Multi-service network architectures

Ernst Nordström

Department of Culture, Media, Computer Science, Dalarna University

SE-781 88 Borlänge, Sweden

eno@du.se

July 7, 2006

Contents

Chapter 1

Communication principles

1.1 Single-service communication networks

The remote communication model shown in Figure 1.1 shows how users are interconnected via access and core (WAN) networks. The access networks are either wired or wireless, but the core network is typically wired, except when the WAN is implemented using satellites. The remote communication model applies to both telephone networks and the global Internet.

Traditionally, WAN communication networks have developed along two tracks: circuit-switched telephone networks and packet-switched data networks. In the former, communication is carried out over “circuits” with dedicated transfer capacity or bandwidth.

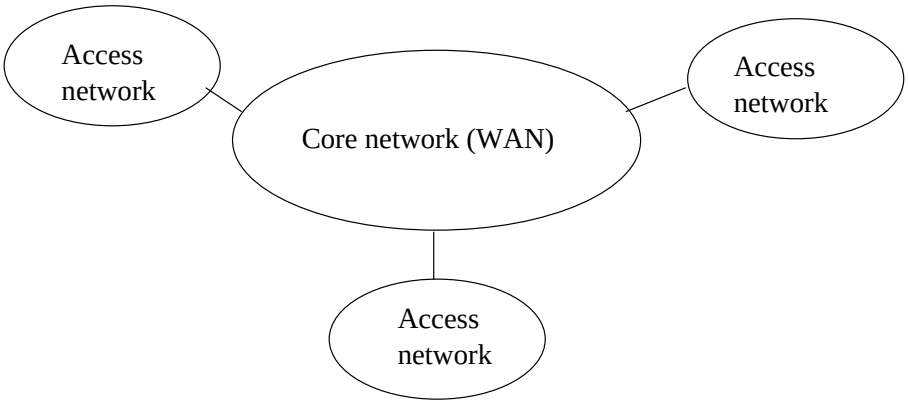


Figure 1.1: Remote communication model

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 1.2.

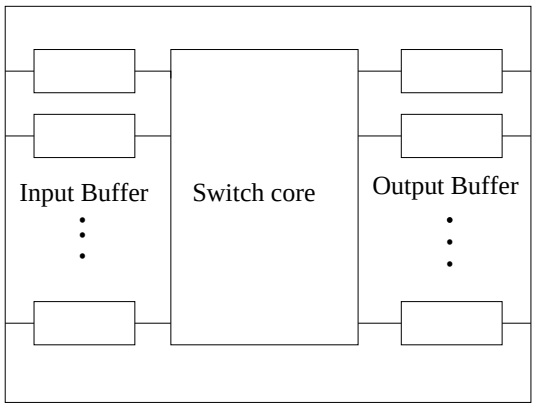


Figure 1.2: Anatomy of a packet switch/router

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 inter-connected 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.

1.2 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.

1.3 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. 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*.

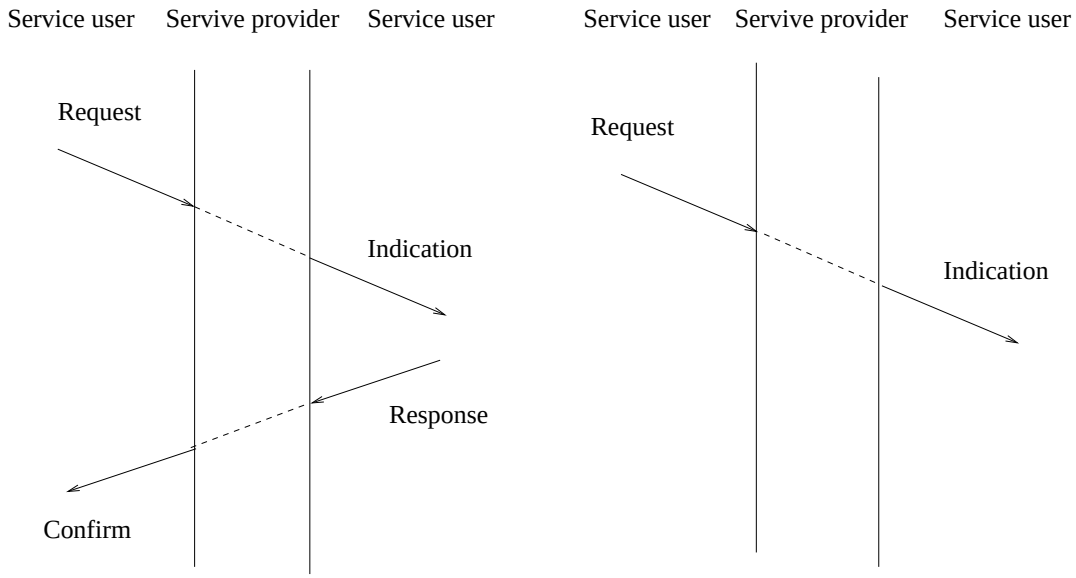
Services are available at *Service Access Points* (SAPs). Each SAP has an address that uniquely identifies it. To make things clearer, the SAPs in the telephone system are the sockets in which modular

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



(a) Confirmed service (b) Nonconfirmed service

Figure 1.3: Time sequence diagrams for service primitives.

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 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.

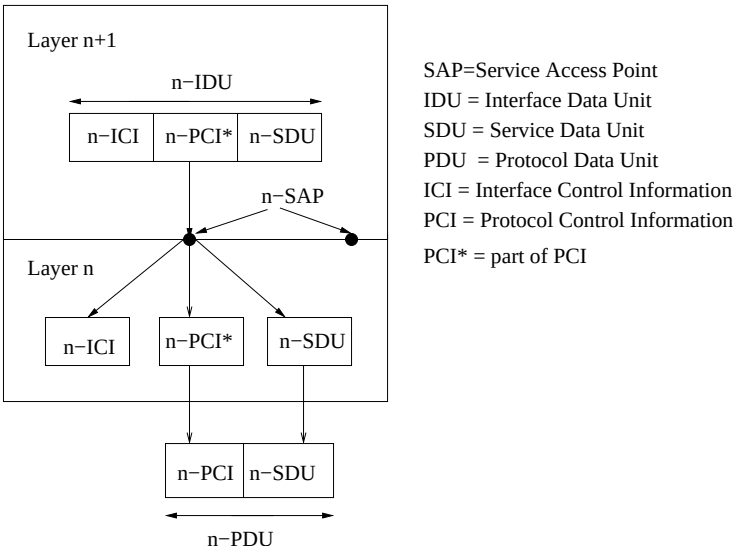


Figure 1.4: Relation between layers at an interface

1.4 ISO OSI reference model

The International Standards Organization (ISO) proposed in 1984 a reference model for layered organization of communication protocols. The model was called the the Open Systems Interconnection (OSI) reference model. The ISO OSI model contains seven layers, see Figure 1.5.

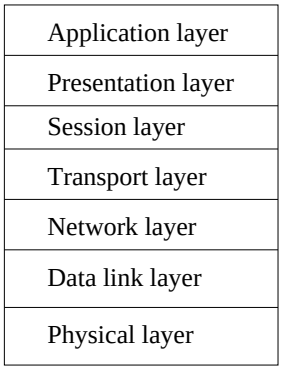


Figure 1.5: The ISO OSI reference model

1.4.1 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 physical transmission medium, which lies below the physical layer. Issues include reliable transfer of bits, modulation of the signal, 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.

1.4.2 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.

1.4.3 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.

1.4.4 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.

1.4.5 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.

1.4.6 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.

1.4.7 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 2

History of communication networks

2.1 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.

2.2 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 320 million in January 2005. The number if autonomous systems (domains) was 16,000 in July 2005, and the number of IP subnetworks was 210,000. 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.

2.3 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.

2.4 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 interoperable 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.

2.5 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.

Chapter 3

ATM service framework

3.1 ATM service architecture

ATM Forum has defined the following service categories [1]:

- 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 3.1 summarizes what QoS and traffic contract parameters are specified for each category.

3.1.1 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).

3.1.2 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.

3.1.3 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.

3.1.4 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).

3.1.5 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.

3.1.6 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.

<table>
<tr><th></th><th>CBR</th><th>real−time VBR</th><th>non− real−time VBR</th><th>ABR</th><th>UBR</th><th>GFR</th></tr>
<tr><td>CLR
Cell Loss Ratio</td><td colspan="3">specified</td><td></td><td>unspecified</td><td>specified</td></tr>
<tr><td>CTD
Cell Transfer Delay</td><td colspan="2">specified</td><td colspan="3">unspecified</td></tr>
<tr><td>CDV
Cell Delay Variation</td><td colspan="2">specified</td><td colspan="3">unspecified</td></tr>
<tr><td>Traffic descriptors
(service contract)</td><td>PCR/
CDVT</td><td colspan="2">PCR/CDVT
SCR/MBS</td><td>PCR/CDVT
MCR/ACR</td><td>PCR/CDVT</td><td>PCR/CDVT
MCR/MBS
MFS</td></tr>
<tr><td>Congestion control</td><td colspan="3">no</td><td>yes</td><td colspan="2">no</td></tr>
</table>

Figure 3.1: Service category attributes specified by the ATM Forum

3.2 ATM Quality of Service

The following QoS parameters are negotiated [1]:

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

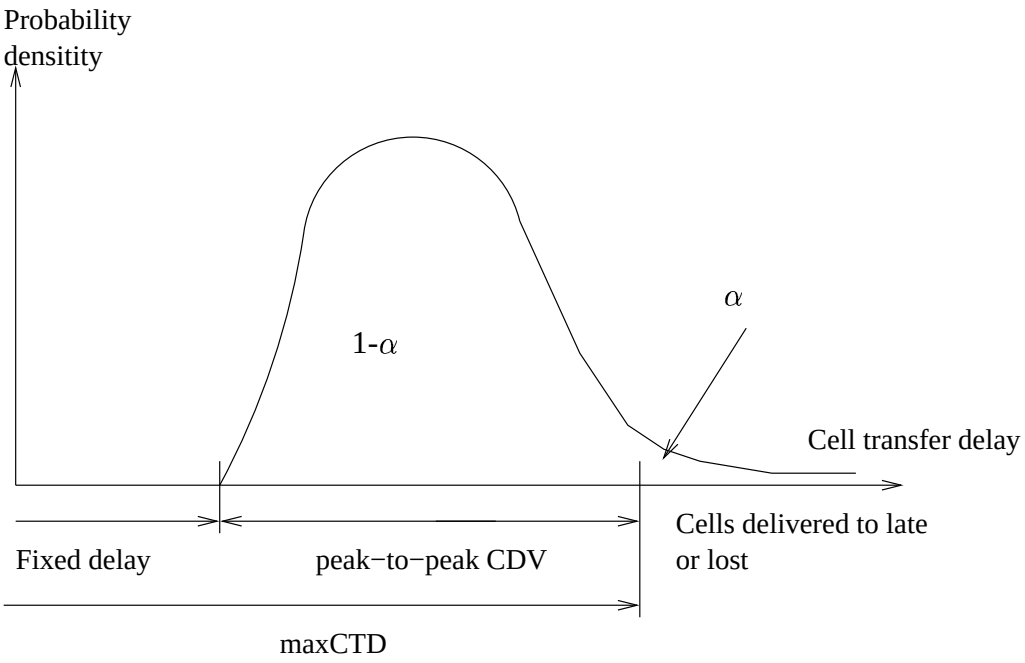


Figure 3.2: Cell transfer delay probability density model

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 3.2. The CLR is defined as:

$$\text{CLR} = \frac{\text{Lost Cells}}{\text{Total Transmitted Cells}} \tag{3.1}$$

The following QoS parameters are not negotiated [1]:

- Cell Error Ratio (CER)
- Severely Errored Cell Block Ratio (SECBR)
- Cell Misinsertion Rate (CMR)

The CER is defined as:

$$\text{CER} = \frac{\text{Errored Cells}}{\text{Successfully Transferred Cells} + \text{Errored Cells}} \tag{3.2}$$

The SEBCR is defined as:

$$\text{SEBCR} = \frac{\text{Severly Errored Cell Blocks}}{\text{Total Transmitted Cell Blocks}} \tag{3.3}$$

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:

$$\text{CMR} = \frac{\text{Misinserted Cells}}{\text{Time Interval}} \tag{3.4}$$

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.

3.3 ATM traffic contract

3.3.1 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) [1].

3.3.2 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.

3.3.3 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.

3.3.4 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 [1].

The GCRA is a virtual scheduling algorithm or a continuous-state Leaky Bucket Algorithm as defined by the flowchart in Figure 3.3. 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.

3.3.5 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.

3.3.6 Traffic enforcement for rt-VBR and nrt-VBR

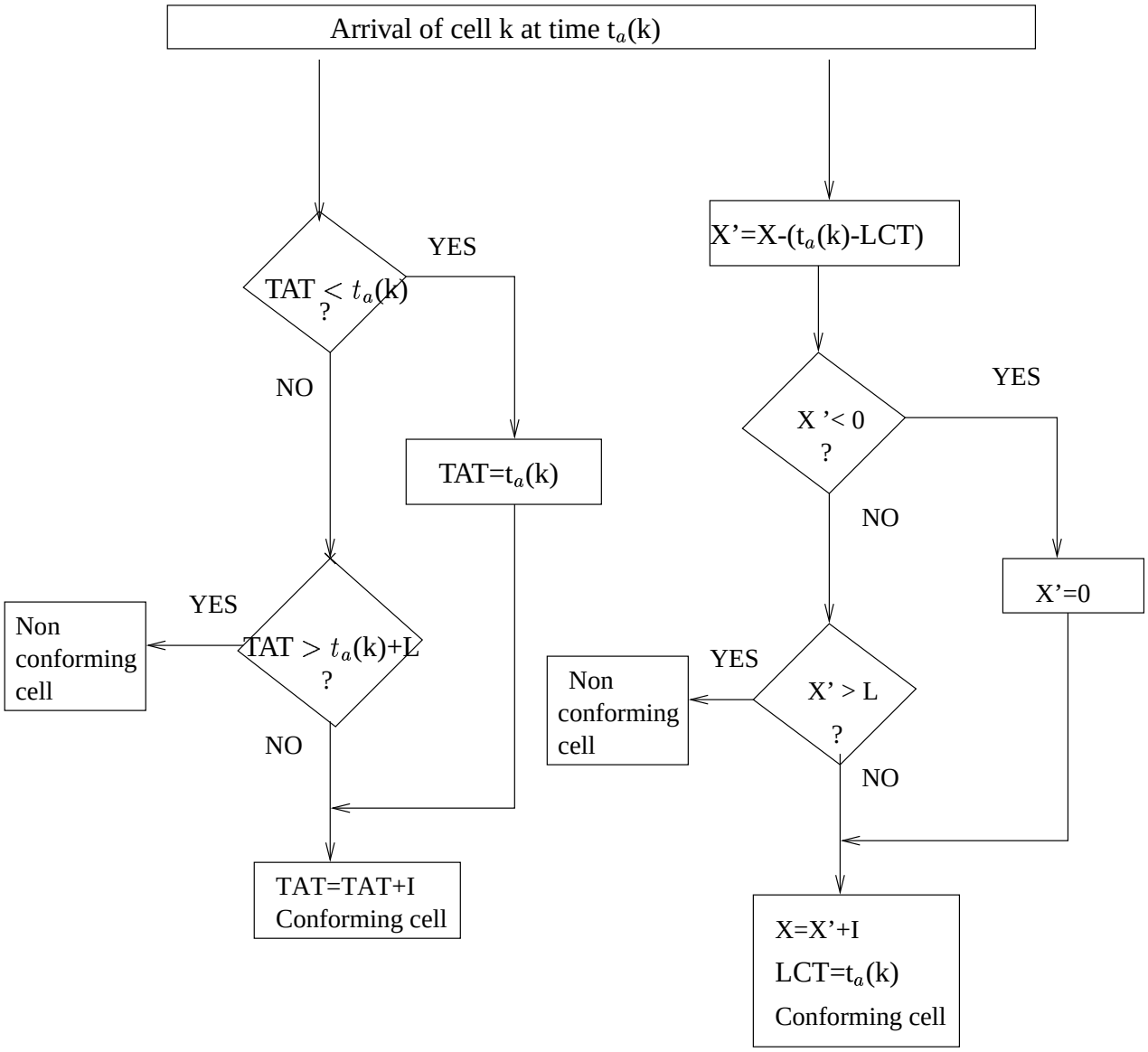
For PCR is enforced by GRCA(T,CDVT). The MBS and SRC are enforced by GCRA(T_s,BT+CDVT), where T_s=1/SCR denotes the cell inter-arrival time when the source is sending at SCR, and BT denotes the burst tolerance time. The MBS is the maximum number of consecutive cells that a source can send at the peak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

$$MBS = \lfloor 1 + \frac{\tau_s}{T_s - T} \rfloor \tag{3.5}$$

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

3.3.7 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



VIRTUAL SCHEDULING ALGORITHM

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

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

CONTINUOUS LEAKY BUCKET ALGORITHM

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 to the connection, X=0 and LCT=t_a(1)

Figure 3.3: Equivalent versions of the GCRA algorithm

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).

Chapter 4

IP service framework

4.1 IP service architecture

The ITU-T has defined six IP QoS classes in recommendation Y.1541 [9]. Figure 4.1 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 4.1: IP QoS class definitions

4.2 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.

4.3 IP traffic contract

4.3.1 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.

4.3.2 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.

4.3.3 Enforcement by the token bucket algorithm

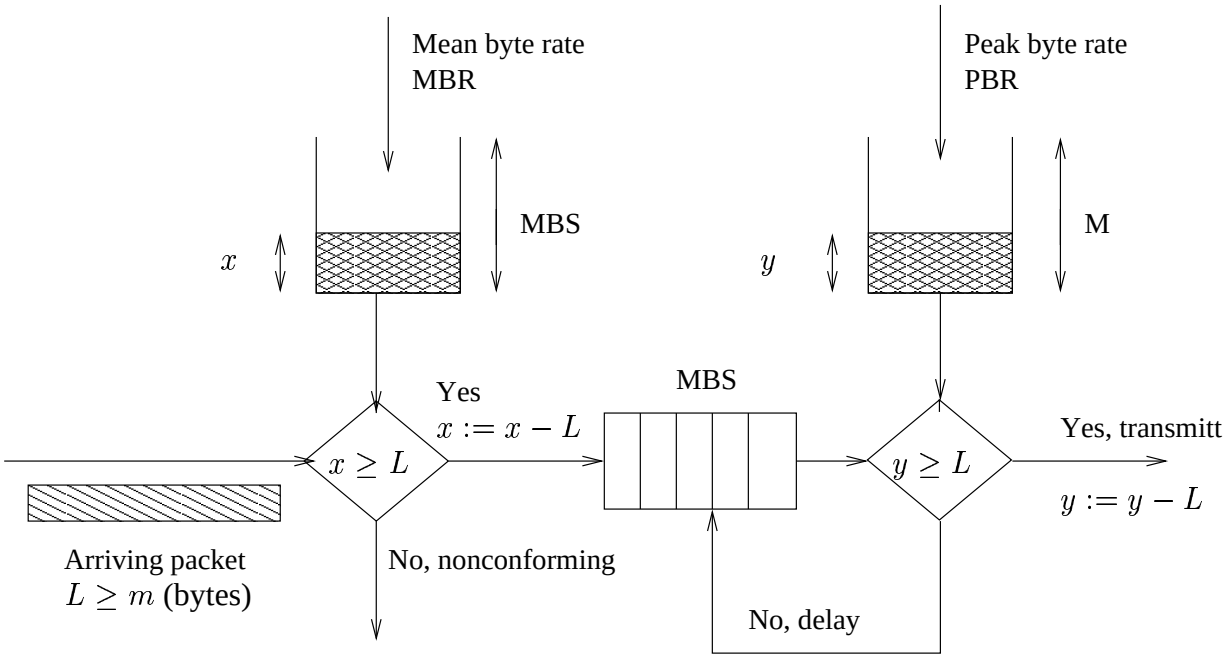


Figure 4.2: Token bucket algorithm

The traffic parameters of the IP source are enforced at the ingress node of the IP domain by a token bucket algorithm [11], see Figure 4.2. 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.

Chapter 5

Multimedia application framework

5.1 Classification

The ITU-T recommendation I.211 specifies two main service categories [8]: 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.

5.2 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 5.1 to 5.3 shows the numerical QoS values and Figure 5.4 presents a graphical summary.

The presentation is based on the ITU draft recommendation G.QoSrqt [10].

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 5.1: 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 5.2: 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 5.3: Performance targets for data applications (part 2)

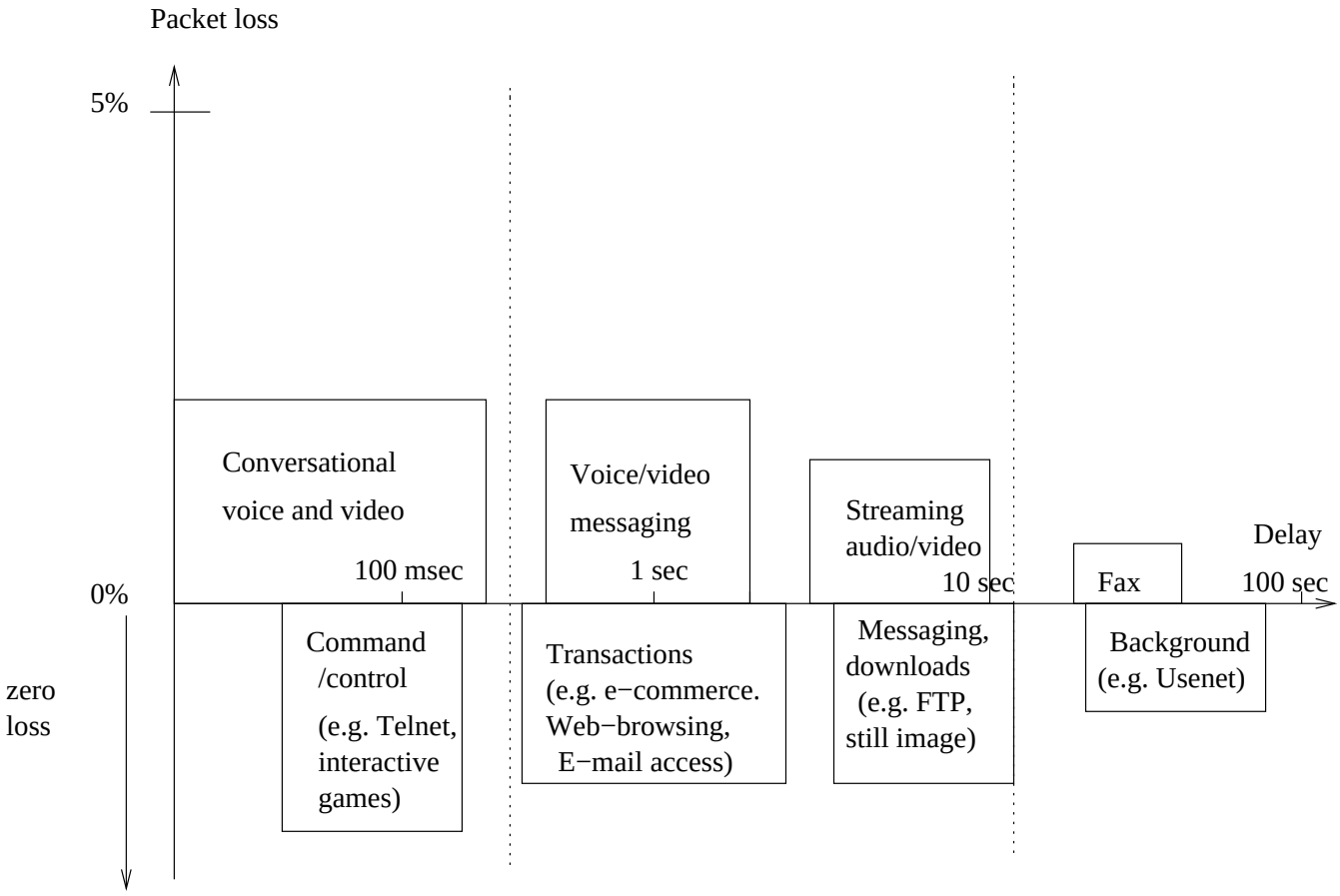


Figure 5.4: Summary of delay and loss requirements for audio, video and data applications.

Chapter 6

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.

6.1 Integrated Services Architecture

IntServ is a per-flow based QoS framework with dynamic resource reservation [3]. 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.

6.2 Differentiated Services Architecture

6.2.1 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 (RFC 2475). DiffServ provides QoS in IP networks by management of aggregates of packet flows. The packets of each flow is identified by Differentiated Services Code Point (DSCP) field in the packet header. The DSCP selects the per-hop-behaviour (PHB) that controls the packet forwarding treatment in each network 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.

We assume that within the autonomous system we have as single DiffServ domain that is a contiguous set of DiffServ nodes that have implemented the same PHB mechanisms and operate with a common service provisioning policy set. All nodes in the DiffServ domain serve data streams in the same way, depending on the aggegate Class of Service (CoS) membership. A DiffServ domain has a well-defined boundary consisting of DiffServ boundary nodes that classify and possible condition ingress traffic. A DiffServ region is a set of one or more contiguous cooperating DiffServ domains (autonomous systems).

Each node in the network may have a number of customers connected to it. In order for a customer to receive differentiated service it must have a service level aggreement (SLA) with its Internet Service Provider (ISP). In general, the SLA specifies the service classes supported and the amount of traffic allowed in each class. An SLA can be static or dynamic. Static SLAs are negotiated on a regular basis (monthly or yearly). In our framework we deal with dynamic SLAs. In this case customers may request services on demand without subscribing to them.

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 QoS requirements 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.

6.2.2 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.

6.2.3 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).

6.2.4 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.

Chapter 7

B-ISDN/ATM

In 1988 the ITU-T standardized *Asynchronous Transfer Mode* (ATM) as the network protocol for implementing *Broadband ISDN* 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.

7.1 ATM reference model

The ATM reference model is shown in Figure 7.1. 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 functions relate to interlayer coordination.

The functionality of the different layers is summarized in Figure 7.2.

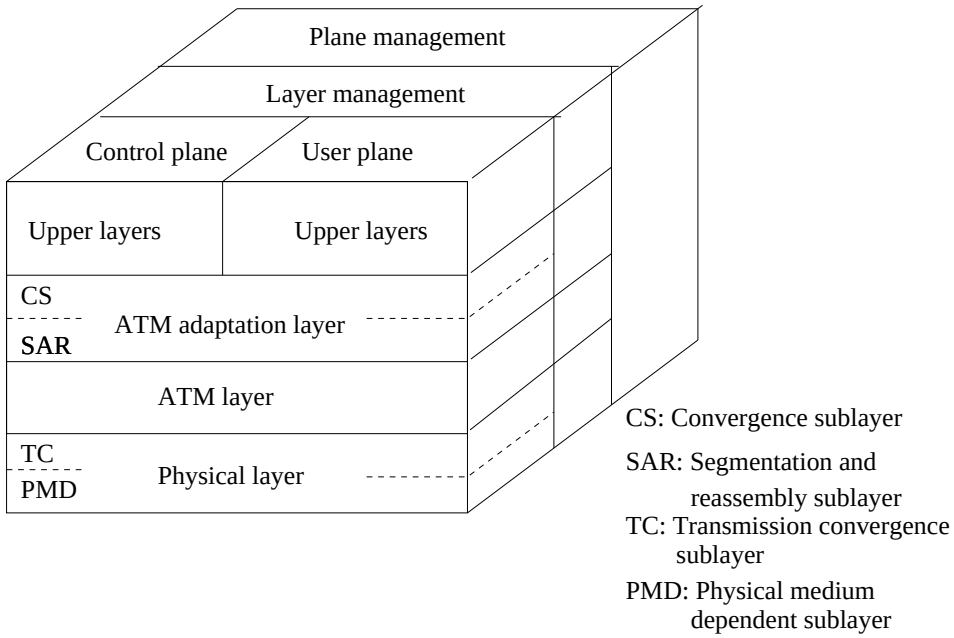


Figure 7.1: ATM reference model

7.2 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.

7.3 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

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 7.2: The ATM layers and sub layers, and their functions

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 7.3.

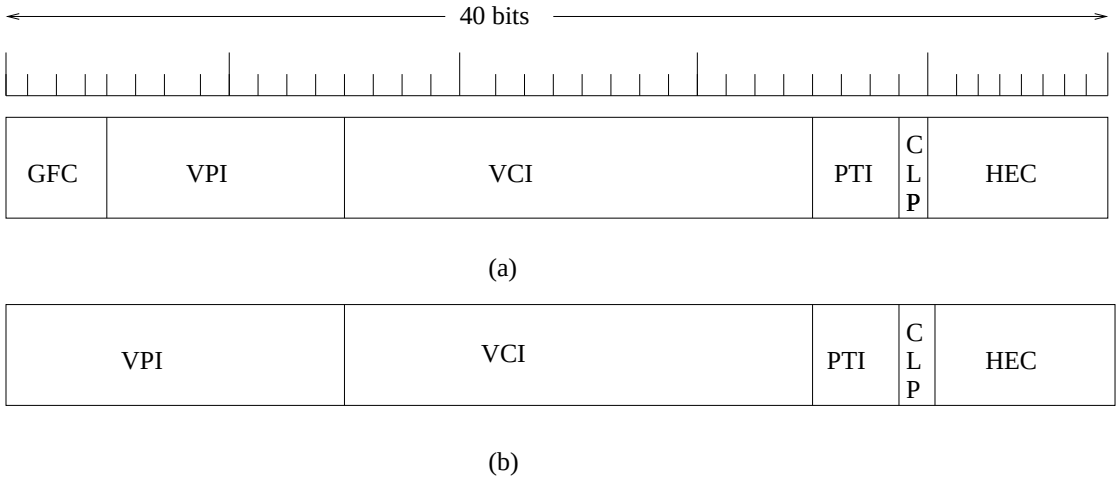


Figure 7.3: (a) ATM layer header at the UNI. (b) The ATM layer header at the NNI.

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.

7.4 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 7.4. 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.

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

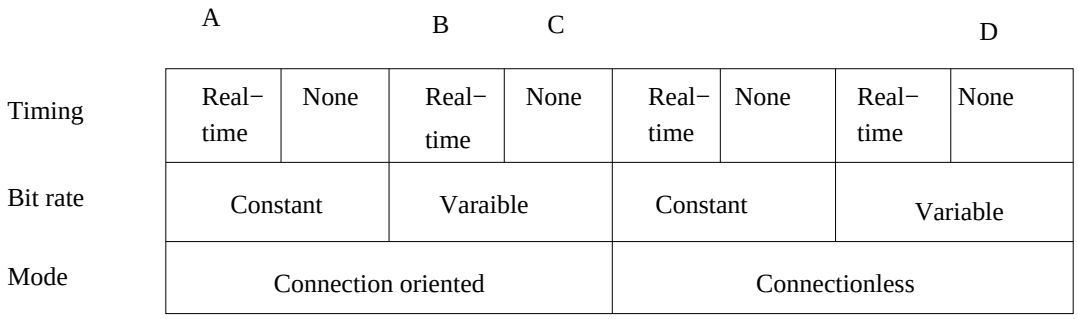


Figure 7.4: ITU service classes supported by AAL.

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.

Chapter 8

Internet

8.1 TCP/IP reference model

The TCP/IP reference model forms the basis for the global Internet. 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.

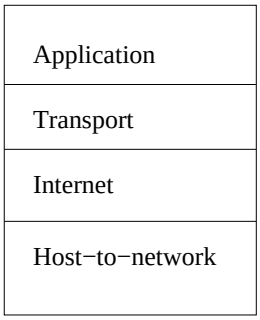


Figure 8.1: TCP/IP reference model

8.2 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.

8.3 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 [6]. The next generation of IP is version 6. IPv6 was defined by IETF in RFC 1883 in 1995 [4]. 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.

8.3.1 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 8.2. A forth class (D) is used for multicast purposes.

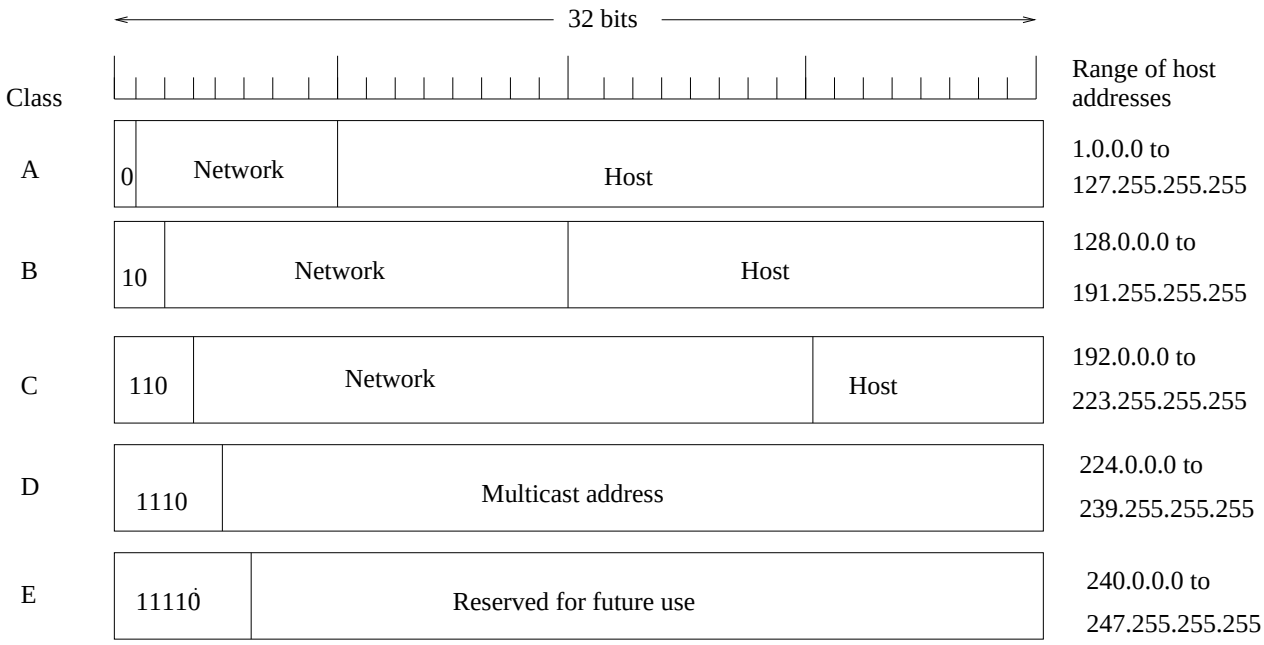


Figure 8.2: IPv4 address formats

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 [5]. 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.

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

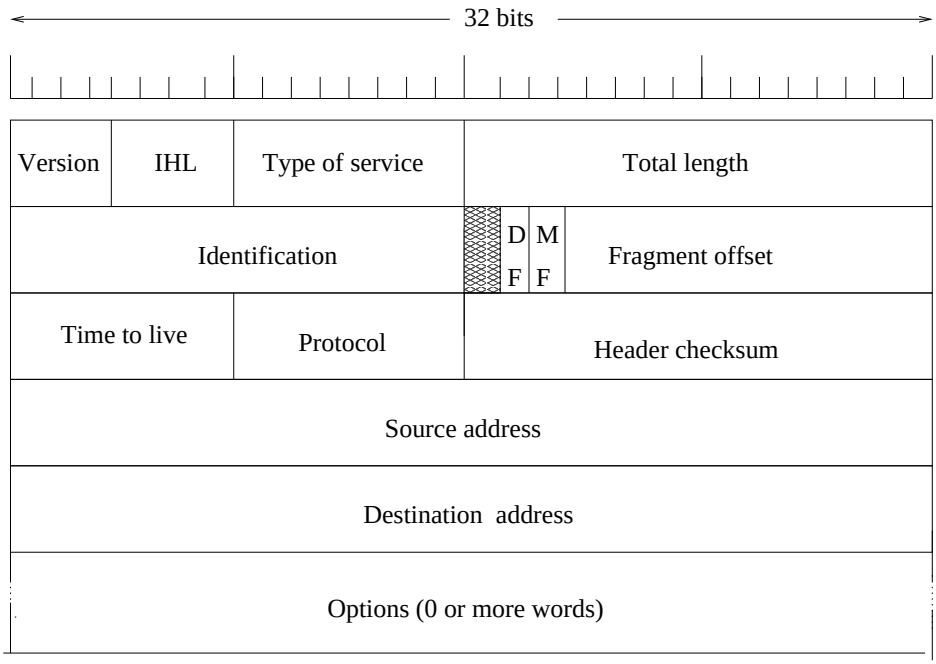


Figure 8.3: Packet header format for IP version 4

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 8.4.

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 8.4: IPv4 options

8.3.2 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 8.6. 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 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.

8.3.3 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.

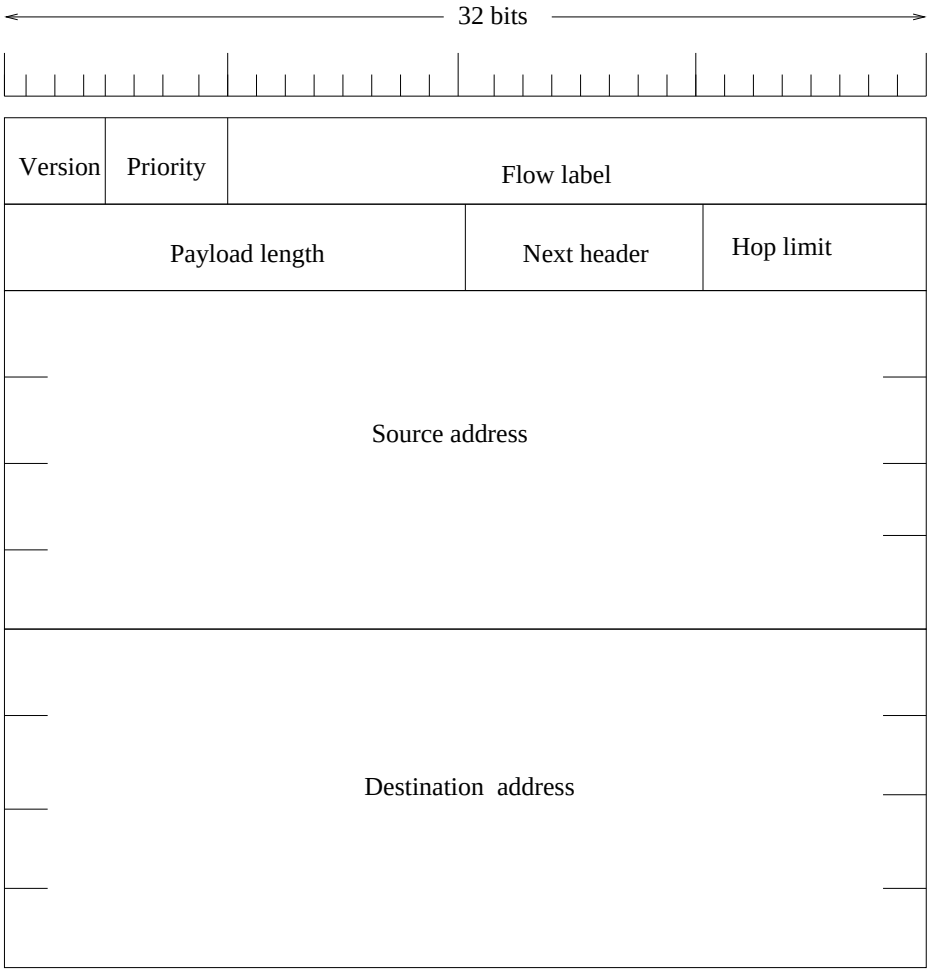


Figure 8.5: Packet header format for IP version 6

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 8.6: IPv6 extension headers

MPLS is an advanced forwarding scheme described in RFC 3031 [13]. 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 be-

tween 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.

8.4 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.

8.4.1 TCP protocol

The *Transport Control Protocol* (TCP) was formally defined by IETF in RFC 793 in 1981 [7]. 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 8.7.

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 8.7: 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 pro-*

tocol 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.

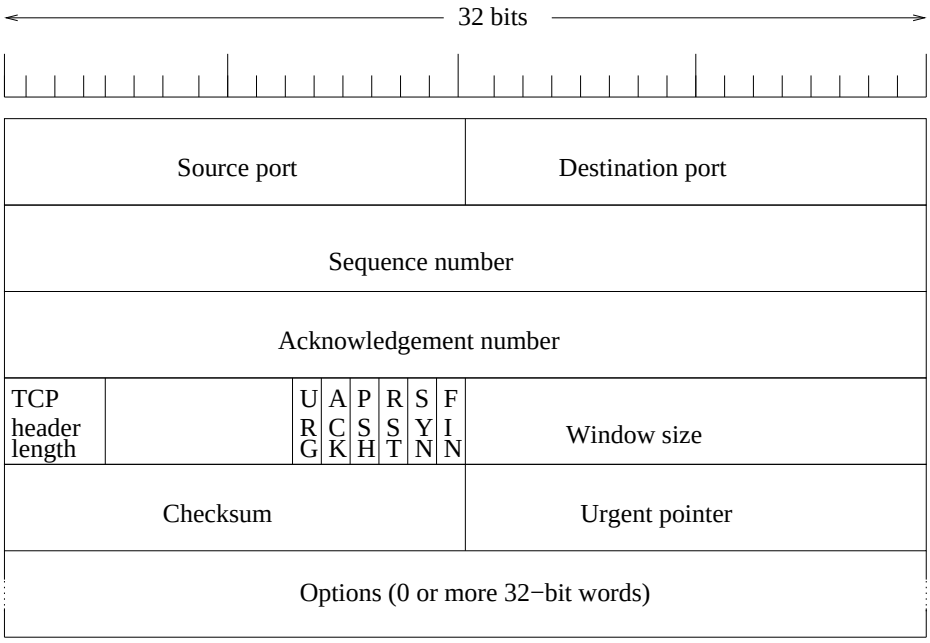


Figure 8.8: Segment header format for TCP

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 8.9.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.

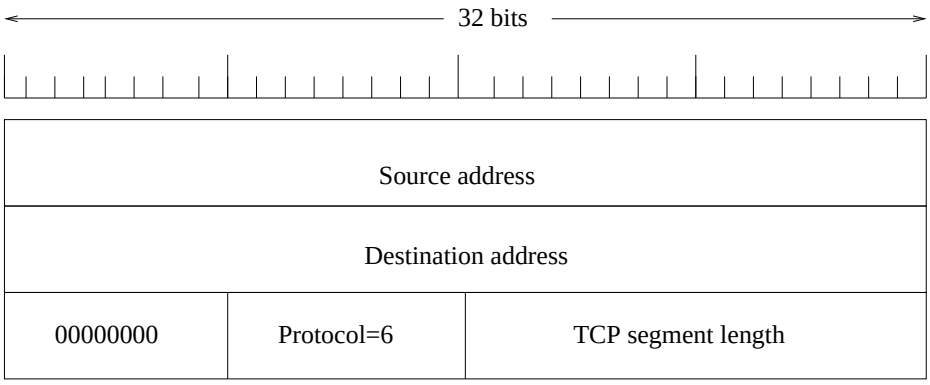


Figure 8.9: The pseudoheader included in the TCP checksum

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 8.10. 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 8.11. The common case of a client connecting to a passive server is shown with heavy lines – solid for the client, dotted for the server. The light-face

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 8.10: The states used in the finite state machine

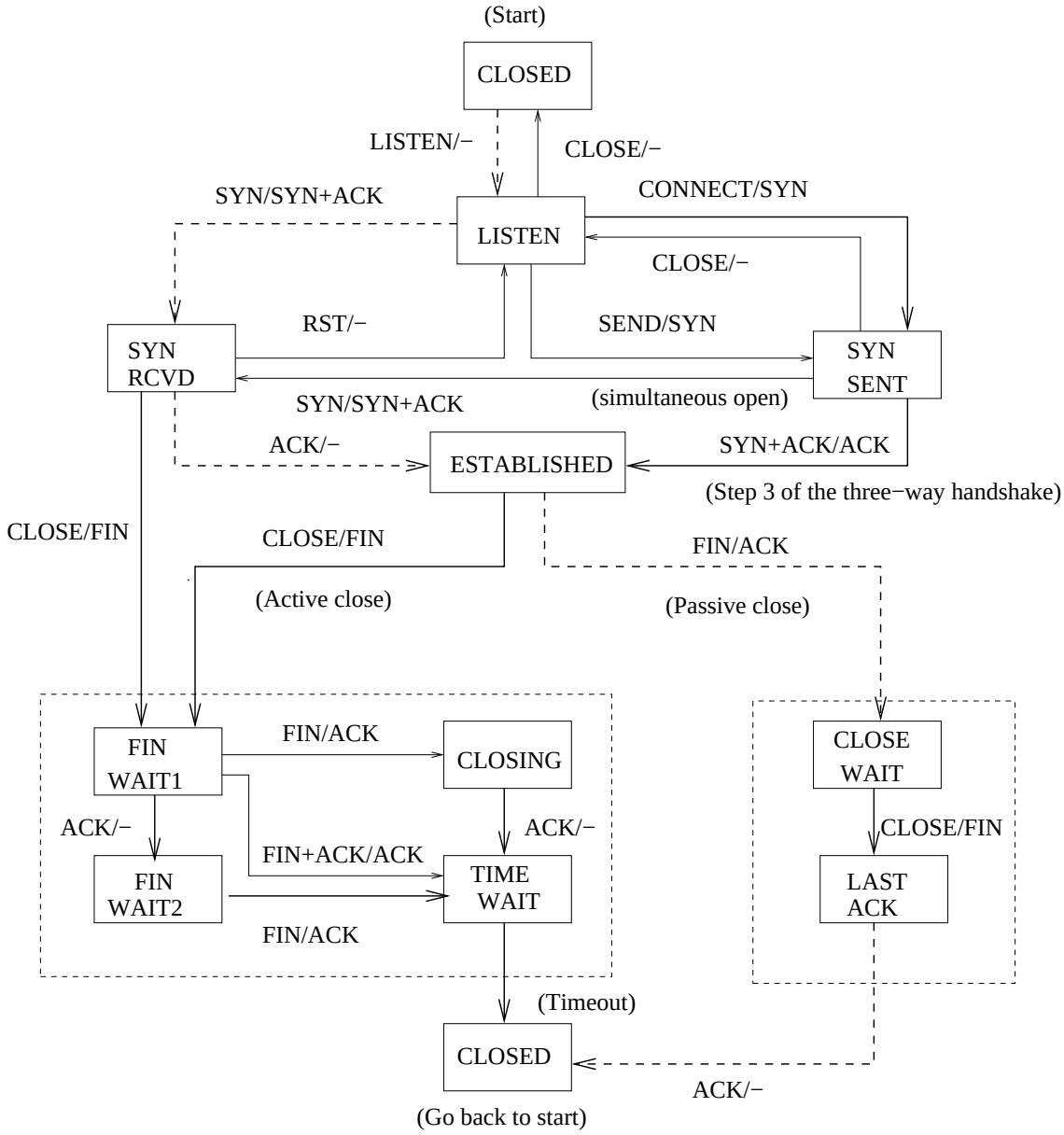


Figure 8.11: TCP connection management fine state machine

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.

8.4.2 UDP protocol

The *User Datagram Protocol* (UDP) was formally defined by IETF in RFC 768 in 1980 [12]. 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.

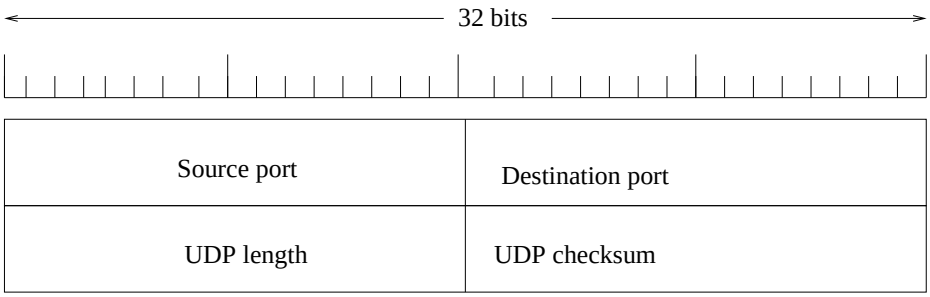


Figure 8.12: Datagram header format for UDP

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.

8.4.3 RTP protocol

The *Real-Time Transport Protocol* (RTP) is specified in RFC 1889 from 1996 [14]. 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.

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:

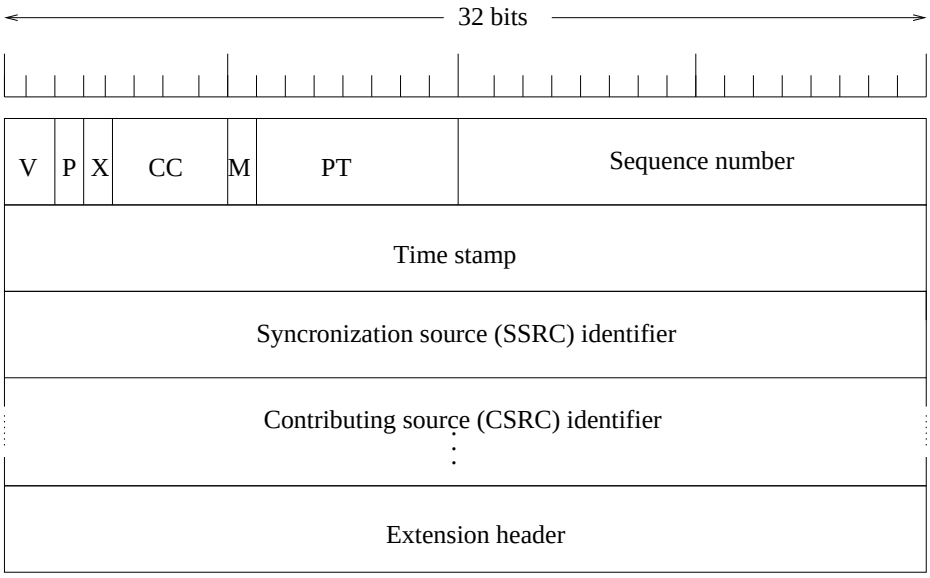


Figure 8.13: Packet header format for RTP

- 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 %).

Bibliography

[1] ATM Forum, Traffic Management Specification, Version 4.0, 1996.

[2] Blake S., Black D., Carlson M,, Davies E., Wang Z. and Weiss. W. An Architecture for Differentiated Services, IETF RFC (Informational) 2475, December 1998.

[3] Braden R., Clark D. and Shenker S., Integrated Services in the Internet Architecture, IETF RFC 1633, 1994.

[4] Deering S. and Hinden R., Internet Protocol Version 6 Specificaation, IETF RFC 1883, 1995.

[5] Fuller V., Li T., Yu J. and Varadhan K., Classless Inter-Domain Routing (CIDR), IETF RFC 1519, 1993.

[6] IETF, Internet Protocol Specification, IETF RFC 791, 1981.

[7] IETF, Transmission Control Protocol Specification, IETF RFC 793, 1981.

[8] ITU-T, ISDN service capabilities, B-ISDN service aspects, Recommendation I.211, 1993.

[9] ITU-T, Network performance objectives for IP-based services, Recommendation Y.1541, 2002.

[10] ITU-T, SG12, End-User Multimedia QoS Categories, Draft recommendaion, G.QOSrqt, 2002.

[11] McDysan D., *QoS and Traffic Management*, first edition, McGraw-Hill, 2000.

[12] Postel J., User Datagram Protcol Specification, IETF RFC 768, 1980.

[13] Rosen E., Viswanathan A. and Callon R,. Multiprotocol Label Switching Architecture, IETF RFC 3031, 2001.

[14] Schultzrinne H., Casner S., Frederick R. and Jacobson V., A Transport Protocol for Real-Time Applications, IETF RFC 1889, 1996.