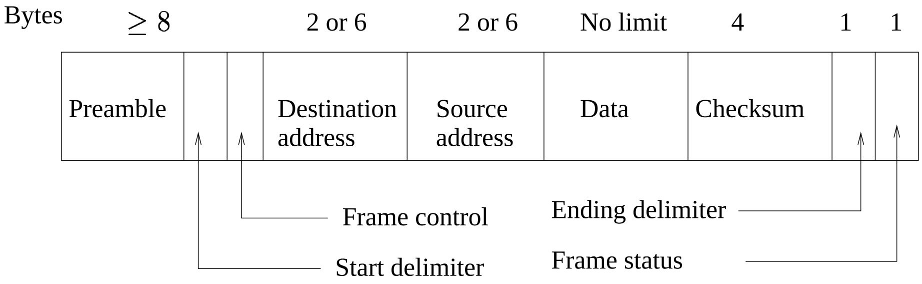


Figure B.1: FDDI frame format

The *Frame status* holds acknowledgments bits, similar to those of Token ring.The other fields are analogous to Token ring.

B.2 DQDB

The *Distributed Queue Dual Bus* (DQDB) is a connection- and packet-oriented MAN standard [70]. It provides a hybrid approach to multi-service networks. The underlying network is synchronous, and portions of the total bandwidth can be reserved for real-time traffic while the remainder can be utilized for data traffic.

The two busses are unidirectional. They each run at 150 Mbps and can be up to 160 km in length. All stations can read and write on both busses. Each stations connects to each bus via an OR-write tap and a passive read tap upstream of the write tap. Stations can only read data passing on the bus but never remove it and only alter it when allowed by the access protocol.

Two slot generators (head-ends), one at the extreme upstream of each bus, generate fixed length empty slots which propagate downstream each channel. Time on each bus is divided into fixed-length slots with a fixed number of slots allocated every 125 μs frame. The DQDB architecture is shown in Figure B.2.

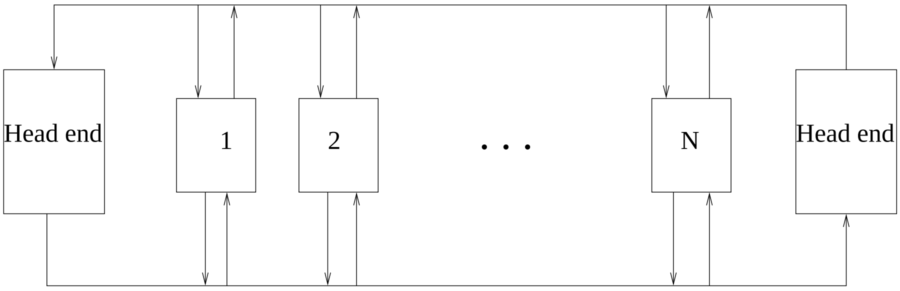


Figure B.2: DQDB architecture

The first 8 bits of each cell is the access control field which indicate the one of two possible types of the cell:

- Pre-Arbitrated (PA) cells
- Queue-Arbitrated (QA) cells

Real-time traffic, e.g. voice, is carried in synchronous PA cells. Data traffic is carried in asynchronous QA cells. QA come in a variety of ways, for example, control, error recovery and data QA cells. Media access for QA cells is done by a distributed queuing algorithm.

The *Ac* field is the access control field which indicates whether the slot is free and its type (PA or QA). The *PA header* and *QA header* provides control information for synchronous and asynchronous cells, respectively. The *SAR* field provides segmentation and reassembly support. The *Data* field contains 44 bytes. The *CRC* field provides protection of the cell contents.

Unlike other 802 LAN protocols, media access of data (QA) cells in DBDB is not greedy. In all others, if a station gets a chance to send, it will. In DQDB, stations queue up in the order they became ready to send and transmit in FIFO order. To simulate the FIFO queue, each station maintains two counters: a *Request counter* denoted *RC* and a *Countdown counter* denoted *CD*. The request counter counts the number of downstream requests pending until the station itself has a frame to send. At that point, *RC* is copied to *CD*, *RC* is reset to 0,

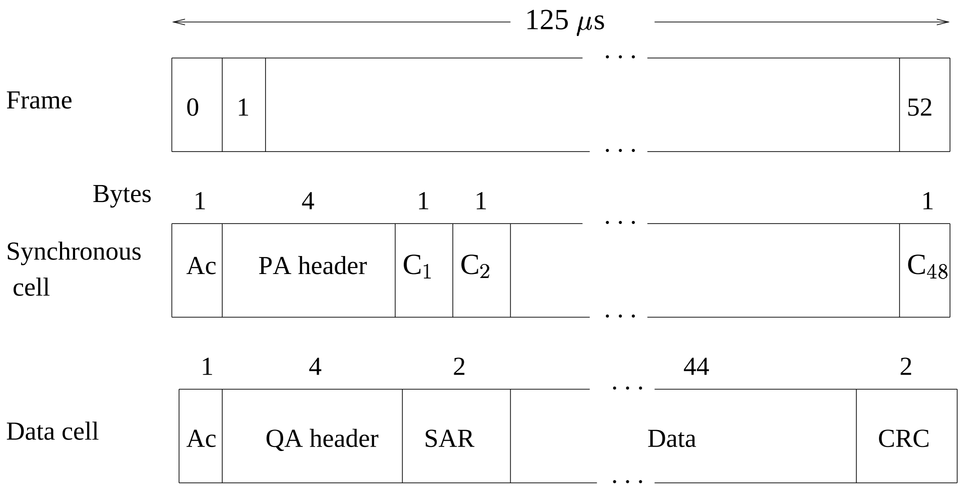


Figure B.3: DQDB frame and cell format

and now counts the number of requests made after the station became ready. To send a cell, a station must first make a reservation by setting the *Request* bit in some cell on the reverse bus. As this cell propagates down the reverse bus, every station along the way notes it and increments its *RC*. The empty cells generated by the head-end pass the stations which may occupy the cell by setting its *Busy* bit. This requires that the station is first in the FIFO queue on the bus. The queue position is represented by the value of the *CD* counter.

The DQDB protocol can achieve full channel utilization. However, a station's share of the channel bandwidth is based on its location on the channel. If an upstream station always has data ready it can occupy almost all the channel bandwidth and leave the down stream stations with very little bandwidth. Extensions of the DQDB protocol to handle this unfairness problem have been proposed.

Appendix C

WAN protocols

In this section we present WAN standards which specify protocols on different layers:

- physical and data link layer: SONET/SDH
- physical and data link layer: Frame relay
- data link layer: HDLC, PPP
- network layer: SMDS
- physical, data link and network layer: X.25, N-ISDN

C.1 SDH/SONET

The *Synchronous Digital Hierarchy* (SDH) and *Synchronous Optical Network*(SONET) are European and US standards for optical circuit-switched communication over *Time Division Multiplex* (TDM)channels [55, 68, 70]. SDH/SONET was standardized in 1989 by ITU-T. SDH/SONET is increasingly popular in Europe and US and will eventually completely replace the Plesichronous Digital Hierarchy (PDH) equipment in the telephone networks.

SDH/SONET provides multiplexing of multiple digital channels, and operation, administration and maintenance (OAM). A SDH/SONET system consists of switches, multiplexers and repeaters, all connected by optical fibers. The basic SONET frame is a block of 810 bytes put out every 125 micro seconds, resulting in 8000 frames/second. The 810-byte SONET frames are best described as rectangle of bytes, 90 columns wide by 9 rows high. A total of 8

× 810 = 6480 bits are transmitted 8000 times per second, giving a total of 51.84 Mbps. This is the basic channel and is called Synchronous Transport Signal-1 (STS-1) in SONET. The basic channel in SDH is called Synchronous Transport Module-1 (STM-1) and has a rate of 155.52 Mbps. All SDH and SONET trunks are a multiple of STM-1 and STS-1, respectively.

A few columns and rows in the SDH/SONET frame contain OAM information. This reduces the user data transmission rate to 50.112 Mbps in SONET. SDH/SONET provides multiplexing and de-multiplexing between channels with different bit rates.

C.2 Frame relay

Frame relay is a connection-oriented standard for exchange of frames over a virtual leased line [55, 68, 70]. The difference between an actual leased line and a virtual leased line is that with an actual one, the user can send traffic all day long at the maximum speed. With a virtual one, data bursts can be sent at full speed, but the long-term average usage must be below a predetermined level. In return, the carrier charges much less for a virtual line than a physical one. In addition to competing with leased lines, frame relay also competes with X.25 permanent VCs, except that it operates at higher speeds (usually 1.5 Mbps) and provides no error recovery or flow control. Studies have shown that frame relay yields a throughput an order of magnitude higher than X.25.

Frame relay has no network-layer protocol. Only the physical layer and data link layer is present. User data transfer is done using the core LAPF (Link Access Procedure for Frame-Mode Bearer Services) protocol. This protocol is similar to LAPB and HDLC except that it does not contain any control field.

Frame relay implements preventive traffic control based on Call Admission Control (CAC) and traffic enforcement (policing). The traffic descriptor is defined as a set of parameters which characterize the connection's statistical properties. It contains three parameters: *Committed Information Rate* (CIR), *Committed Burst Size* (BE), and *Excess Burst Size* (BE). CIR is the average bit rate in which the network guarantees to transfer information units over a measurement interval T. This T is defined as BC/CIR. The BC is the maximum number of bits that can be transmitted during the interval T. The BE is the maximum number of uncommitted bits that the network will attempt to carry during the interval T.

Once a connection has been established in the network, the edge node of the frame relay network must monitor the connection's traffic flow to ensure that the actual usage of the network resources does not exceed the traffic contract. Traffic enforcement allows the network to enforce the end user's information rate and discard information when the subscribed access rate is exceeded.

Bits	8	1	1	1	1	1	Variable	16	8
	Flags	C/R	DLCI	FECN	BECN	DA	Data	FCS	Flags

Figure C.1: Frame relay frame format.

The 8-bit *Flags* field is used to start and end the frame. The 1-bit *C/R* field designates whether the frame is a command or response. The *Data Link Connection Identifier* (DLCI) identifies the virtual circuit. The 1-bit *Extended Address* (EA) field indicates whether a normal or extended DLCI field is used. The 1-bit *Forward Explicit Congestion Notification*(FECN) field, 1-bit *Backward Explicit Congestion Notification* (BECN) field, and the 1-bit *Discard Eligibility* (DE) field is used for congestion control purposes. The *Data* field contains up to 16,000 bytes of data. The 16-bit *Frame Check Sequence* (FCS) field contains a CRC code.

C.3 HDLC

High-level Data Link Control (HDLC) is a relatively old data link protocol which forms the basis for standards such as LLC, LAPB, LABF and LABD [55, 68, 70]. HDLC is an OSI standard. HDLC is connection-oriented, bit-oriented and use bit stuffing for data transparency. With the use of bit stuffing, arbitrary bit patters can be inserted into the data field of the frame.

The frame begins and ends with the same bit sequence. The *Address* field is primarily of importance on line with multiple terminals, where it is used to identify one of the terminals. The *Control* field is used for sequence numbers, acknowledgments and other purposes. The *Data* field may contain arbitrary information and may be arbitrary long. The *Checksum* field protects against bit errors.

Bits	8	8	8	≥ 0	16	8
	01111110	Address	Control	Data	Checksum	01111110

Figure C.2: HDLC frame format

C.4 PPP

Point-to-Point Protocol (PPP) is a relatively new (1994) data link protocol for point-to-point lines [55, 68, 70]. PPP is defined by RFC 1661 [65] issued by IETF. Point-to-point communication is primarily used in two situations. First point to point connections exists between Internet routers connected over leased lines. Second, a host (home PC) can call an Internet Service Provider's (ISP's) router and then act like a full-blown Internet host.

PPP handles error recovery, can carry packets from multiple protocols, allows IP addresses to be negotiated at connection time and permits authentication. PPP is connection-oriented, character-oriented and uses character stuffing for data transparency. PPP frames can be sent over dial-up telephone lines, SONET/SDH or bit-oriented HDLC lines.

Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	01111110	Address	Control	Protocol	Data	Checksum	01111110

Figure C.3: PPP frame format

The frame format for PPP is similar to HDLC. The frame begins and ends with the same bit sequence. The *Address* field is set to all ones, indicating that all stations are to accept this frame. The *Control* field usually has a fixed value indicating an unnumbered frame with no sequence numbers and acknowledgments. In wireless networks, reliable transmission using numbered mode can be used. The *Protocol* field tells which type of packets is carried by the PPP frame. The *Payload* field is variable length, up to some negotiated maximum. The *Checksum* field protects against bit errors.

C.5 SMDS

Switched Multi-Megabit Data Service (SMDS) is a high-speed connectionless packet-switched WAN standard [70]. It is typically used for LAN interconnection over public WANs. SMDS can use fiber- or copper-based media; it support speeds of 34 Mbps (45 Mbps in the US) and eventually 155 Mbps. Low transit delay is a particular performance target. SMDS is defined in technical specifications developed by Bellcore in the US and has been adapted for European conditions such as European transmission speeds.

Bytes	8	8	≤ 9188
	Destination address	Source address	Data

Figure C.4: SMDS packet format

The *Source Address* and *Destination Address* are in E.164 format, which make them the same as ISDN numbers and similar to telephone numbers, and are globally unique. Groups of addresses can be defined to support multicast applications.Address screening is a useful feature which is provide filtering of packets based on their addresses. The *User Data* field can be up to 9188 bytes long.

SMDS can be implemented over different network technologies. The principal platform currently used for SMDS access is based on DQDB defined by the IEEE 802.6 MAN standard. DQDB is based on fixed length cells of 53 bytes (same length as ATM) and a variable length SMDS packet must therefore be segmented into an appropriate number of cells when carried by DQDB.

Similar to frame relay, SMDS is efficient for bursty traffic. Short bursts may be sent at maximum speed. SMDS limits the average traffic rate between source-destination host pairs.

C.6 X.25

X.25 emerged in the late 70s as a connection-oriented packet-switched WAN standard [55, 68, 70]. X.25 was designed for high error rate links.

X.25 network devices fall into three general categories: *Data Terminal Equipment* (DTE), *Data Circuit-Terminating Equipment* (DCE), and *Packet Switching Exchange* (PSE). DTE devices are end systems that communicate across the X.25 network. DCE devices are communication devices such as modems and packet switches, that provide the interface between DTE located on the premises of individual subscribers. PSE are switches that compose the bulk of the carriers network.

The standard covers three levels of protocols:

- Physical level
- Link level
- Packet level

The physical level deals with the physical interface between an attached station (computer, terminal) and the link that attaches that station to the packet-switching node. The physical-layer standards includes X.21 and EIA-232.

The link level provides for the reliable transfer of data across the physical link.The link-level standard is referred to as LAPB (Link Access Control Protocol Balanced). LAPB is a subset of HDLC. Hop-by-hop flow control, error detection and error recovery are three important functions of LAPB.

The packet level provides a VC service at 64 kbps. The VCs is either permanent (fixed) or switched (on-demand call set up). The packet-level provides flow control, error recovery, bit padding and segmentation and reassembly on the network level.

4 bits	12 bits	8 bits	≤ 4093 bytes
GFI	LCI	PTI	Data

Figure C.5: X.25 packet format

The *General Format Identifier* (GFI) field identifies packet parameters, such as whether the packet carries user data or control information, what kind of windowing is being used, and whether deliver confirmation is required. The *Logical Channel Identifier* (LCI) field identifies the VC across the local DTE/DCE interface. The *Packet Type Identifier* (PTI) field identifies the packet as one of 17 different packet types. The *User Data* field contains encapsulated upper-layer information. The User data field can contain between 61 and 4093 bytes.

C.7 N-ISDN

Narrow-band Integrated Services Digital Network (N-ISDN) is digital circuit-switched system for support of voice and nonvoice services. [55, 68, 70].

N-ISDN was proposed by ITU-T in 1984. Users access N-ISDN over different channels, where the 64 kbps B channel and the 16 kbps D channel are most well known. The B channel is the basic user channel and is e.g. used for voice, video or data. Four kinds of service is possible over a B channel:

- circuit-switched service
- X.25 data service
- frame relay service
- leased line service

The D channel is primary used for out-of-band signaling for call set up and release purposes. All traffic over the D channel employs a link-layer protocol known as LAPD (Link Access Protocol D channel). The LAPD standard provides two forms of service to LAPD users: the unacknowledged information service and the acknowledged information service. The latter service is similar to the service offered by LAPB and HDLC.

Basic access consists of two full-duplex 64 kbps B channels and one full-duplex 16 kbps D channel. The total bit rate is 144 kbps. The basic access is intended to meet the needs of most residential users and small companies. Operators today also sell Internet access over the N-ISDN basic access.

Primary access consists of the mix 23B+1D in US and 30B+1D in Europe. The total bit rate is 1488 kbps in US and 1936 kbps in Europe. The primary access is intended for users with greater capacity requirements, such as offices with a *Digital Private Branch Exchange* (PBX).

Appendix D

Wired access networks

D.1 Twisted pair

Over 400 miljon subscribers are connected to the Public Switched Telephone Network (PSTN) via twisted pair copper wires [29]. The twisted pair runs from the subscriber to the central telephone office. A modem (modulator/demodulator) is used to convert the digital signal from the host computer to analog signals suitable for twisted pair transmission. At the central office the analog signal is converted into a digital signal and sent to the central office which connects the Internet Service Provider (ISP). The connection from the last central office to the ISP is normally digital. (PCM).

D.1.1 Analog modem

The fastest analog modems today are based on the ITU standards v.90 and v.92. The v.90 standard has a downlink which runs at 56 kbps, and an uplink which runs at 33.5 kbps. The downlink analog signal is created as follows. The local telephone exchange converts the digital signal to an analog signal using Pulse Amplitude Modulation (PAM). An amplitude level corresponds to a symbol representing up to 8 bits. In v.90 only 7 bits/symbol is used, resulting in a theoretical downlink rate of 56 kbps. However, due to imperfect lines and restrictions on high power levels on phone lines even fewer symbols are used. In practice, the downlink speed is 53.3 kbps at best. The uplink analog signal in v.90 is obtained by phase-amplitude modulation known as QAM (Quadrature Amplitude Modulation). The more recent v.92 standard gives a downlink which runs at 56 kbps (actually 53.3 kbps), and an uplink which runs at 48 kbps. The v.92 standard use PAM, instead of QAM, to encode the analog signal for the uplink. Due to problems with crosstalk (interference) between nearby copper wires the uplink in v.92 uses fewer number of amplitude levels (symbols) than the downlink.

D.1.2 ISDL

ISDN DSL (ISDL) offers a basic rate (128 kbps) service that is interoperable with ISDN terminal adapters and routers. ISDL is a data-only service suitable for remote LAN access, Internet access and videoconferencing.

D.1.3 HDSL

High-data rate DSL (HDSL) comes in two variants. In US it runs at T1 speeds, i.e. at 1.544 Mbps over two copper pairs. In Europe it runs at E1 speeds, i.e. 2.048 Mbps over three copper pairs. The maximum distance of the twisted pair connecting the HDSL modem and the central office is 5500 m in case of T1 speed, and 4900 m in case of E1 speed. The HDSL connection is symmetric, i.e. the bit rates of the uplink and downlink HDSL are the same. HDSL gives the same performance as previous systems for T1/E1 but over a four as large distance. HDSL does not require repeaters on the lines as old-style T1/E1 implementations do. Due for the need of several cooper pairs, HDSL is primarily aimed at the business market.

D.1.4 SDSL

Similar to HDSL, Single-line DSL (SDSL) delivers the same T1 or E1 speeds, but it does it on a single set of twisted pair. The distance limit is 3000 m. A single line can in most cases support a voice channel and the T1/E1 uplink/downlink channels simultaneously.

D.1.5 ADSL

ADSL employs advanced modulation techniques to take advantage of a frequency spectrum that is not utilized by telephone traffic. Standard voice calls utilize the spectrum between 0-4 KHz, while ADSL utilizes frequencies between 26 kHz and 1 MHz. Attenuation, which increases with line length and frequency, dominates the constraints on data rate over the twisted pair wire.

ADSL provides up to 8 Mbps downstream and up to 1 Mbps upstream in conjunction with basic telephony service on the same line. The maximum distance between the customer's ADSL modem and the central office is between 2700 m and 5500 m, depending on the link speed. Of all the xDSL variants, ADSL has received the most attention from the industry primarily because it is ideally suited to Internet traffic. In the majority of Web applications, bandwidth requirements are highly asymmetric, with uplink (from the user to the ISP) approximately 10% of downlink bandwidth needs.

The available bandwidth can be divided into multiple channels in two ways: Frequency Division Multiplexing (FDM) and Echo Cancellation with Hybrid (ECH). In FDM different portions of the frequency spectrum are used for the uplink and downlink channel. In ECH, the uplink and downlink channels has overlapping frequency spectra. Due to its larger bandwidth the downlink channel also has its own portion of the spectrum. The separation between the uplink and downlink channels are done by local echo cancellation. Since ECH use lower frequencies than FDM the risk of radio interference is lower.

There are two common line encodings for ADSL: Carrierless Amplitude and Phase (CAP) and Discrete Multitone (DMT). DMT has won the standards battle and is now the standard and the more common of the two. DMT divides the available bandwidth into a number of sub channels. Each sub channel has a bandwidth of 4 kHz. The downlink frequency band is divided into 256 sub channels, and the uplink band is divided into 32 sub channels. Phase-amplitude modulation (QAM) with a maximum of 8 bits/symbol is used to transmitt data over each sub channel.

D.1.6 RADSL

RADSL is derived from ADSL technologies with some added features. RADSL automatically adjust the line speed to the gauge of the wire, the distance between the subscriber and the central telephone office, and condition of the line.

D.1.7 VDSL

Very High Rate DSL (VDSL) is a high-speed asymmetric DSL technology. Because higher speeds requires shorter cable lengths, VDSL is most attractive in close proximity to the switching equipment – typically residing in the same building. Depending on cable distance, VDSL will provide between 13 to 52 Mbps on the downlink and 1.5 to 2.3 Mbps on the uplink. The maximum distance of the twisted pair connecting the VDSL modem and the central office is between 300 m and 1400 m.

D.2 Wired LANs and MANs

Access is of course possible via LANs and MANs which are connected to the Internet core network. Different LAN and MAN technologies are described in Appendix A and B.

D.3 Fiber

Fiber access networks are often called Fiber-To-The-X networks, where X can be Home, Curb, Building, or Neighborhood among others [29]. The outside world e.g. Internet or satellite is connected to a Central Office (CO). Below we discuss two techniques for connecting the subscribers to the CO: active optical networks and passive optical networks.

D.3.1 Active optical network

An active optical network architecture can have the form of a star or ring, see Figure D.1 and D.2. The network is called active since it contains active units, switches and add-drop multiplexers (ADMs), that need to be powered. In the star topology, the Optical Line Termination (OLT) at the CO is connected to the remote node (switch) via a fiber. The remote directs the traffic to different Optical Network Units (ONUs), situated at the subscribers' curb or building. In the ring network topology, the OLT and the ONUs are connected to each other in a SONET/SDH ring via Add-Drop Multiplexors (ADMs). In both the star and ring topology, the Network Termination (NT) unit at the subscriber home is connected via xDSL to the ONU. Note that the capacity of the feeding fiber is shared between multiple endpoints.

D.3.2 Passive optical network

In the passive network, the network does not have any active electronics (and hence does not need any powering arrangements). An passive optical network architecture can have

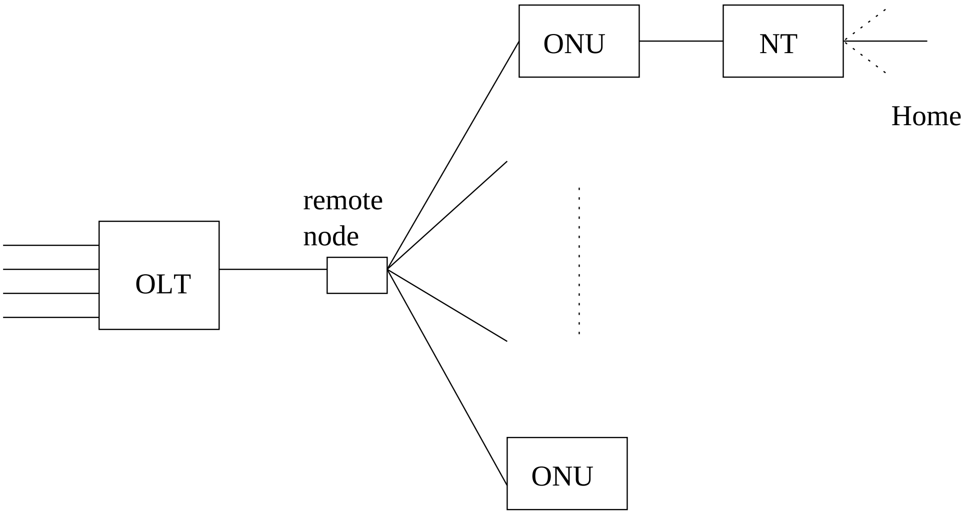


Figure D.1: Active optical network:star topology

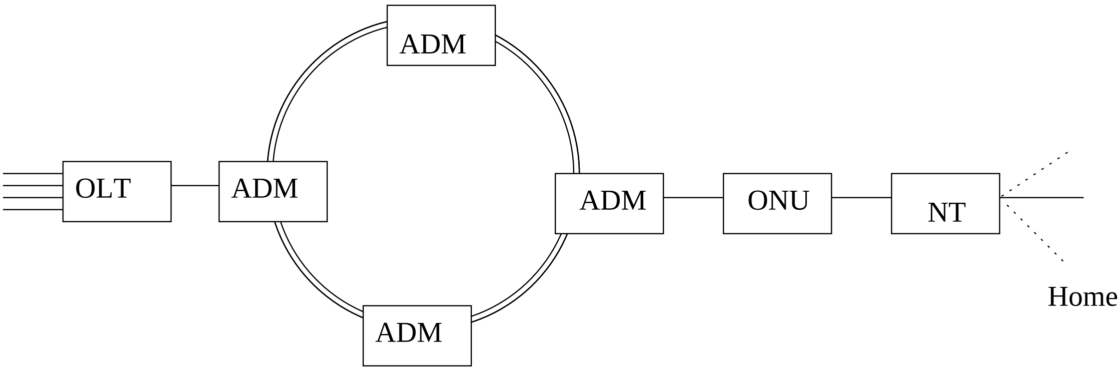


Figure D.2: Active optical network: ring topology

the form of a star or ring, see Figure D.3 and D.4. At the remote node, a passive splitter replicates the downstream optical signal from the feeder fiber onto the individual distribution fibers which terminates at an ONU. A coupler combines signals from individual homes onto the feeder fiber using Optical Time Division Multiplexing. As with the active network, the feeder capacity is shared between multiple endpoints. Passive optical network standards are under development for both ATM (aPON) and Ethernet (ePON) by the ITU-T and the IEEE 802.3 working group, respectively.

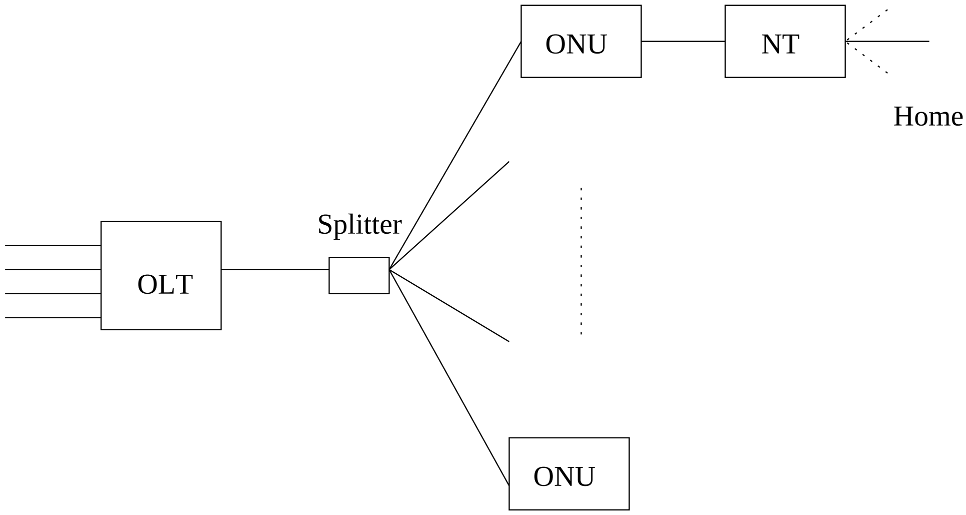


Figure D.3: Passive optical network: star topology

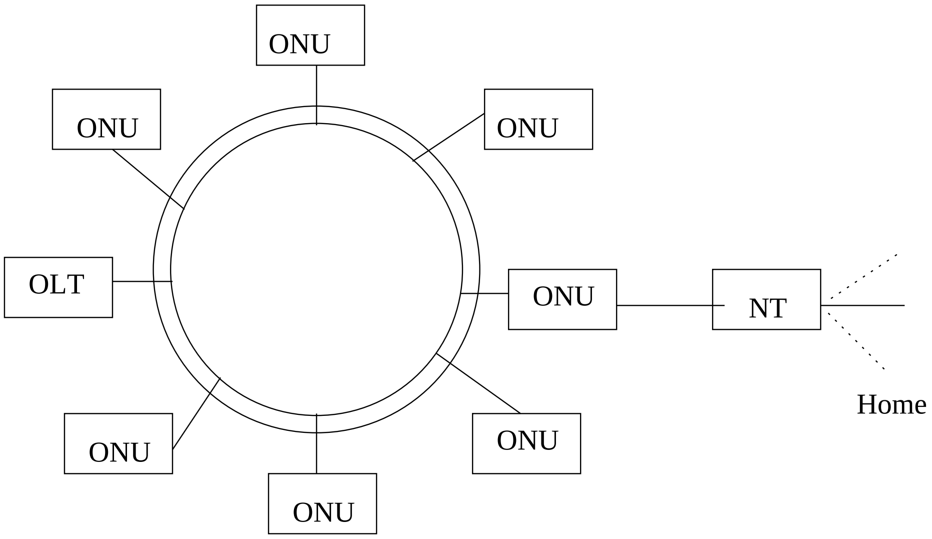


Figure D.4: Passive optical network: ring topology

D.4 Hybrid Fiber Coax

The topology of the Hybrid-Fiber-Coax network is shown in Figure D.5 [29].

The access networks extends from the headend via fibers to fiber nodes, which are connected to coax distribution networks. The fiber node performs optical to electrical signal conversion and vice versa. In a two-way HFC system the coax distribution network has bidirectional amplifiers. Each fiber node can serve 500-2000 subscribers. Each subscriber has Coaxial Terminal Unit (CTU) which connects to the coax distribution network via a tap. The most familiar CTUs are the cable modem and the set top box.

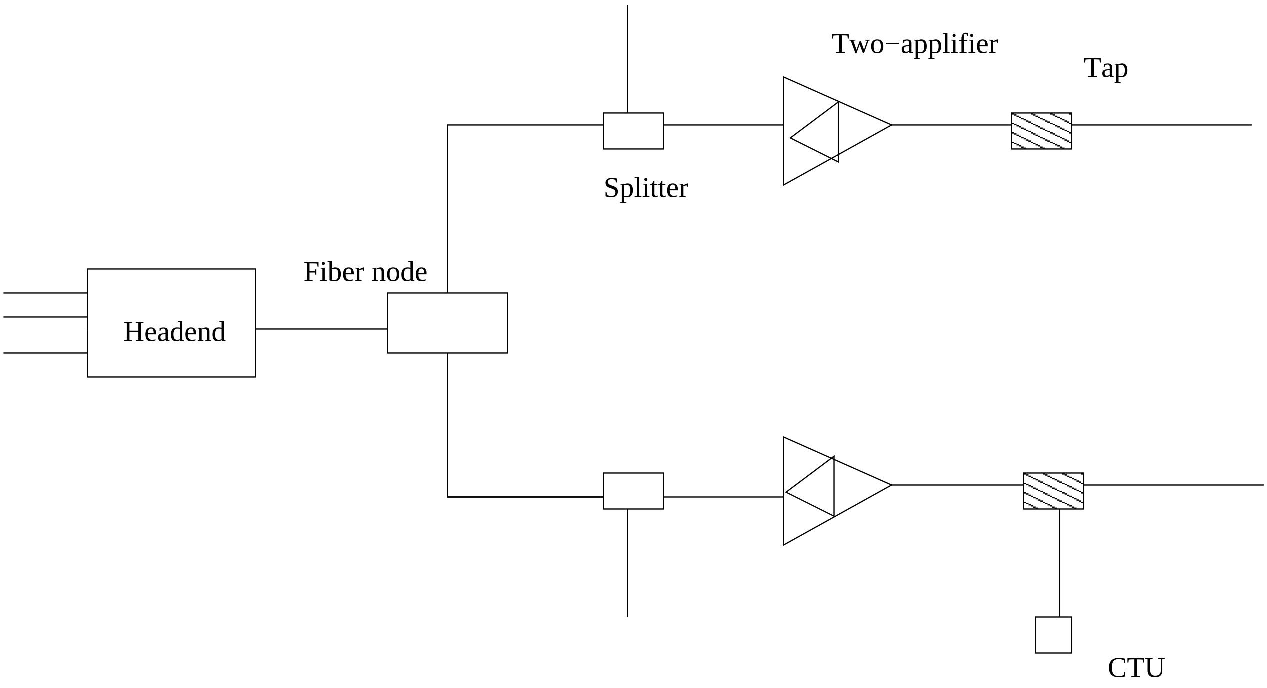


Figure D.5: Hybrid Fiber Coax network

The HFC network has the following features:

- Point-to-multipoint downstream and multipoint-to-point upstream. Collisions can only occur on the uppstream channel, which needs an efficient scheme to avoid and resolve collisions.

- Inability to detect collisions by stations. Thus, stations rely on the head end to notify them of the results of upstream transmissions.

- Large propagation delay. The maximum round-trip-delay (RTD) is significantly longer than that of Ethernet. Neutralizing the effect of propagation delay is of synchronization concern so that the transmissions from stations arrive at the right time slots assigned by head end. Consequently, the MAC protocol should have a ranging protocol to measure the propagation delay for each station.

- Asymmetric data rate on upstream and downstream channel.

- Non-uniform user distribution. Most subscribers are distributed over the last few miles of the network.

The IEEE 802.14 and MCNS (Multimedia Cable Network System) are the groups which are working on establishing standard for data transmission over HFC systems. The groups are working with the physical and Media Access Control (MAC) layers. The IEEE 802.14 has issued a draft standard, and the MCNS standard was approved by the ITU-T in 1998. A summary of the two standards now follows.

Both standards have similar key features on the physical layer. FDMA is used in the downstream and upstream channel. Each FDMA channel is further slotted by TDMA. Separate spectra is allocated for the downstream and upstream channels. The downstream channel is further divided into multiple subchannels each with 6 MHz. Digital signal modulation is based on QAM or quadrature phase-shift keying (QPSK).

Both the 802.14 and MCNS standard model the upstream channel as a stream of minislots. The headend must coordinate accesses to this shared bandwidth since stations cannot listen to the upstream channel. The headend assigns the usage of upstream bandwidth and describes this assignment in the *bandwidth allocation map*. Once the map is sent over the downstream channel, stations can learn the assignment from the map and proceed accordingly. Basically, some of the upstream minislots are assigned as *request minislots,* each of which can accommodate a request PDU. The other minislots are *data minislots* where a data PDU may occupy multiple contiguous minislots.

The 802.14 standard attempts to provide complete support for ATM, thus making MAC-CS layer and the ATM layer necessary. The MAC-CS transforms data passing through the LLC SAP into ATM PDUs for transmission over the network. The MCNS standard support IP at the network layer and Ethernet at the data link layer.

Both the 802.14 and MCNS standard support real-time and non-real-time service classes.

Appendix E

Wireless access networks

E.1 Wireless LAN

E.1.1 IEEE 802.11

Network topology

The basic topology of an 802.11 network is shown in Figure E.1. A Basic Service Set (BSS) consists of two or more wireless nodes, or stations (STAs), which have recognized each other and have established communications [62, 69]. In the most basic form, stations communicate directly with each other on a peer-to-peer level sharing a given cell coverage area. This type of network is often formed on a temporary basis, and is commonly referred to as an *ad hoc* network, or Independent Basic Service Set (IBSS).

In most instances, the BSS contains an Access Point (AP). The main function of an AP is to form a bridge between wireless and wired LANs. The AP is analogous to a basestation used in cellular phone networks. When an AP is present, stations do not communicate on a peer-to-peer basis. All communications between stations or between a station and a wired network client go through the AP. APs are not mobile, and form part of the wired network infrastructure. A BSS in this configuration is said to be operating in the infrastructure mode.

The Extended Service Set (ESS) consists of a series of overlapping BSSs (each containing an AP) connected together by means of a Distribution System (DS). Although the DS could be any type of network, it is almost invariably an Ethernet LAN. Mobile nodes can roam between APs and seamless campus-wide coverage is possible.

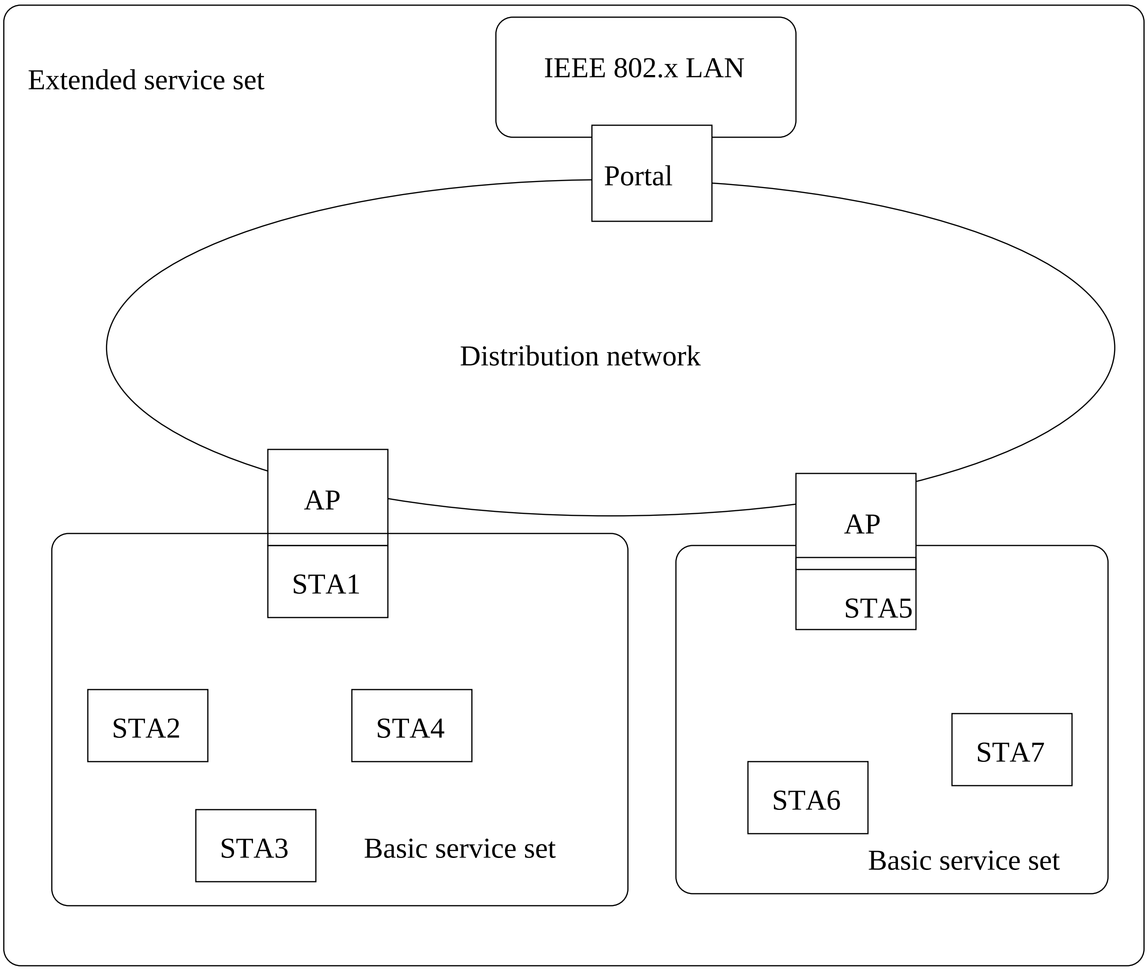


Figure E.1: IEEE 802.11 architecture

Physical layer

The original IEEE 802.11 specifies three physical layers:

- Direct Sequence Spread Spectrum (DSSS) operating in the 2.4 GHz unlicensed ISM band, at data rates of 1 Mbps and 2 Mbps.
- Frequency Hopping Spread Spectrum (FHSS) operating in the 2.4 GHz ISM band, at data rates of 1 Mbps and 2 Mbps.
- Infrared (IR) at 1 Mbps and 2 Mbps operating at a wavelenght between 850 nm and 950 nm.

Two additional physical layers are defined in IEEE 802.11a and 802.11b:

- IEEE 802.11a uses Orthogonal Frequency Division Multiplexing (OFDM) operating in the 5 GHz ISM band, at data rates up to 54 Mbps.
- IEEE 802.11b uses High Rate DSSS (HR/DSSS) scheme, operating in the 2.4 GHz ISM band, at data rates of 5.5 Mbps and 11 Mbps.

Currently, binary phase-shift keying (BPSK) and quadrature phase-shift keying (QPSK) modulation schemes are used in DSSS WLAN systems. Gaussian shaped frequency shift keying (GFSK) modulation scheme is used in FHSS WLAN systems. They are sufficient in 1 and 2 Mbps systems, but they do nor meet the demands of higher data rate transmissions schemes. To achieve higher speeds, different techniques are considered by the 802.11 committee.

Medium access control layer

The basic access method for 802.11 is the Distributed Coordination Function (DCF) which uses Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). This requires each station to listen for other users. If the channel is idle, the station may transmit. However if it is busy, each station waits until transmission stops, and then enters into a random back off procedure. This prevents multiple stations from seizing the medium immediately after completion of the preceding transmission.

Packet reception in DCF requires acknowledgment as shown in Figure E.2. The period between completion of packet transmission and start of the ACK frame is one Short Inter Frame Space (SIFS). ACK frames have a higher priority than other traffic. Fast acknowledgment is one of the salient features of the 802.11 standard, because it requires ACKs to be handled at the MAC sublayer. Transmissions other than ACKs must wait at least one DCF inter frame space (DIFS) before transmitting data. If a transmitter senses a busy medium, it determines a random back-off period by setting an internal timer to an integer number of slot times. Upon expiration of a DIFS, the timer begins to decrement. If the timer reaches zero, the station may begin transmission. However, if the channel is seized by another station before the timer reaches zero, the timer setting is retained at the decremented value for subsequent transmission. The method described above relies on the *Physical Carrier Sense*. The underlying assumption is that every station can hear all other stations. This is not always the case.

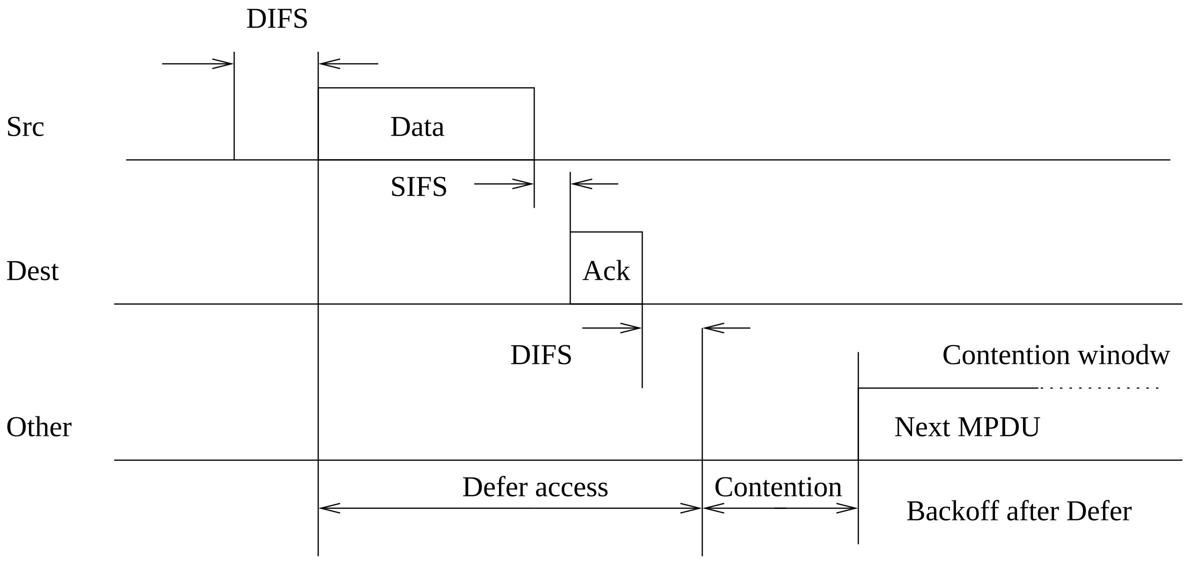


Figure E.2: CSMA/CA backoff algorithm

Referring to Figure E.3, the AP is within range of the STA-A, but STA-B is out of range. STA-B would not be able to detect transmissions from STA-A, and the probability of collision is greatly increased. This is known as the *Hidden Node problem*.

To combat this problem, a second carrier sense mechanism is available. *Virtual Carrier Sense* enables a station to reserve the medium for a specified period of time through the use of RTS/CTS frames. In the case described above, STA-A sends an RTS frame to the AP.

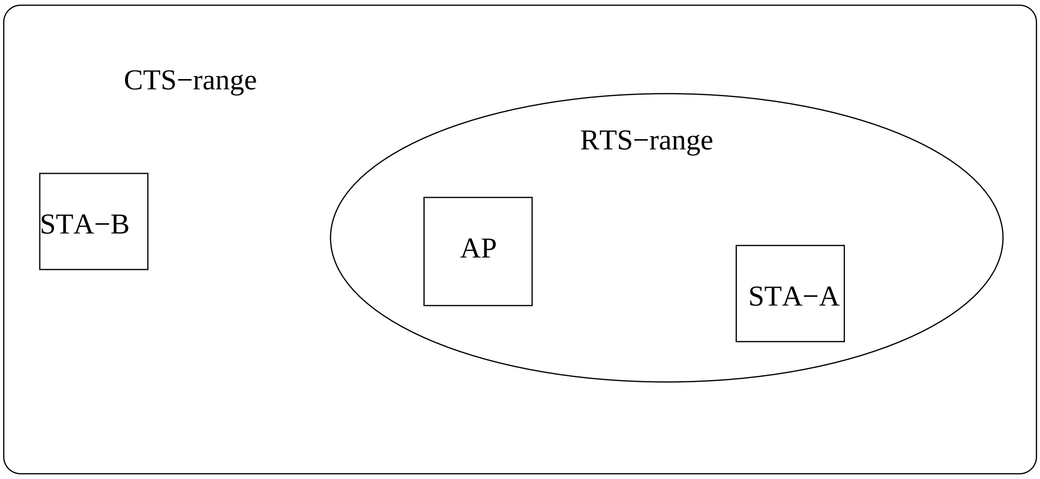


Figure E.3: Hidden node problem solved by RTS/CTS messages

The RTS will not be heard by STA-B. The RTS frame contains a duration/ID field which specifies the period of time for which the medium is reserved for a subsequent transmission. The reservation information is stored in the Network Allocation Vector (NAV) of all stations detecting the RTS frame. Upon receipt of the RTS, the AP responds with a CTS frame, which also contains a duration/ID field specifying the period of time for which the medium is reserved. While STA-B did not detect the RTS, it will detect the CTS and update its NAV accordingly. Thus, collision is avoided even though some nodes are hidden from other stations. The RTS/CTS procedure is invoked according to a user specified parameter. It can be used always, never, or for packets which exceed an arbitrarily defined length.

As mentioned above, DCF is the basic media access control method for 802.11 and it is mandatory for all stations. The Point Coordination Function (PCF) is an optional extension to DCF. PCF provides a time division duplexing capability to accommodate time bounded, connection-oriented services such as cordless telephony.

E.1.2 Bluetooth

Bluetooth aims at so-called ad hoc piconets, which are LANs with a very limited coverage and without the need for an infrastructure [62, 69]. This different type of network is needed in order to connect different small devices in close proximity without expensive wiring or the need for a wireless infrastructure. After some studies by Ericsson in 1994, five companies (Ericsson, Intel, IBM, Nokia, Toshiba) founded the Bluetooth consortium in 1998. Bluetooth represents a single-chip, low-cost, radio-based wireless network technology. Since then many other companies and research institutions have joined the Bluetooth special interest group. Bluetooth has not been adopted by any standardization organization but is a de facto standard.

Physical layer

The design of bluetooth's physical layer is constrained by the need for low power consumption and use of a frequency which is available worldwide. Bluetooth uses the license-free ISM frequency band at 2.4 GHz. A frequency-hopping/time-division duplex scheme is used for transmission with a hopping rate of 1,600 hops per second. The time between two hops is called a slot, which is an interval of 625 μs, thus, each slot uses a different frequency. By countries where the available bandwidth is at least 80 MHz (US, most parts of Europe) Bluetooth uses 79 hop carriers equally spaced with 1 MHz. In Japan, France, and Spain, national frequency restrictions only permit 23 hop carriers. On average, the frequency-hopping sequence 'visits' each hop carrier with an equal probability. All devices using the same hopping sequence with the same phase form a bluetooth *piconet*.

With a transmitting power of up to 100 mW, Bluetooth devices have a range of up to 10 m (or even 100 m with special transceivers). Having this power and relying on battery power, a device cannot be in an active transmit mode all the time. Therefore, Bluetooth defines several low-power states for the device.

Connections (and thus piconets) can be initiated by any device which then becomes the master. This is done via sending *page* messages if the device already knows the address of the receiver, or *inquiry* messages followed by a page message if the receiver's address is unknown.

Medium access control

Several mechanisms control medium access control is a Bluetooth system. First of all, one device within a piconet acts as a master, other devices (up to seven) act as slaves. The master determines the hopping sequence using its unique identifier as well as the phase of the sequence using its internal hardware clock. This unique setting of master parameters prevents two different piconets from having the same hopping sequence and thus separates them via CDMA. Within a piconet, the master controls medium access using a polling and reservation scheme.

All Bluetooth devices have the same networking capabilities, i.e. they can be master or slave. There is no distinction between terminals and base stations, any two or more devices can form a piconet. The unit establishing the piconet automatically becomes the master, all other devices will be slaves. Within a piconet only one master can exist at any given time.

Bluetooth offers two different types of services:

- **Synchronous connection-oriented link (SCO)**: Standard telephone (voice) connections require symmetrical, circuit-switched, point-to-point connections. For this type of link, the master reserves two consecutive slots (forward and return slots) at fixed intervals.

- **Asynchronous connectionless link (ACL)**: Typical data applications require symmetrical or asymmetrical, packet-switched, point-to-multipoint transfer scenarios. Here the master uses a polling scheme.

Bluetooth can either support a single ACL, three SCOs, or an ACL and a SCO at the same time. SCOs always supports 64 kbps synchronous connections. ACL can support different bit rates depending on the packet types. Data rates are up to 432.6 kbps on a symmetric link using five consecutive slots and unprotected data. Asymmetric links can carry 721.0 kbps in one direction and 57.6 kbps in the other direction, also using five consecutive slots and no data protection.

Using a SCO link, three different types of single-slot packets can be used. Each SCO link carries voice at 64 kbps, addionally no FEC, 2/3 FEC, or 1/3 FEC. The 1/3 FEC triples the amount of data. Voice data over a SCO is never retransmitted.

For ACLs carrying data, 1-slot, 3-slot or 5-slot packets can be used. Additionally, data can be protected using a 2/3 FEC scheme. Bluetooth also offers a fast BEC scheme for reliable transmission. Each packet is acknowledged in the slot following the packet. If a packet is lost, a sender can retransmit it immediately in the next slot after negative acknowledgement.

Networking

All users within one piconet have the same hopping sequence and, thus, share the same 1 MHz channel. As more users join the piconet, the throughput per user drops quickly. Having only one piconet available within the 80 MHz in total is not very efficient. This led to the idea of forming groups of piconets called *scatternets*. Only those units that really have to exchange data share the same piconet, so that many piconets with overlapping coverage can exist simultaneously.

If a device wants to participate in more than just one piconet, it has to synchronize to the hopping sequence of the piconet it wants to take part in. If a device acts as slave in one piconet, it simply starts synchronize with the hopping sequence of the piconet it wants to join. After synchronization, it acts as a slave in this piconet and does not participate in its former piconet any longer. To enable synchronization, a slave has to know the identity of the master determining the hopping sequence. Before leaving a piconet, a slave informs the current master that it will be unavailable for a certain amount of time. The remaining devices in the piconet continue communication as usual.

A master can also leave its piconet and act as a slave in another piconet. It is clearly not possible for a master of one piconet to act as the master of another piconet for this would lead to identical behavior of those two piconets (they both would have the same hopping sequence).

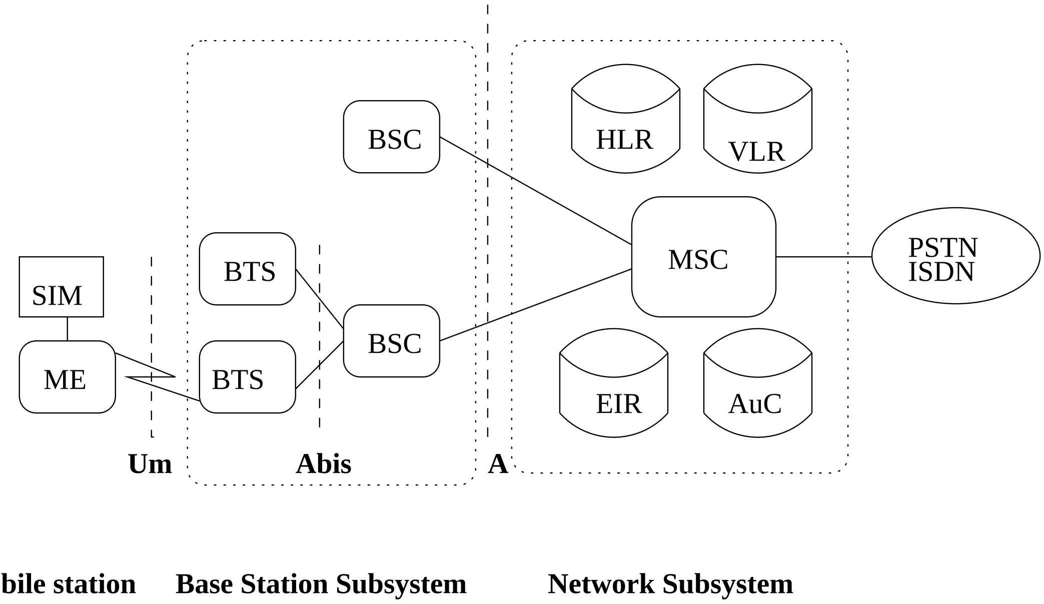
Communication between different piconets thus takes place through devices jumping back and forth between these nets. If this is done periodically, for instance, isochronous data streams can be forwarded from one piconet to another.

E.2 Mobile systems

E.2.1 GSM

The Global Systems for Mobile communications (GSM) is system for exchange of digital voice and data signals between a set of terminals having at least one mobile member [62, 69]. GSM is an European standard from the early 90s developed by ETSI. Today, GSM has 40% market share of the world cellular market. The analog AMPS system still holds 30 % , whereas the Japanese PDC holds 15 %. Standard GSM provides voice and data transfer over circuit-switched network connections at 9.6 kbps.

A GSM network is composed of several functional entities, whose functions and interfaces are specified. Figure E.4 shows the layout of a generic GSM network. The GSM network can be divided into three broad sections. The Mobile Station is carried by the subscriber. The Base Station Subsystem controls the radio link with the Mobile Station. The Network Subsystem, the main part of which is the Mobile services Switching Center (MSC), performs the switching of calls between the mobile and other fixed or mobile network users, as well as handling mobility management. Not shown is the Operation and Maintenance Center, which oversees the proper operation and setup of the network. The Mobile Station and the Base Station Subsystem (BSS) communicate across the Um interface, also known as the air interface or radio link. The BSS communicates with the MSC across the A interface.



SIM: Subscriber Identity Module	BSC: Base Station Controller	MSC: Mobile services Switching Center
ME: Mobile Equipment	HLR: Home Location Register	EIR: Equipment Identity Register
BTS: Base Transceiver Station	VLR: Visitor Location Register	AuC: Authentification Center

Figure E.4: Architecture of a GSM network

Mobile Station

The Mobile Station (MS) consists of the mobile equipment (the terminal) and a smart card called the Subscriber Identity Module (SIM). The SIM provides personal mobility, allowing the user to have access to subscribed services irrespective of a specific terminal. By inserting the SIM card into another GSM terminal, the user is able to receive calls at that terminal, make calls from that terminal and receive other subscribed services.

The mobile equipment is uniquely identified by the International Mobile Equipment Identity (IMEI). The SIM card contains the International Mobile Subscriber Identity (IMSI) used to identify the subscriber to the system, a secret athentification, and other information. The IMEI and the IMSO are independent, thereby allowing personal mobility. The SIM card may be protected against unauthorized use by a password or personal identity number.

Base Station Subsystem

The Base Station Subsystem is composed of two parts, the Base Transceiver Station (BTS) and the Base Station Controller (BCS).These communicates across the standardized Abis interface, allowing operation between components made by different suppliers.

The BTS houses the radio transceivers that define a cell and handles the radio (Um) interface protocols with the mobile station. Due to the potentially large number of BTSs, the requirements for a BTS are ruggedness, reliability, portability, and minimum cost.

The Base Station Controller (BSC) manages the radio resources for one or more BTSs, across the Abis interface. It manages the radio interface channels (setup, teardown, frequency hopping, etc.) as well as handovers. The BSC is the connection between the mobile station and the MSC.

Network Subsystem

The central component of the Network Subsystem is the MSC. It acts like a normal switching node of the PSTN or ISDN, and in addition provides all the functionality needed to handle a mobile subscriber, including registration, authentication, location updating, inter-MSC handovers, and call routing to a roaming subscriber. These services are provided in conjunction with four intelligent databases, which together with the MSC form the Network Subsystem. The MSC also provides the connection to the public fixed networks.

The Home Location Register (HLR) and the Visitor Location Register (VLR), together with the MSC, provide call routing and roaming capabilities of GSM. The HLR is a database that contains all the administrative information of each subscriber registered in the corresponding GSM network, along with the current location of the subscriber. The location assists in routing incoming calls to the mobile, and is typically the SS7 address of the visited MSC. There is logically one HLR per GSM network, although it may be implemented as a distributed database.

The VLR contains selected administrative information from the HLR, necessary for call control and provision of the subscribed services, for each mobile currently located in the geographical area controlled by the VLR. Although the VLR can be implemented as an independent unit, to date all manufacturers of switching equipment implement the VLR together with the MSC, so that the geographical area controlled by the MSC corresponds to that controlled by the VLR. The proximity of the VLR information to the MSC speeds up access to information that the MSC requires during a call.

The other two registers are used for authentication and security purposes. The Equipment Identity Register (EIR) is a database that contains a list of all valid mobile equipment on the network, where each mobile equipment is identified by its International Mobile Equipment Identity (IMEI). An IMEI is marked as invalid if it has been reported stolen or is not type approved. The Authentication Center (AuC) is a protected database that stores a copy of the secret key stored in each subscriber's SIM card, used for authentication and ciphering on the radio channel.

Multiple access

The radio spectrum in the bands 890-915 MHz for the uplink (mobile station to base station) and 935-960 MHz for the downlink has been reserved in Europe for mobile networks. One or more carrier frequencies are assigned to individual base stations, and each carrier is divided into eight time slots using TDMA. Groups of eight consecutive time slots form TDMA frames, with a duration of 4.615 ms. A transmission channel occupies one time slot position within a TDMA frame. TDMA frames of a particular carrier frequency are numbered, and both the mobile station and the base station are synchronized on this number. Larger frames are formed from groups of 26 and 51 TDMA frames (there are also larger groups), and position within such frames defines the type and function of a channel.

Traffic channels

Dedicated, or traffic, channels provide a bi-directional point-to-point transmission link to a mobile subscriber. Full-rate Traffic Channels (TCH/F) and half-rate Traffic Channels (TCH/H) are allocated together with a low bit-rate Slow Associated Control Channel (SACCH), which typically transmits measurements needed for handover decisions. There are also eighth-rate Traffic Channels, also called Stand-alone Dedicated Control Channels (SDCCH), which are used primarily for transmitting location updating information. In addition, a TCH slot can be pre-empted for signalling, in which case it is called a Fast Associated Control Channel (FACCH), which can be either full-rate or half-rate. TCHs are defined within a 26-frame multiframe.

Common channels

Common channels can be accessed both by idle mode mobiles, in order to change to dedicated mode, and by dedicated mode mobiles, to monitor surrounding base stations for handover information. The common channels, which are defined within a 51-frame multiframe,include:

- **Broadcast Control Channel (BCCH)**: Continually broadcasts, on the downlink, information including base station identity, frequency allocations, and frequency-hopping sequences.

- **Frequency Correction Channel (FCCH)** and **Synchronisation Channel (SCH)**: Used to synchronize the mobile to the time slot structure of a cell by defining the beginning of a TDMA frame.

- **Random Access Channel (RACH)**: Slotted Aloha channel used by the mobile to request access to the network.

- **Paging Channel (PCH)**: Used to alert the mobile station of incoming call.

- **Access Grant Channel (AGCH)**: Used to allocate an SDCCH to a mobile for signaling (in order to obtain a dedicated channel), following a request on the RACH.

Network aspects

Radio transmission forms the lowest functional layer in GSM. In any telecommunication system, signaling is required to coordinate the necessarily distributed functional entities of the network. The transfer of signaling information in GSM follows the layered OSI model. On top of the physical layer described above is the data link layer providing error-free transmission between adjacent entities, based on the ISDN's LAPD protocol for the Um and Abis interfaces, and on SS7's Message Transfer Protocol (MTP) for the other interfaces. The functional layers above the data link layer are responsible for Radio Resource management (RR), Mobility Management (MM) and Call Management (CM).

The RR functional layer is responsible for providing a reliable radio link between the mobile station and the network infrastructure. This includes the establishment and allocation of radio channels on the Um interface, as well as the establishment of A interface links to the MSC. The handover procedures, an essential element of cellular systems, is managed at this layer, which involves the mobile station, the base station subsystem, and, to a lesser degree, the MSC. Several protocols are used between the different network elements to provide RR functionality.

The MM functional layer assumes a reliable RR-connection, and is responsible for location management and security. Location management involves the procedures and signalling for location updating, so that the mobile's current location is stored at the HLR, allowing incoming calls to be properly routed. Security involves the authentication of the mobile, to prevent unauthorized access to the network, as well as the encryption of all radio link traffic. The protocols in the MM layer involve the SIM, MSC, VLR, and the HLR, as well as the AuC (which is closely tied with the HLR). The machines in the network subsystem exchange signalling information through the Mobile Application Part (MAP), which is built on top of SS7.

The CM functional layer is divided into three sub layers. The Call Control (CC) sub layer manages call routing, establishment, maintenance, and release, and is closely related to ISDN call control. The idea is for CC to be as independent as possible from the underlying specifics of the mobile network. Another sub layer is Supplementary Services, which manages the implementation of the various supplementary services, and also allows users to access and modify their service subscription. The final sub layer is the Short Message Service layer, which handles the routing and delivery of short messages, both from and to the mobile subscriber.

HSCSD

The High Speed Circuit Switched Data (HSCSD) is an enhanced GSM data service only requiring software upgrades in the MS and MSC. The idea with HSCSD is to split a traffic stream into several streams, using separate traffic channels for each stream, and combine these streams again. In theory, an MS could use all eight slots within a TDMA frame to achieve an air interface user rate (AIUR) of 115.2 kbps.

HSCSD exhibits some major disadvantages. It still uses the connection- oriented mechanisms of GSM which are not at all efficient for computer data traffic, which typically is bursty. The channels in HSCSD is allocated statically, i.e. statistical multiplexing is not possible. Furthermore, for n channels HSCSD requires n times signalling during handover, connection setup and release. Each channel is treated separately..

GPRS

The General Packet Radio Service (GPRS) is a data service based on an extended GSM network. The GSM system allocates between one and eight time slots within a TDMA frame (per direction) for each GPRS user. The time slots are not allocated in a fixed, pre-determined manner but on demand. All time slots can be shared by the active users, up and down link are allocated separetly.

Users of GPRS can specify a QoS profile. This profile determines the service precedence (high, normal, low), reliability class, delay class, and the user data throughput.

The GPRS architecture introduces two new network elements, which are called GPRS support nodes (GSN). All GSNs are integrated into the standard GSM architecture, and many new interfaces have been defined, see Figure E.5. The gateway GPRS support node (GGSN) is the networking unit between the GPRS network and external data network (PDN). This node contains routing information for GPRS users, performs address conversion, and tunnels data to the user via encapsulation. The GGSN is connected to external networks (e.g. IP or x.25) via the G_i interface and transfers packets to the SGSN via an IP-based GPRS backbone network (G_n interface).

The other new element is the serving GPRS support node (SGSN) which supports the MS via the G_b interface. The SGSN, for example, requests user addresses from the GPRS register (GR), keeps track of individual MSs' location, is responsible for collecting billing information, and performs several security functions such as access control. The SGSN is connected to a BSC via frame relay and is basically on the same hierarchy level as the MSC. The GR, which is typically a part of the HLR, store all GPRS-relevant data.

Packet data is transmitted from a PDN, via GGSN and SGSN directly to the BSS and finally to the MS. The MSC, which is responsible for data transport in traditional circuit-switched GSM, is only used for signaling in the GPRS scenario.

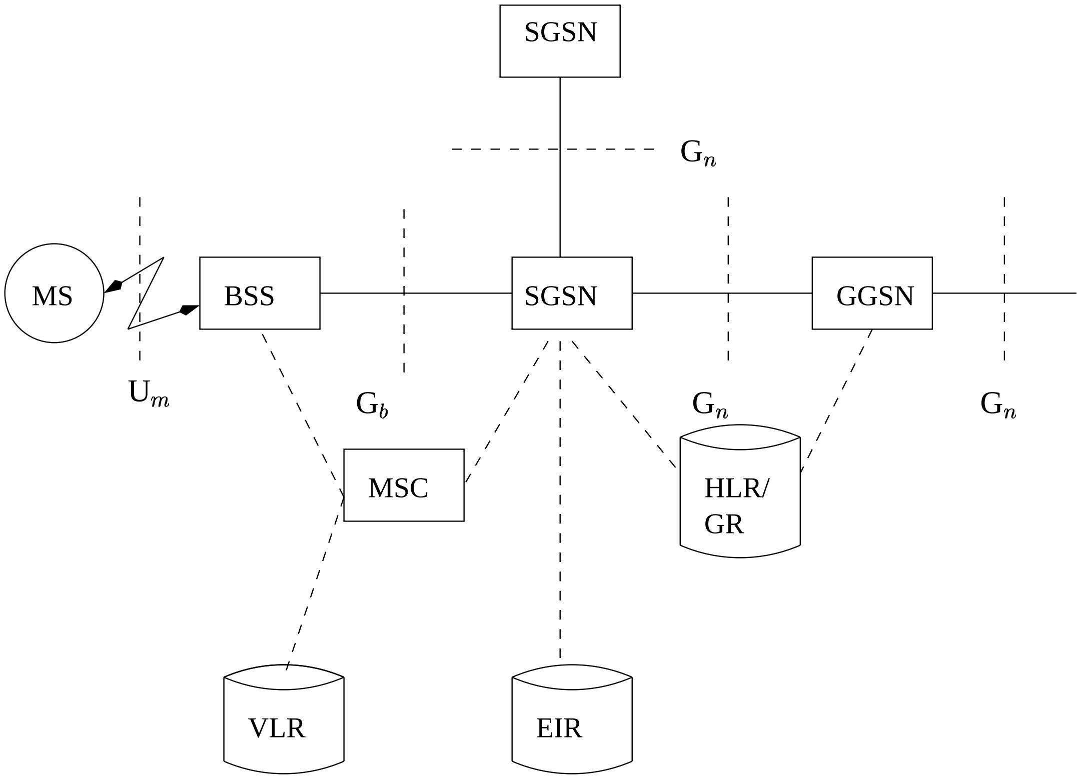


Figure E.5: Architecture of a GSM/GPRS network

EDGE

The Enhanced Data Rate for GSM Evolution (EDGE) is an extension/enhancement of GPRS that provides packet-oriented service at up to 384 kbps. EDGE uses a new modulation technique (8-PSK) and other techniques with the same 2 kHz wide carrier and the same frequencies as GSM. EDGE can be introduced incrementally offering some channels with EDGE enhancement that can switch between EDGE and GSM/GPRS.

E.2.2 UMTS

The International Mobile Telecommunications 2000 (IMT-2000) is a program initiated by ITU-T for a worldwide communication system that allows for terminal and user mobility supporting the idea of universal personal telecommunication. The European proposal for IMT-2000 prepared by ETSI is called Universal Mobile Telecommunications Systems (UMTS) [62]. UMTS represents an evolution from the second generation GSM system to the third generation rather than a completely new system.

UMTS offers tele services and bearer services. Only bearer services will be standardized specifying bit rate, bit error rate and delay time. Tele services are the actual applications (including man-machine interface) from the users perspective. A tele service can make use several bearer services. Tele services can be created independently by each service provider or network operator and offered in the network to the customers. The only exception to this rule is four UMTS tele services standardized by ETSI: speech, fax, SMS, and emergency call.

Bearer services have different QoS parameters for maximum transfer delay, delay variation and bit error rate. Offered data rate targets are:

- 144 kbps satellite and rural outdoor
- 385 kbps urban outdoor
- 2048 kbps indoor and low range outdoor

The bearer services are divided into four different QoS classes:

- Conversational class
- Streaming class
- Interactive class
- Background class

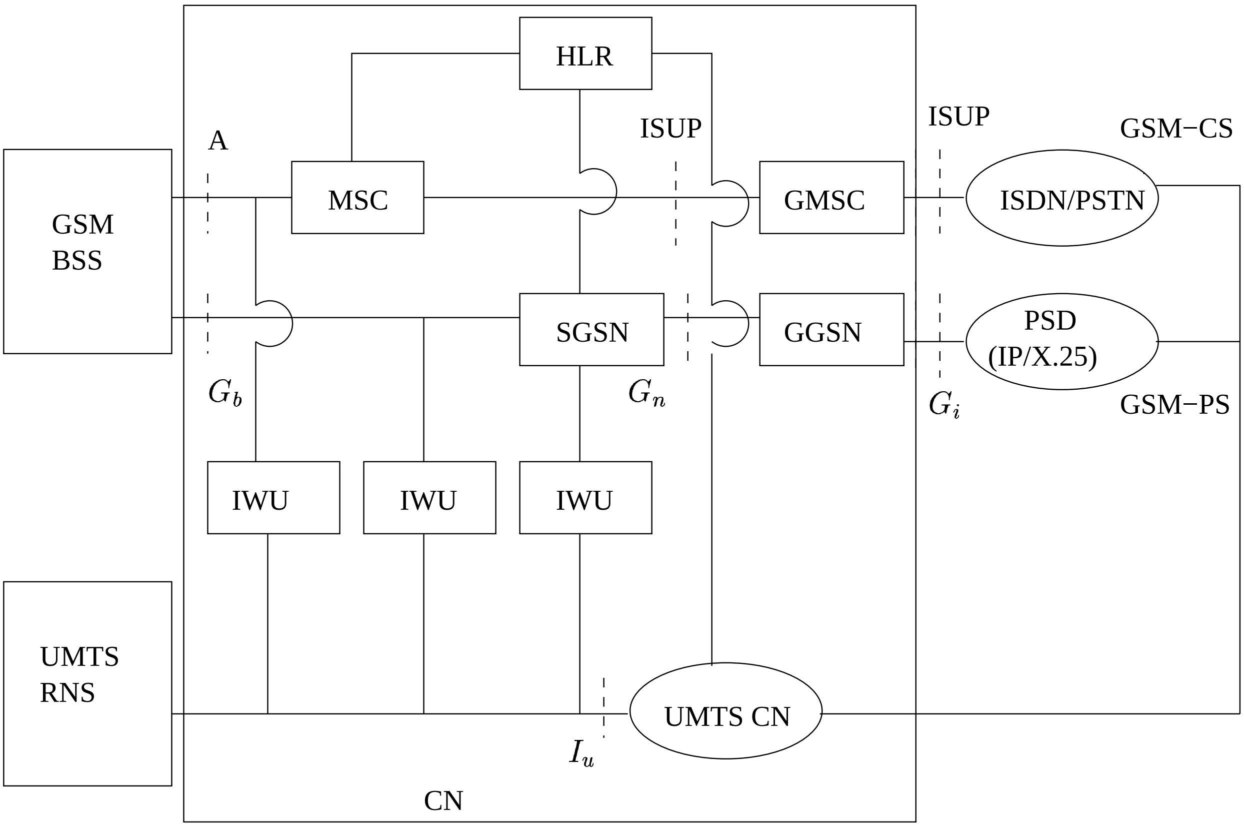
Conversational and Streaming class are mainly intended for real-time traffic flows. The main divider between them is how delay sensitive the traffic is. Conversational real-time service, like video-telephony, are the most delay sensitive applications and those data streams should be carried by the Conversational class. The Streaming class is based one way transport and examples include real time audio stream. Both Conversational and Streaming class preserves the time relation (variation) between the information entities in the stream.

The Interactive class and the Background class are mainly meant for traditional Internet applications like WWW, Email, FTP and News. They both have loser delay requirements compared to Conversational and Streaming classes. The error rate is kept low means of channel coding and retransmission. The main difference between Interactive class and Background class is that Interactive class is mainly used by interactive applications, e.g. interactive Email or interactive web browsing, while the Background class is meant for background traffic, e.g. background download of emails or files. Responsiveness of the interactive applications is ensured by giving traffic in the Interactive class higher priority in the scheduling of network resources.

UMTS architecture

The UMTS network infrastructure, as defined by ETSI, is divided into two separate domains, the Access Network (AN) domain and the Core Network (CN) domain connected to each other via an IWU (Inter Working Unit). The interface between these domains is called I$_u$ and it allows different instances of CNs to be connected to the Access Network. In UMTS phase 1 it is likely that the UMTS Access Network, i.e. UMTS Terrestial Radio Access Network (UTRAN), will be connected with the GSM phase 2+ NSS (Network Subsystem) functioning as the Core Network. The architecture of the UTRAN network is shown is Figure E.7. The UMTS network architecture following this evolution path from GSM platform towards UMTS is illustrated in Figure E.6.

The GSM NSS in phase 2+ will be capable of handling both conventional circuit switched transmission introduced already in GSM phase 1 and the packet switched transmission provided by GPRS. The circuit switched transmission path between the GSM BSS (Base Station Subsystem) and external networks is routed through the GSM network via the MSC (Mobile services Switching Center) and the GMSC (Gateway MSC) while the packet switched



BSS: Base Station Subsystem

RNS: Radio Network Subsystem

IWU: Inter Working Unit

HLR: Home Location Register

SGSN: Serving GPRS Support Node

GGSN: Gateway GPRS Support Node

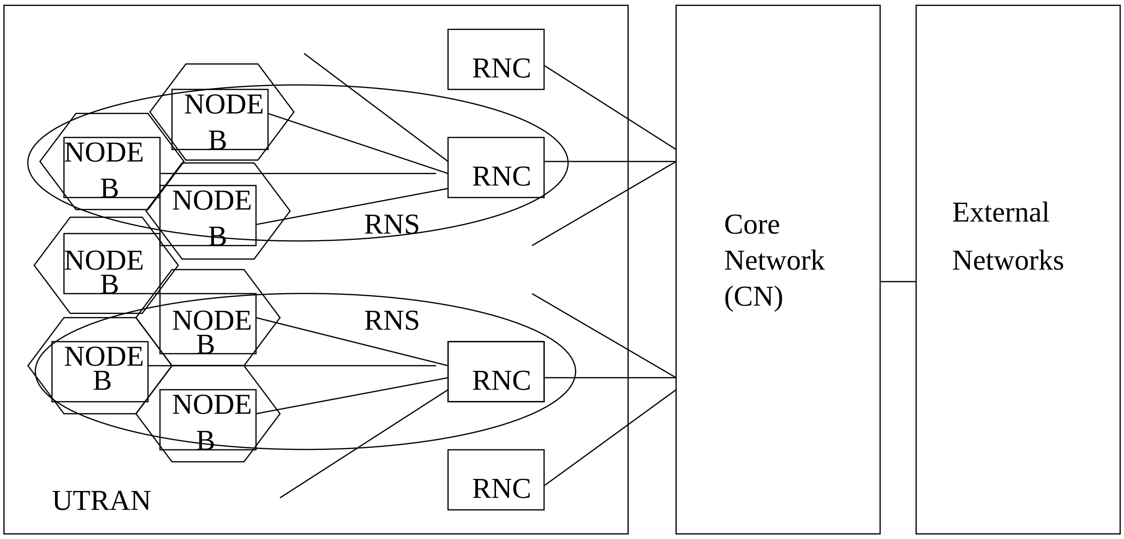
MSC: Mobile services Switching Center

GMSC: Gateway MSC

Figure E.6: Evolution of GSM platform towards UMTS

transmission is routed via GPRS components SGSN (Serving GPRS Support Node) and the GGSN (Gateway GPRS Support Node). The UTRAN will be interconnected to this core network via two IWUs, one between the I$_i$ interface and the the GSM A interface, and another between the I$_u$ interface and the GPRS G$_b$ interface. This architecture makes it possible for both GSM and UMTS customers to be connected both to circuit switched (e.g. PSTN and N-ISDN) and packet switched networks (e.g. Internet and intranets). Additionally, users should also be able to roam between GSM and UMTS networks.

In order to provide real multimedia services for the UMTS customers, the UMTS Core Network (UMTS CN) is introduced later in subsequent phases of UMTS. The UMTS CN will be connected to UTRAN directly via the I$_u$ interface without an IWU. However, the UTRAN remain interconnected to the GSM NSS to ensure interworking between UTRAN and GSM NSS.



Node B: Base Sation

RNS: Radio Network Subsystem

RNC: Radio Network Controller

Figure E.7: UTRAN network architecture

ATM will be used in the CN in the first phase of UMTS, while IP will be use in the second phase.

Wideband CDMA (WCDMA) technology was selected for the UTRAN air interface. In WCDMA the user data is multiplied with quasi-random bits derived from WCDMA spreading codes. WCDMA has two basic modes: Frequency Division Duplex (FDD) and Time Division Duplex (TDD).

E.3 Satellite

Applications of communication via satellite include [62]:

- **Global telephone backbones**: Satellites have been used for international telephone calls during 30 years. Today, satellites are increasingly being replaced by optical cables. While the signal to a geostationary satellite has to travel 72,000 km from a sender via satellite to the receiver, the distance is typically less than 10,000 km if a fibre optical link crossing the Pacific or Atlantic ocean is used. The one-way, single-hop time delay for geostationary satellites is 0.25 s.

- **Connections for remote places or developing areas**: Many places over the world do not have direct wired connection to the telephone network or Internet due to their geographical location or the to the current state of infrastructure of a country. Satellite now offer a simple and quick connection to global networks.

- **Global mobile communication**: The latest trend for satellites is the support of global mobile data communication. Due to the high latency, geostationary satellites are not ideal for this task, therefore, satellites using lower orbits are needed. The basic purpose of satellites for mobile communication is to complement the coverage area of cellular phone systems (such as AMPS and GSM).

A communication satellite can cover a certain ares on earth with is beam, the so-called *footprint*. Within the footprint, communication with the satellite is possible for mobile users via a *mobile user link*. Communication with the base station controlling the satellite and acting as gateway is possible via the *gateway link*. Additionally, satellites might have capabilities to communicate directly with each other via *inter-satellite links*.

Four different types of orbits can be identified:

- **Geostationary earth orbit (GEO)**: GEO satellites have a distance of almost 36,000 km to the earth. Examples are almost all TV and radio broadcast satellites, many weather satellites, and satellites operating as backbones for the telephone network.

- **Medium earth orbit (MEO)**: MEOs operate at a distance of about 5,000 to 12,000 km. Up to now there has not been many satellites in this class, but some upcoming systems use this class for various reasons.

- **Low earth orbit (LEO)**: While some time ago LEO satellites where mainly used for espionage, several of the new satellite systems now rely on this class using altitudes of 500 to 1,500 km.

- **Highly elliptical orbit (HEO)**: This class comprises all satellites with non-circular orbits. Currently, only a few commercial communication systems using satellites with elliptical orbits are planned.

GEO

- **Advantages**: Three GEO satellites are enough for a complete coverage of almost any spot on earth. Senders and receivers can use fixed antenna positioning, no adjusting is needed. Therefore, GEOs are ideal for TV and radio broadcasting. Lifetime expectations for GEOs are rather high, at about 15 years. GEOs typically do not need handover due to a large footprint.

- **Disadvantages**: Northern an southern regions of the earth have more problems receiving these satellites due to the low elevation above 60 $\deg$, i.e. larger antennas are needed in this case. The biggest problem for voice and data communication is the high latency of 0.25 s one-way. Thus, many retransmission schemes which are known for fixed networks fails.

LEO

As LEOs circulate on a lower orbit, it is obvious that they discover much shorter period (typically 95 to 120 minutes). Each LEO satellite will only be visible from earth for around 10 minutes.

- **Advantages**: The delay for packets delivered via LEO is is comparable with long-distance wired connections (10 ms). Smaller footprints of LEOs allow for better frequency reuse, similar to the concepts used for cellular networks. LEOs can provide a much higher elevation is polar regions and, thus, a better global coverage.

- **Disadvantages**: The biggest problem with the LEO concept is the need for many satellites if global coverage is to be reached. Several systems involve 50-200 or even more satellites in orbit. The short time of visibility with a high elevation requires additional mechanisms for handover, as does routing of packets from satellite to satellite. LEOs have relatively short life times of about 5-8 years.

MEO

MEOs can be positioned between LEOs and GEOs, both in terms of their orbit and due to their advantages and disadvantages.

- **Advantages**: MEO systems require only a dozen satellites for global coverage which is more than GEO systems but much less than LEO systems. Satellites periods are about six hours which allows simpler system designs.

- **Disadvantages**: Due to the large distance to each, one-way delay is about about 70-80 ms.

Routing

A satellite system together with gateways and fixed terrestrial network has to route data transmissions from one user to another as any other network does. Routing in the fixed segment (on earth) is achieved as usual, while two different solutions exists for the satellite network in space.

In the first solution the satellite supports inter-satellite links. One user sends data up to a satellite and the satellite forwards it to the one responsible for the receiver via other satellites. This last satellite sends the data down to earth. This means that only one uplink and one downlink per direction is needed. The ability of routing within the satellite network reduces the number of gateways needed on earth.

In the second solution the satellite system does not offer inter-satellite links. The user sends data to the satellite which now forward the data to a gateway on earth. Routing takes place in fixed networks as usual until another gateway is reached which is responsible for the satellite above the receiver. Again data is sent to to the satellite which forwards it down to the receiver. This solution requires two uplinks and two downlink. Depending on the orbit and speed of the routing in the satellite network compared to the terrestrial network, the solution with ISLs might offer lower latency.

Localization

Localization of users in satellite networks is done similarly to that of terrestrial networks. One additional problem arises from the fact that now the 'base stations', i.e. the satellites, move as well. The gateway of a satellite maintain several registers. The *home location register* (HLR) stores all static information about a user as well as his or hers current location. The last known location of a mobile user is stored in the *visitor location register* (VLR). The functionality of HLR and VLR are similar to their counterparts in cellular networks, e.g. GSM. A particularly important register in satellite networks is the *satellite user mapping resister* (SUMR). This register stores the current position of satellites and a mapping of each user to the current satellite through which communication is possible.

Handover

An important topic in satellite systems using MEOs and in particular LEOs is handover. Imagine a cellular mobile phone network with fast moving base stations. This is exactly what such satellite systems are - each satellite represents a base station for a mobile phone. Thus compared to a terrestrial mobile phone networks additional instances of handover can be necessary due to the movement of satellites.

- **Intra-satellite handover**: A user might move from one spot beam to another sport beam of the same satellite. Using special antennas, a satellite can create several spot beams thin its footprint. The same effect might be caused by the movement of the satellite.

- **Inter-satellite handover**: If a user leaves the footprint of a satellite or if the satellite moves away, a handover if the user to the next satellite takes place.

- **Gateway handover**: While the mobile user and satellite might still have good contact, the satellite might move away from the current gateway. Therefor, the satellite has to connect to another gateway.

- **Inter-system handover**: While the three types of handover mentioned above takes place within the satellite-based communication system, this type of handover concerns other systems. As soon as traditional cellular networks are available, users might switch to this type since it is is typically cheaper and offers lower latency. Current systems allow for use of dual mode (or even more modes), but unfortunately, seamless handover between satellite systems and terrestrial systems or vice versa has no been possible up to now.

Examples

The Iridium system has been in operation since 1998. It is the first commercial LEO system covering whole the world. The Iridium system consists of 66 satellites and offers 2.4 kbps connections. Routing between satellites is done via inter-satellite links. Link access is based on a FDMA/TDMA scheme with TDD.

A direct competitor of Iridium is Globalstar. This system uses a lower number of satellites with fewer capabilities per satellite and the overall system is cheaper. Globalstar does not provide inter-satellite links and global coverage, but higher bandwidth is granted to the customers (9.6 kbs). Globalstar use the CDMA link access scheme.

The Intermediate Circular Orbit (ICO) is a commercial MEO system. ICO needs less satellites (10) to reach global coverage. The offered bandwidth is 4.8 kbps and the link access scheme is FDMA/TDMA.

Finally, a very ambitious LEO project is Teledesic which plans to provide high bandwidth satellite connections worldwide with high QoS. In contrast to other systems, this satellite network is not primarily planned for access using mobile phones, but to enable worldwide access to the Internet via satellite. The goal is to offer 64 Mbps downlinks and 2 Mbps uplinks. The plan involves 288 satellites in low orbit. Routing between the satellites will rely on inter-satellite links and the link access scheme will be FDMA/TDMA. Fast packet switching is planned for the satellite network. Service start is targeted to 2003.

E.4 Broadcast systems

Broadcast systems distributes information to multiple receivers on a downlink channel. Interactive applications require an uplink channel, which can be provided by e.g. N-ISDN, ADSL or GSM.

Future television and radio broadcast transmission will be fully digital. Already several radio stations produce and transmit their programmes digitally via the Internet or digital radio. Digital television is on its way. Besides transmitting video and audio, digital transmission allows for the distribution of arbitrary digital data.

E.4.1 Digital Audio Broadcasting (DAB)

The Digital Audio Broadcasting (DAB) is a standard propose by ETSI in 1997 for broadcast of digital audio data [62]. However, DAB can also be used for data services. DAB transmits in the VHF and UHF frequency bands. Within every frequency block of 1.5 MHz, DAB can transmit up to six stereo audio programs with data rate of 192 kbps each. Depending on the redundancy coding, a data service with rate up to 1.5 Mbps is available as an alternative. Bandwidth can be assigned dynamically to music programs or data services by dynamic bit rate management during transmission.

Transmission in DAB is done as follows. Audio services are encoded (MPEG compression) and coded for transmission (FEC). All data services are multiplexed and also coded for redundancy. A multiplexer combines all user data streams and forwards them to the transmission multiplexer. This unit creates the final frames used for transmission and broadcast them on the radio channel using coded orthogonal frequency division multiplex (COFDM) and differential quadrature phase shift keying (DQPSK) signal modulation.

DAB defines different transmission modes, each of which has certain strengths that make it more efficient for either cable, terrestrial, or satellite transmission.

E.4.2 Digital Video Broadcasting (DVB)

Digital Video Broadcasting (DVB) is a standard for digital TV broadcast proposed by ETSI in 1997 [62]. Like DAB, DVB can also be used to distribute any form of digital data. DVB is defined for the cable, terrestrial and satellite environment. Similarly to DAB, DVB use MPEG to code/compress the digital video. Transmission rates between 5 to 38 Mbps is possible over 8 MHz channels. DVB frames containing 188 bytes coded with redundancy (FEC). The DVB transmitter sends the frames using COFDM as multiplexing scheme. The signal is modulated by QAM or QPSK.

Bibliography

[1] Adas A., Traffic models in broadband networks, *IEEE Communications Magazine,* Vol 35., No 7, July 1997

[2] Andersen A. and Nielsen B., A Markovian approach for modelling packet traffic with long range dependence, *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 5, June 1998.

[3] Ash G., An analytical model for adaptive routing networks, *IEEE Transactions on Communications*, Vol. 41, No. 11, November 1993.

[4] ATM Forum, Traffic Management Specification, Version 4.0, 1996.

[5] ATM Forum, Private Network-Network Interface specification version 1.0 (PNNI 1.0), ATM Forum af-pnni-0055.00, March 1996.

[6] Blake S., Black D., Carlson M,, Davies E., Wang Z. and Weiss. W. An Architecture for Differentiated Services, IETF RFC (Informational) 2475, December 1998.

[7] Bohn R., Braun H., Claffy K and Wolf S., Mitigating the comming Internet crunch: multiple service levels via presendence, Tehnical Report, UCSD, San Diego Supercomputer Center, and NSF, March 1994.

[8] Braden R., Clark D. and Shenker S., Integrated Services in the Internet Architecture, IETF RFC 1633, 1994.

[9] Brichet F., Roberts J., Simonian A. and Veitch D., Hevay traffic analysis of a storage model with long range dependent on/off sources, *Queueing Systems*, No 23., pp. 197-215, 1996.

[10] Carpenter R., *Neurophysiology,* second edition, Edward Arnold, 1990.

[11] Clark D., A model for cost allocation and pricing in the Internet, In McKnight L. and Bailey J., *Internet Economics,* MIT press, Cambridge, MA, 1997.

[12] Courcoubetis C. and Siris V., An approach to pricing and resource sharing for Available Bit Rate (ABR) services, FORTH-ICS/TR-212, November 1997

[13] Courcoubetis C., Siris V. and Stamolis G., Application of the many sources asymptotic and effective bandwidths to traffic engineering, *Telecommunications Systems,* No. 12, pp. 167-191, 1998.

[14] Crovella M. and Bestavros A., Self-similarity in world wide web traffic – evidence and possible causes, *IEEE/ACM Transactions on Networking,* Vol. 5, No 6, pp. 835-846, 1997.

[15] Deering S. and Hinden R., Internet Protocol Version 6 Specificaation, IETF RFC 1883, 1995.

[16] Duffield N, Lewis J., O'Connell N., Russell R. and Tomey F., Entropy of ATM streams, *IEEE Journal of Selected Areas in Communications,* Vol 13, No 6., August 1995.

[17] Dziong Z., *ATM Network Resource Management,* first edition, McGraw-Hill, 1997.

[18] Dziong Z. and Mason L., Call admission and routing in multi-service loss networks, *IEEE Transactions on Communications,* Vol. 42, No. 2/3/4, pp. 2011-2022, February/March/April, 1994.

[19] Eun D., Ahn H., Roh H. and Kim J., Effects of long-range dependence of VBR video traffic on queueing performances, *Proc. of GLOBECOM'97,* pp. 1440-1444, Phoenix, USA, Nov. 1997.

[20] Fahmy S., Jain R., Kalyanaraman S., Goyal R. and Vandalore B., On determining the fair bandwidth share for ABR connections in ATM networks, *Proc. IEEE International Conference on Communications (ICC)'98,* volume 3, pp. 1485-1491, June 1998.

[21] Farago A., Blaabjerg S., Gordos G. and Henk T., A new degree of freedom in ATM network dimensioning: optimizing the local configuration, *IEEE Journal of Selected Areas in Communications*, Vol 13., No. 7, September 1995.

[22] Feldmann A., Characteristics of TCP connection arrivals, Tech. Rep., AT&T LAbs-Research, 1998.

[23] Frost V., Melamed B., Traffic modelling for telecommunication networks, *IEEE Communications Magazine*, Vol. 32, No. 2, February 1996.

[24] Fuller V., Li T., Yu J. and Varadhan K., Classless Inter-Domain Routing (CIDR), IETF RFC 1519, 1993.

[25] Garrett M and Willinger W., Analysis, modelling and generation of self-similar VBR video traffic, *Proc. of the ACM SIGCOMM'94*, London, UK, pp. 269-280. 1994

[26] Gibbens R., Kelly F. and Key P, Dynamic alternative routing – modelling and behaviour, *Proc. 12th International Teletraffic Congress*, Turin, Italy, 1988.

[27] Girard A. and Cote Y., Sequential routing optimization for circuit switched networks, *IEEE Transactions on Communications*, Vol. COM-32, No. 12, December 1984.

[28] Guerin R., Ahmadi H. and Nagshineh M., Equivalent capacity and its application to bandwidth allocation in high-speed networks, *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, September 1991.

[29] Gulliksson H., Lindström J. *Multimedia över nätverk*, in Swedish, Studentlitteratur, 2000.

[30] Hui J., A layered broadband switching architecture with physical or virtual path configurations, *IEEE Journal on Selected Areas in Communications*, vol. 9, no 9., pp. 1416-1426, Dec. 1991.

[31] IETF, Internet Protocol Specification, IETF RFC 791, 1981.

[32] IETF, Transmission Control Protocol Specification, IETF RFC 793, 1981.

[33] ITU-T, ISDN service capabilities, B-ISDN service aspects, Recommendation I.211, 1993.

[34] ITU-T, Network performance objectives for IP-based services, Recommendation Y.1541, 2002.

[35] ITU-T, SG12, End-User Multimedia QoS Categories, Draft recommendaion, G.QOSrqt, 2002.

[36] Jacobsen S., Dittman L., A fluid flow queueing model for heterogeneous on/off traffic, RACE BLNT Workshop, Munchen, 1990.

[37] Jacobsen V., Congestion avoidance and control, *Proc, ACM SIGCOMM'88*, pp. 314-329, August 1988.

[38] Jain R., Congestion control and traffic management in ATM networks: recent advances and a survey, *Computer Networks and ISDN Systems*, Vol. 28, No. 13, pp. 1723-1738, October 1996.

[39] Kelly F., Routing in circuit-switched networks: optimization, shadow-prices and decentralization, *Adv. Appl. Prob*, No. 20, 1988.

[40] Kelly F., Notes on effective bandwidths, In Kelly F., Zachary S. and Ziedins I, editors, *Stochastic networks: theory and applications, Vol. 4 of Royal Statistical Society Lecture Notes*, pp. 141-168, Oxford University Press, Oxford, 1996

[41] Kelly F., Charging and accounting for bursty connections, In Mcknight L and Bailey J, editors, *Internet Economics*, pp. 253-278, MIT press, Cambridge, MA, 1997.

[42] Kelly F. Rate control for communications networks: shadow price, proportional fairness and stability, *Journal of the Operational Research Society*, No 49., 1998.

[43] Leland W, Taqqu M., Willinger W. and Wilson D., On the self-similar nature of Ethernet traffic (extended version), *IEEE/ACM Transactions on Networking*, Vol 2, No. 1, pp 1-15, 1994.

[44] MacKie-Mason J. and Varian H., Pricing congestible resources, *IEEE Journal on Selected Areas In Communications*, Vol. 13, No. 7, pp. 1141-1149, 1995.

[45] Maglaris B., Anastassiou D., Sen P., Karlsson G. and Robbins J., Performance lodels of statistical multiplexing in packet video communications, *IEEE Transactions on Communications*, vol 36, no 7, pp. 834-844, July 1988.

[46] McDysan D., *QoS and Traffic Management*, first edition, McGraw-Hill, 2000.

[47] Michiel H. and Laevens K., Teletraffic engineering in a broadband era, *Proceedings of the IEEE*, Vol 85, No 12., December 1997.

[48] Miyao Y., A call admission control scheme in ATM networks, In proceceedings of ICC'91, pp. 391-396, 1991.

[49] Norden S. and Turner J., Inter-domain routing algorithms, preprint, 2002.

[50] Norros I.,On the use of fractional brownian motion in the theory of connectionless networks, *IEEE Journal of Selected Areas in Communications*, Vol. 13, No. 6, August 1995.

[51] Odlyzko A., A modest proposal for preventing Internet congestion, Preprint, AT&T Labs Research, 1997

[52] Onvural R., *Asynchronous Transfer Mode networks*, second edition, Artech House, 1995

[53] Partridge C., *Gigabit networking*, Addison-Wesley, 1994

[54] Paxson V. and Floyd S., Wide-area traffic: the failure of Poisson modelling, *Proc. ACM SIGCOMM'94*, London, UK, 1994.

[55] Peterson L., Davie B., *Computer Networks*, second edition, Morgan Kaufman, 2000.

[56] Pioro M., Wallstrm B., Multihour optimisation of non-hierchical circuit-switched communications networks with sequential routing,*Proc. 11th International Teletraffic Congress*, pp. 788-794, Kyoto, Japan, September, 1985.

[57] De Prycker M., *Asynchronous Transfer Mode: solution for broadband ISDN*, third edition, Prentice Hall, 1995.

[58] Postel J., User Datagram Protcol Specification, IETF RFC 768, 1980.

[59] Robertazzi T., *Computer networks and systems: queueing theory and performance evaluation,*third edition, Springer, 2000.

[60] Rosen E., Viswanathan A. and Callon R,. Multiprotocol Label Switching Architecture, IETF RFC 3031, 2001.

[61] Seres G., Szlavik A., Zatonyi J. and Biro J., Quantifying resource usage: a large deviation-based approach, *IEICE Transactions on Communications*, Vol. E84-B, No. 9, September 2001

[62] Schiller J, *Mobile Communications,* Addison-Wesley, 2000.

[63] Schultzrinne H., Casner S., Frederick R. and Jacobson V., A Transport Protocol for Real-Time Applications, IETF RFC 1889, 1996.

[64] Shenker S., Clark D., Estrin D. and Herzog S., Pricing in computer networks: reshaping the research agenda, *ACM Computer Communication Review,* pp. 19-43, 1996

[65] Simpson, V., The Point-to-Point Protocol (PPP), IETF RFC 1661, 1994.

[66] Siris A., Songhurst D., Stamoulis G. and Stoer M., Usage-based charging using effective bandwidth: studies and reality, *Proceedings of the 16th International Teletraffic Congress*, Edinburgh, 1999

[67] Songhurst D. (editor), *Charging communication networks: from theory to practice*, Elsevier, 1999.

[68] Stallings W., *Data and Computer Comunications,* sixth edition, Prentice Hall, 2000.

[69] Stallings W., *Wireless Communications and Networks*, first edition, Prentice Hall, 2002.

[70] Tanenbaum A., *Computer Networks*. third edition, Prentice Hall, 1996.

[71] Wang Z. and Crowcroft J., QoS routing for supporting resource reservatio. *IEEE Journal on Selected Areas in Communications*, Vol 14, No 7, pp. 1228-1234, 1996.