CHAPTER 4

The Transport Layer

The Transport layer is a process-to-process delivery Layer. There can be multiple processes running on a particular host. While the network layer is concerned with host to host delivery of packets, the transport layer ensures delivery upto the correct application. For this the transport layer needs the address of processes running on a system. The main functions of the Transport layer are:

1. **Application Addressing/ Port Address:** A datagram in the network layer needs an IP address for transmission and delivery to the correct host. But a transport layer needs more information than this to deliver to the correct process. At the transport layer, a separate address called as the port number is required to choose among multiple processes running on the host. The destination port number is required for delivery and the source port number is required for the reply.
2. **Segmentation and Reassembly:** This Layer breaks the information, supplied by Application layer in smaller units called segments. It numbers every byte in the segment and maintains their accounting. At the destination these segments are re-combined to create the message.
3. **Connectionless Vs Connection-oriented Service:** The transport layer provides two type of services namely, Connection-oriented and Connectionless Service. In a connection oriented service, a dedicated connection is established, used and then released. All packets follow the same path. But, in a connectionless service, no connection is established. All packets are independent and they can follow different paths to reach a destination.
4. **Process to Process Flow Control:** The transport layer provides process to process flow control. The Data Link Layer also provides flow control, but it provides only from hop to hop level. The transport layer provides this service upto the process level.
5. **Process to Process Error Control:** The transport layer provides process to process Error control facility. While the Data Link Layer also provides flow control mechanism, but it provides this facility only from one hop to another. The transport layer provides this service at the process level.

4.1 PORT ADDRESSING

The port numbers are 16 bit integers, with a value between 0 and 65,535. **The client** program defines itself with a port number which is chosen randomly. This number is called **as an ephemeral port number.** The server process is also defined with a port number. But this port number cannot

be chosen randomly like the client process. **The Internet uses universal port numbers** for servers and these numbers are called as well known port numbers. Every client process knows the well known port numbers of the corresponding server process.

IANA (International Assigned Number Authority) divides the port numbers as follows:

1. **Well known port numbers:** The ports from 0 to 1023 are known as well known ports. They are controlled by IANA.
2. **Registered ports:** The ports from 1024 to 49,151 are not controlled by IANA. They can only be registered with IANA to prevent duplication.
3. **Dynamic ports:** The ports from 49,152 o 63,535 are neither controlled nor registered. They can be used by any process.

Socket Address: The combination of IP address and Port address defines a unique host/process combination. The source and destination socket addresses are required for transport layer communication. Together the combination of IP address and Port address is known as Socket Address.

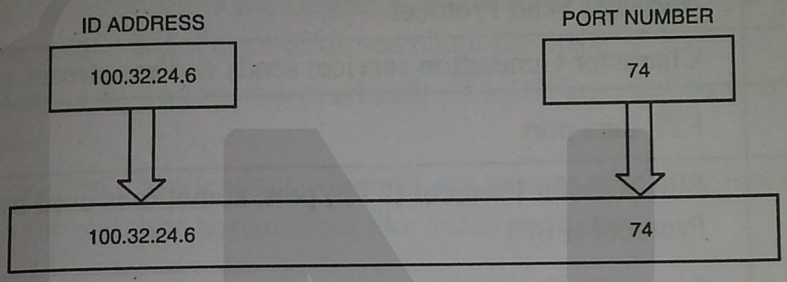


Fig 4.1. A Socket address

4.2 CONNECTIONLESS VS CONNECTION-ORIENTED SERVICE

The transport layer provides two kind of services- Connection-oriented and Connectionless Service. In a connection oriented service, a dedicated connection is established, used and then released. All packets follow the same path. But in a connectionless service, no connection is established. All packets are independent and they can follow different paths to reach a destination.

4.3 RELIABLE AND UNRELIABLE SERVICE

Transport layer provides two kind of services—reliable or unreliable. Their usage depends on the type of application. If the application layer program needs reliability then the reliable transport layer protocol is used by implementing the flow and error control at the transport layer. This kind of service will definitely have some overheads and therefore will be slow and complex. In contrast, if the application program does not need reliability and instead it uses its own flow and error control mechanisms, then an unreliable service may be used.

For example, UDP is connectionless and unreliable but TCP is connection oriented and reliable Protocol.

Table 4.1. Some Well Known Port Numbers used by UDP and TCP both

Port # / Layer	Name	Description
1	tcpmux	TCP port service multiplexer
5	rje	Remote Job Entry
7	echo	Echo service
9	discard	Null service for connection testing
11	systat	System Status service for listing connected ports
13	daytime	Sends date and time to requesting host
17	qotd	Sends quote of the day to connected host
18	msp	Message Send Protocol
19	chargen	Character Generation service; sends endless stream of characters
20	ftp-data	FTP data port
21	ftp	File Transfer Protocol (FTP) port; sometimes used by File Service Protocol (FSP)
22	ssh	Secure Shell (SSH) service
23	telnet	The Telnet service
25	smtp	Simple Mail Transfer Protocol (SMTP)
37	time	Time Protocol
43	nicname	WHOIS directory service
67	bootps	Bootstrap Protocol (BOOTP) services; also used by Dynamic Host Configuration Protocol (DHCP) services
68	bootpc	Bootstrap (BOOTP) client; also used by Dynamic Host Control Protocol (DHCP) clients
69	tftp	Trivial File Transfer Protocol (TFTP)
70	gopher	Gopher Internet document search and retrieval
105	csnet-ns	Mailbox nameserver; also used by CSO nameserver
107	rtelnet	Remote Telnet
109	pop2	Post Office Protocol version 2
110	pop3	Post Office Protocol version 3
123	ntp	Network Time Protocol (NTP)

4.4 THE USER DATAGRAM PROTOCOL (UDP)

The service provided by UDP is an unreliable service that provides no guarantees for delivery and no protection from duplication. The UDP is very basic in its operation and therefore requires very little overheads. Data sent by upper layers is divided into packets called Datagram. The main features of UDP are:

1. It provides a connectionless Unreliable Service. Each UDP datagram contains the source and destination socket address.
2. No end to end connection is established. UDP datagrams are simply injected into the network by the source. The routers forward these packets on way to the destination. Different datagrams may follow different paths in reaching the destination and may even arrive out of turn.
3. No reliability is ensured. No acknowledgements are sent for the received datagrams. If some datagrams are lost, no retransmissions are done.
4. Very little error detection is performed. If some error is detected in a datagram, it is simply dropped. The source is not even informed about the error.
5. Due to its simple operation, the overheads of UDP are very low. It works on a best effort delivery basis.
6. It is useful for simple applications which do not require reliability like routing protocols, DNS, etc. and also for real time applications like audio and video.

4.4.1 UDP datagram format

Data sent by upper layers is divided into packets called Datagram. Each UDP datagram is sent within a single IP datagram. The UDP datagram has a 8-byte header as shown in Fig. 4.2.

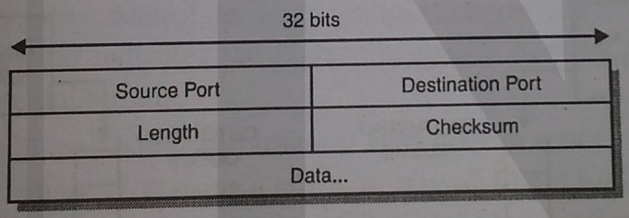


Fig. 4.2. UDP Datagram Header

(*a*) **Source Port:** It is an optional field. When meaningful it indicates the port of the sending process and may be assumed to be the port to which a reply should be addressed in the absence of any other information. If not used, a value of zero is inserted.

(*b*) **Destination Port:** It has the meaning within the context of a particular Internet Destination address.

(*c*) **Length:** It is the size in bytes of the UDP packet, including the header and data. The minimum length is 8 bytes.

(*d*) **UDP Checksum:** This is used to verify the integrity of the UDP header. The checksum is performed on a **pseudo header** consisting of information obtained from the IP header (source and destination address) as well as the UDP header. The pseudo-IP header contains the source and destination IP addresses, the protocol, and the UDP length.

4.4.2 UDP Operation

(*a*) **Connectionless Service:** UDP provides a connectionless service *i.e.* each user datagram sent by UDP is an independent datagram. These datagrams are not numbered, no connection establishment or connection release is necessary. Each datagram can follow a different path.

(*b*) **Flow control and error control:** No flow control, so the receiver can overflow with incoming messages. No error control mechanism except for checksum.

(*c*) **Encapsulation and Decapsulation:** UDP encapsulates and decapsulates messages in an IP datagram in order to send the message from one process to the other.

(*d*) **Queuing:** In UDP, queues are associated with ports as shown in Fig. 4.3 . A process starts at the client site by requesting a port number from the OS. The client process is assigned a port number from the Dynamic Port addresses list. One outgoing and another incoming queue is also created at the client side.. The queues function as long as the process is running. They are destroyed as soon as the process terminates. The client process can send message to its outgoing queue by using the source port number specified in the request. UDP removes the queue messages one by one by adding UDP header and delivers them to IP. If the outgoing queue overflows then the OS tells that client process to wait before sending the next message. When a client receives the message, UDP checks if the incoming queue has been created or not. If the queue has been created then the UDP sends the received datagram to the end of the queue. If the queue is not present then UDP will simply discard the user datagram. If the incoming queue overflows then UDP discards the user datagram and arranges to send the port unavailable message to the server. On the other hand, the mechanism to create the server is different as the server asks for the incoming and outgoing queues using its well known ports as soon as it starts running. The queues exits as long as the server is running.

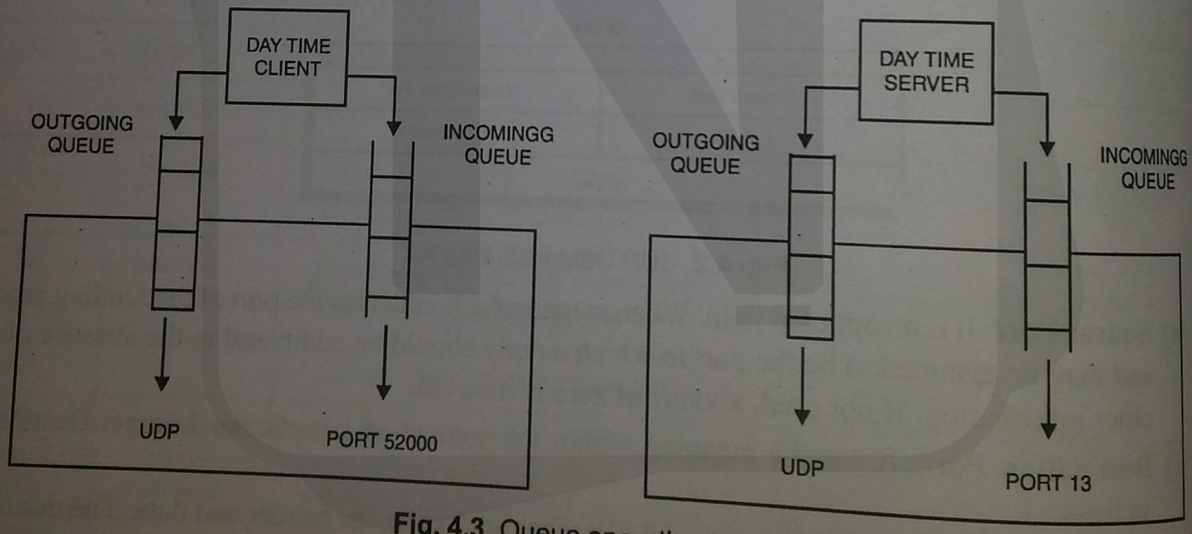


Fig. 4.3. Queue operation in UDP

4.5 TRANSMISSION CONTROL PROTOCOL (TCP)

TCP provides a Connection-Oriented Reliable Service. A virtual connection is established and all datagrams follow the same path in reaching the destination. The main features of the TCP are:

1. TCP provides reliable transmission of data in an IP environment.

2. TCP provides facility for stream transfer of data. Streaming is not possible with UDP, but the virtual connection provided by the TCP makes it possible.
3. Stream data transfer makes it possible for TCP to deliver an unstructured stream of bytes to the destination. This service is useful for applications because they do not have to chop data into blocks before handing it off to TCP. The two processes appear to be connected by a virtual tube, as shown if Fig. 4.4.

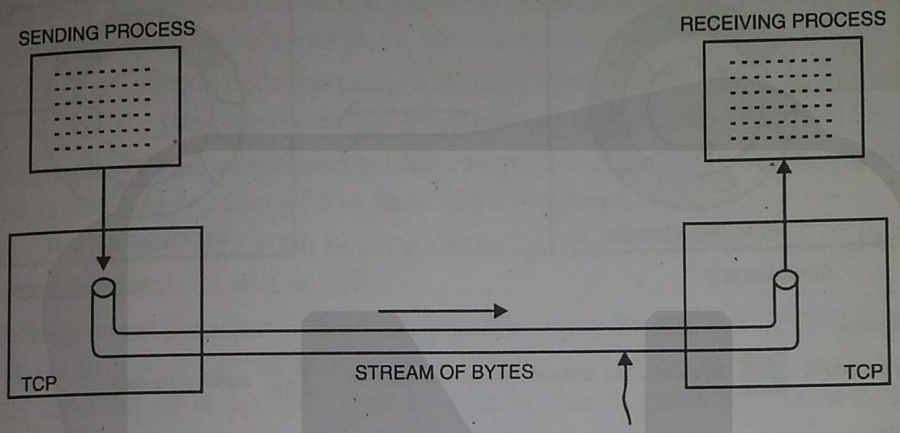


Fig 4.4. The streaming Process in TCP

4. TCP groups bytes into segments and passes them to IP for delivery.
5. TCP provides end-to-end reliable packet delivery. It does this by forwarding an acknowledgement number, which indicates the next byte to be received by the destination.
6. The reliability mechanism of TCP allows devices to deal with lost, delayed, duplicate or misread packets. Bytes not acknowledged within a specified time period are retransmitted.
7. TCP offers efficient flow control *i.e.* when sending acknowledgements back to the source, the receiving TCP process indicates the highest sequence number that it can receive without overflowing its internal buffers.
8. TCP Buffering: TCP maintains two buffers at each node: a sender buffer and a receiver buffer. Buffer can be implemented by using a circular array of 1 byte locations. Whenever a packet is to be sent, the packet may not be sent immediately due to unavailability of channel at that time. So, the packet is buffered first in the Sender Buffer. Similarly, when a packet arrives, it is first put in the Receiver Buffer before actual consumption. At the sender node, the buffer contains three kind of slots:
   a. Packets sent, but awaiting acknowledgement. These are shown as grey slots.
   b. Packets that are not yet sent. These are shown as dotted slots.
   c. Slots which are empty and ready for occupancy. These slots are shown in white.

Similarly, at the receiver side, the buffer consists of two kind of slots:

   a. Slots which are empty and ready for occupancy. These slots are shown in white.
   b. Packets that have not been read yet. These are shown as grey slots.

This is shown in Fig. 4.5 below:

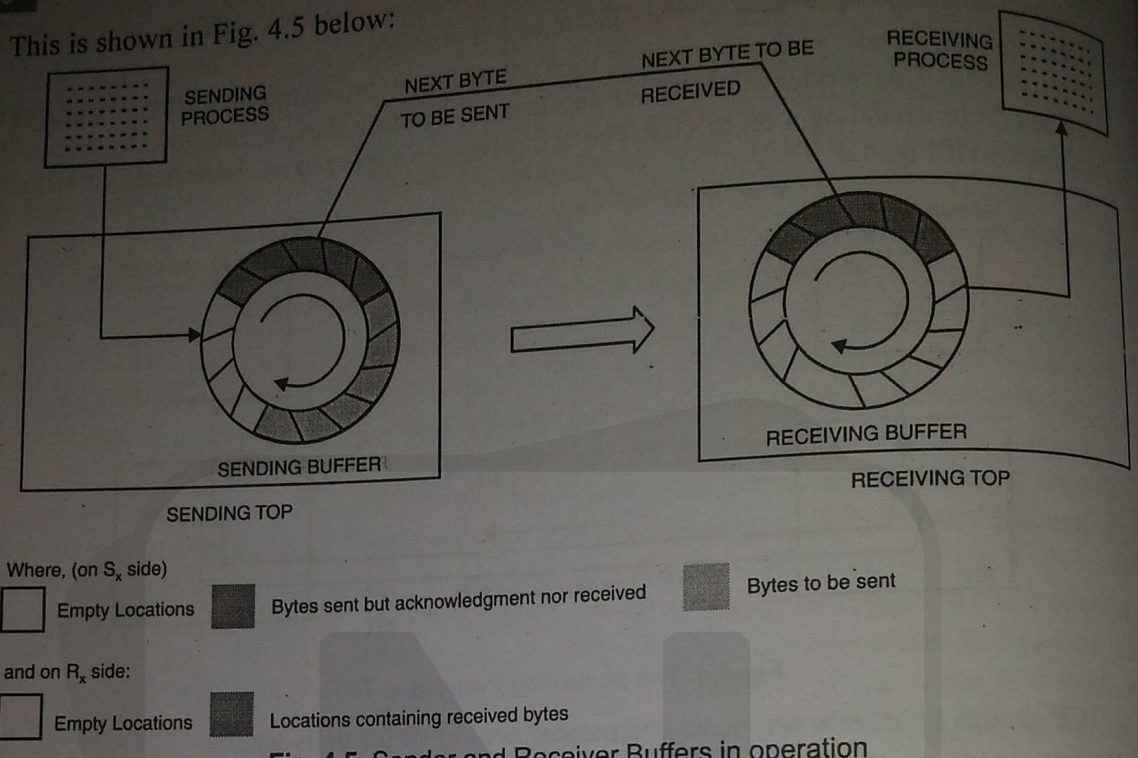


Fig. 4.5. Sender and Receiver Buffers in operation

9. The IP layer sends data in form of packets and not as a stream of bytes. However, upper layers send data to transport layer as a stream of data. At the transport layer, TCP groups a number of bytes together into a packet called as a segment. A header is added to each segment for the purpose of exercising control. The segments are encapsulated in an IP datagram and then transmitted. Process of segmentation and reassembly must be transparent to the receiving process. The segments may be received out of order, lost or corrupted when it reaches the receiving end. Therefore, some sequencing mechanism is needed. The segments can be of different sizes. Each segment can carry hundreds of bytes. The Segmentation procedure is shown in Fig. 4.6.

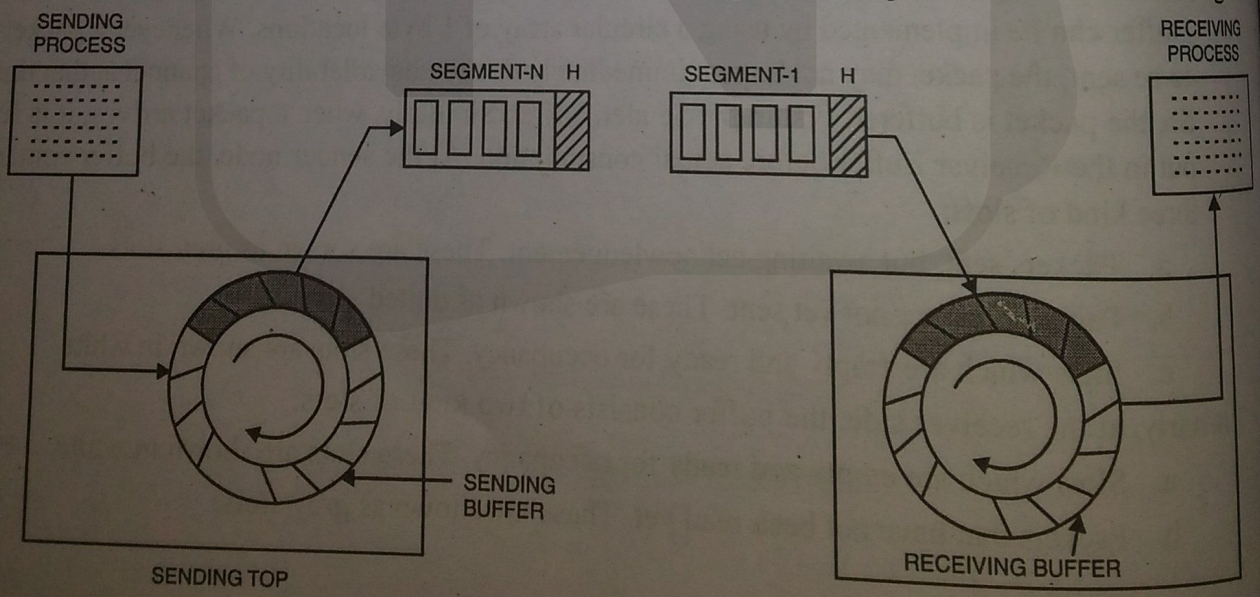


Fig 4.6. Segmentation procedure at sender and receiver

10. TCP always works in full-duplex mode i.e Simultaneous transmission in both directions.

4.6 TCP OPERATION

4.6.1 Byte, Sequence and Acknowledgement Numbering

1. **Each Byte in TCP is numbered :** The Bytes in TCP are numbered sequentially. However, the starting number is a random number and independent in each direction. This starting sequence number can be any number between 0 and 2^{32} -1. This Byte numbering is necessary to maintain the sequence of bytes, as received in the stream of data.

   Example, Suppose that a node has to transfer 1000 Bytes and the Random Starting Byte number generated is 1501. Then the Bytes will be numbered sequentially from 1501 to 2500.

2. **The Segments are also numbered by their Sequence Numbers :** The Sequence Number for a Segment is the Byte Number of first Byte in the Segment.

   Example, if a segment has 1000 Bytes with the first Byte Number as 1501, then the Sequence Number of this Segment will be 1501.

3. **Acknowledgement Number :** Received Bytes are acknowledged in TCP. The Acknowledgement Number is the Byte Number of next Byte, which a node expects to receive. If a node has correctly received all Bytes uptil Byte Number x, then it sends x+1 as its Acknowledgement Number. The acknowledgement Numbers are always cumulative.

   Example: If a host has correctly received all Bytes till Byte Numbers 1610, it sends 1611 as an acknowledgement.

4.6.2 TCP Segment Header

The TCP Segment consists of a header and optional data. The header length varies from 20to 60 Bytes. The header has the format as shown in Fig. 4.7.

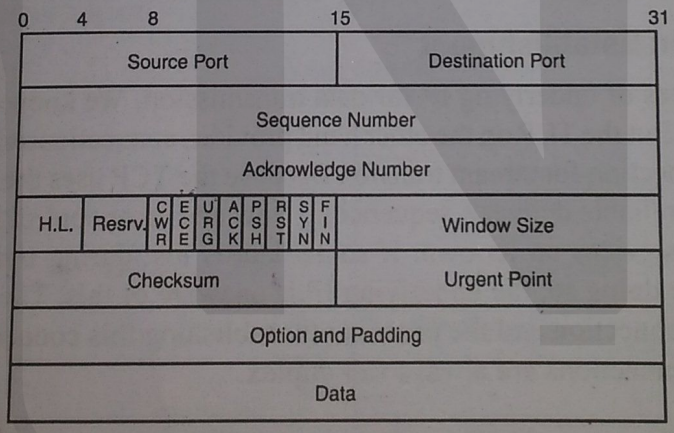


Fig. 4.7. A TCP Segment

1. **Source Port:** It is a 16-bit number identifying the application. Port assignments are used by TCP as an interface to the application layer.
2. **Destination Port:** It is a 16-bit number identifying the application for on receiving host.
3. **Sequence Number:** It is a 32-bit number identifying the current position of the first data byte in the segment within the entire byte stream for the TCP connection.
4. **Acknowledgement Number:** It is a 32-bit number identifying the next data byte, the sender expects from the receiver. **Thus, the number will be one greater than the most recently received data byte.**

5. **Header Length or Offset:** It is a 4-bit field that specifies the total TCP header length in 32-bit words. The minimum Length is 5 and maximum is 15.
6. **Reserved:** It is a 6-bit field reserved for future use. It is a 16 bit integer used by TCP for flow control in form of data transmission
7. **window size:** This number tells the sender how much data the receiver is willing to accept. The maximum value for this field would limit the window size to 65,535 bytes. The window scale option may be used to make use of even larger windows.
8. **Checksum:** The checksum field is calculated based on the contents of the TCP header and data fields. This is a 16 bit value. It is compared with the value that the receiver generates using the same computation. If the values matches, the receiver can be sure that the segment has arrived intact.
9. **Urgent Pointer:** It can be used by the TCP sender to notify the receiver of urgent data that should be processed by the receiving application as soon as possible. This 16-bit field tells the receiver the last byte, where the urgent data in the segment ends.
10. **Options:** To provide additional functionality, several optional parameters may be used between a TCP sender and receiver. Depending on the options used, the length of this field will vary in size but it cannot be larger than 40 bytes due of the size of the header length field (4 bits). The most common option is the MAXIMUM SEGMENT SIZE (MSS) option. A TCP receiver tells the TCP sender the maximum segment size, it is willing to accept through the use of this option. Other options are used for different flow control and congestion control techniques.
11. **Padding:** The options are variable in size. Therefore, it may be necessary to **pad** the TCP header with zeroes so that the segment ends on a 32-bit word boundary as defined by the standard.
12. **Data:** This variable length field carries the application data from TCP sender to the receiver.

4.6.3 TCP Connection Establishment

The TCP uses the services of underlying IP for data transmission. We know that IP is an unreliable connectionless protocol. But the TCP on the other hand provides connection-oriented Reliable Service. It provides a virtual connection for stream transfer. Because the TCP uses the underlying IP services, it means that all onus of reliable delivery, sequencing, flow and error control lies with the TCP itself. TCP provides all these services on its own. If some data is lost during transmission, the TCP re-transmits the data. While doing so, the underlying IP is unaware of this. The connection established by the TCP is a virtual connection and the process of establishing this connection is known as Three Way Handshake. TCP connections are always full duplex.

Three Way Handshaking

The TCP connection establishment process is known as **Three Way Handshaking**. When the Server is ready to establish a connection, its TCP issues a *Passive Open* request. This *Passive Open* is an indication that the Server is ready to accept requests from the clients. A Client wanting to connect to this Server issues an *Active Open* request. The sequence of steps is shown in Fig. 4.8.

1. For this the Client sends a SYN segment to the Server. This SYN segment is used to synchronise the Sequence Numbers of the Client and Server. An initial randomly generated Sequence Number (say x) is sent in this segment.

2. The Server replies with a SYN + ACK segment. This purpose of this segment is to acknowledge the SYN segment of the client and to convey the Server's Sequence Number to the Client. The Server sends its own Sequence Number (say y) and acknowledges receipt of SYN by piggybacking $x + 1$.
3. The client replies back with an acknowledgement. The Client acknowledges receipt of SYN by piggybacking $y + 1$.

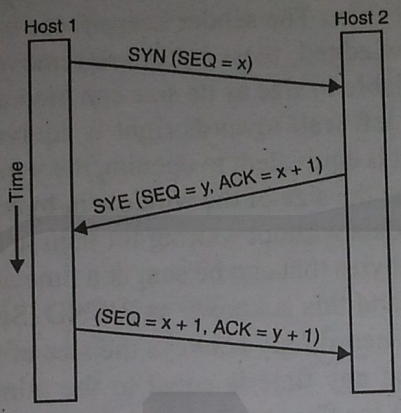


Fig 4.8. TCP 3-Way Handshake Connection Establishment

4.6.4 Data Transfer

After the connection has been established, both Server and Client can send data to each other in form of segments. Received Bytes are acknowledged by Piggybacking acknowledgements to the segments. The last Byte received is acknowledged by sending the Sequence Number of next expected Byte.

4.6.5 Connection Termination

The connection termination is again a three way Handshake process. The Client wanting to close the connection sends a FIN (Finish) segment to the server.

The Server responds by sending an acknowledgement of the FIN segment. The Server acknowlegses the receiving of FIN segment by sending a FIN + ACK Segment.

Finally, in the end, the Client replies by sending an ACK. This ACK segment acknowledges the receipt of Server FIN.

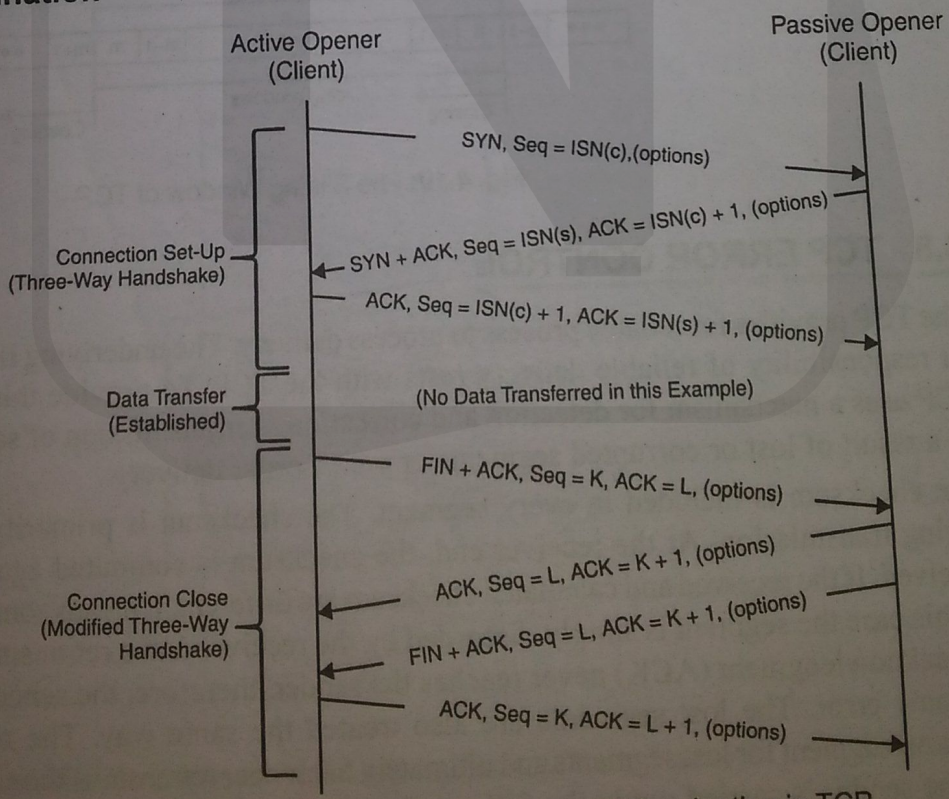


Fig 4.9. Connection Establishment and Termination in TCP

4.7 TCP FLOW CONTROL

The flow control mechanism used in TCP uses a sliding window (shown in Fig. 4.10). This window is different from the sliding window of data link layer and is of variable size. This window is maintained at the sender end. However, it is controlled by the receiver and the network. The receiver gives instructions about the load that it can receive and the sender obeys these instructions. The bytes stored in the window are the ones which have been transmitted by the sender but sender but they are yet to be acknowledged by the receiver. The sender keeps these bytes in the window until they have been acknowledged. Once acknowledged, these bytes are removed from the window to make way for new bytes. The window is variable in size as its size can increase or decrease. This window has a left and a right wall. Moving the left wall towards right is equivalent to closing the window while moving the right wall towards right is equivalent to opening the window. The number of bytes between the left and the right wall determine the size of the window in bytes. Opening the window means that the sender can send more bytes of data without waiting for their acknowledgements while closing the window means that the number of bytes that can be sent at a time are lesser. The receiver conveys the size of the window that it expects and this is known as RWND. Similarly the network also monitors the congestion in the network and accordingly conveys the size of the window. This size is known as CWND. The size of the window at any time is equal to the minimum of the RWND and CWND values. This sliding window is used for flow control as the sender can send the bytes without waiting for acknowledgements. Since, the sender can only send bytes equal to size of this window, therefore a flow control is maintained. This sender does not swamp the receiver with more bytes then it can handle.

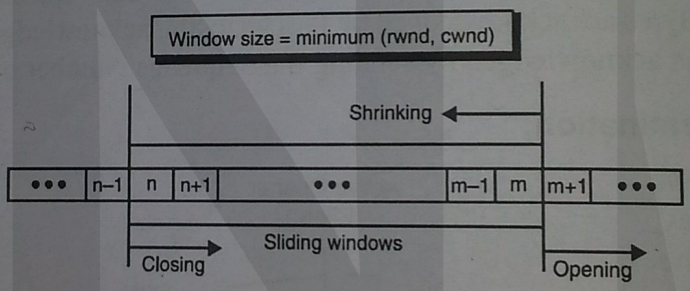


Fig. 4.10: The Sliding Window of TCP

4.8 TCP ERROR CONTROL

The TCP provides for reliable process to process delivery. The underlying IP is unreliable and therefore all responsibility of reliable delivery rests with the TCP. To provide this reliable transmission, the TCP uses a mechanism for detection and correction / retransmission of segments affected. Error can be a result of lost or corrupted segments or out of order delivery.

The checksum is included in every segment. The checksum is primarily used to detect any errors during transmission. At the receiver end, the checksum is computed again on the basis of segment received. If the received and calculated checksum are different, it means some error during transmission. In this case the segment is simply discarded by the receiver and a retransmission is required. Because the acknowledgment (ACK) never reaches the sender, therefore, the sender comes to know that there is some error. The lost segments are also treated the same way. The receiver does not send any acknowledgment for lost segments and ultimately the sender retransmits those segments. A retransmission by the sender is carried out in the following cases:

1. **Retransmission after Timeout**: whenever a segment is transmitted, the sender also starts a Retransmission Timeout. Retransmission Timeout is the maximum time interval, for which the receiver waits for an acknowledgement. This Retransmission Timeout is calculated on the basis of Round Trip Time. If no acknowledgement is received and the Timer times out, then the sender knows that a Retransmission is required and it immediately sends the segment again.

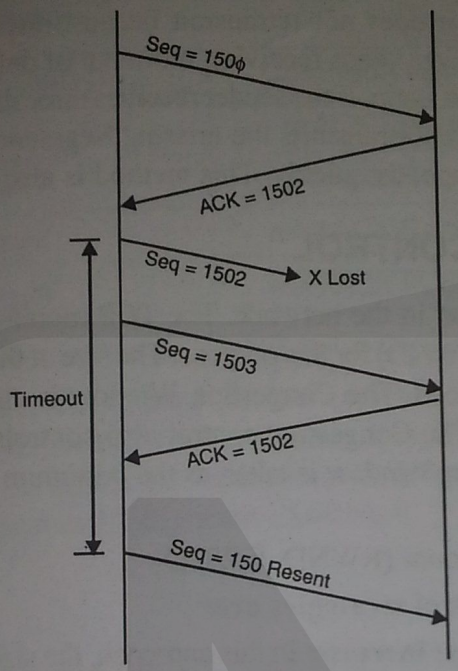


Fig. 4.11: Re-transmission after a Timout

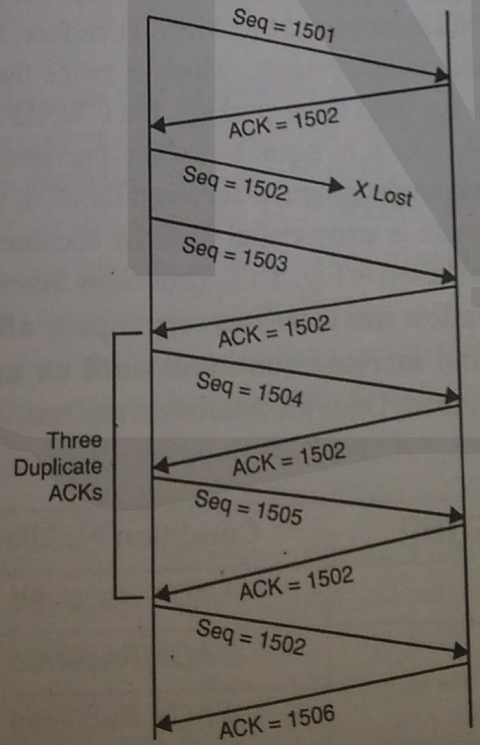


Fig. 4.12: Re-transmission after 3 duplicate ACKs

2. **Retransmission after three Duplicate Acknowledgments (Fast Retransmission):** Normally a Retransmission is carried out after a Retransmission Timeout. But sometimes this Retransmission Timeout is long and quick Retransmission is required. This is the case when out of order segments arrive at the receiver and one intermediate segment is lost. All these out of order segments cannot be saved for long due to limited buffer capacity. In this case the receiver sends an acknowledgment for the last correct segment received. The sender comes to know that one segment is lost but it does not retransmit because the Timeout hasn't occurred. In the meantime, when the receiver keeps receiving more out of delivery segments, it sends the same duplicate acknowledgment again. If the sender receives three duplicate acknowledgements before its timeout, it immediately Retransmits the missing Segment. In this case, the sender does not wait for a timeout and responds quickly. This method is also known as Fast Retransmission.

4.9 TCP CONGESTION CONTROL

TCP also keeps track of congestion in the network. The TCP monitors congestion in the network and maintains a Congestion Window (CWND) for this purpose. The size of the Congestion Window is decreased whenever some congestion is noticed. The Congestion Window is closely related to the Flow Control Mechanism, as discussed before. The Congestion control also controls the same Sliding Window at the Receiver end. The size of the sliding window is taken as the minimum of Receiver Window (RWND) or the Congestion Window (CWND).

Sliding Window Size = minimum (RWND, CWND)

Some TCP Congestion Control strategies are:

1. **Slow Start and Exponential Increase:** In this approach, the size of Congestion Window (CWND) is kept at minimum in the beginning. Initially the CWND is set as equal to 1 Maximum Segment Size (MSS) and on receipt of each acknowledgment, the CWND size is incremented by 1 MSS. Therefore, the sender sends one segment and then waits for the acknowledgment. On receipt of the acknowledgment, the CWND is increased to 2 MSS. Therefore, now the sender can send 2 segments before waiting for their acknowledgements, which is twice the initial CWND. When the sender receives acknowledgments for both these packets, the CWND is again increased by 2 MSS. This makes a total CWND of 4, which is again twice the previous size. Now the sender can send 4 consecutive segments. If acknowledgment is received for all 4, then in the next phase, CWND will increase to 8. In this way there is exponential increase because the window size is doubling after every phase. This process is shown in Fig. 4.13. Thus **Slow Start and Exponential Increase makes a cautions start but the window size increases very rapidly after successful acknowledgements. This process of exponential increase continues until an upper limit known as Slow Start Threshold (ssthresh) is reached.** Once the ssthresh is reached, TCP enters a Congestion Avoidance phase.

Phase	Initial CWND	Condition Fulfilled	New CWND
1	1	1 ACK Received	2
2	2	2 ACK Received	4
3	4	4 ACK Received	8
4	8	8 ACK Received	16

Fig. 4.11. Slow Start and Exponential Increase in CWND

2. **Congestion Avoidance (Additive Increase):** .Once the ssthresh is reached, TCP enters a Congestion Avoidance phase. In this phase, the exponential increase in CWND stops. Now the increase is more limited (additive). **In this phase when all segments of a window are acknowledged, CWND is not doubled, but it is incremented only by 1 MSS.** CWND increase by 1 means an additive increase and not exponential. Thus while the window size still increases, this increase is much more gradual. This process is shown in Fig. 4.14. Here we start showing the window k^{th} phase onwards.

Phase	Initial CWND	Condition Fulfilled	New CWND
k	m	m ACK Received	m+1
K+1	m+1	m+1 ACK Received	m+2
K+2	m+2	m+2 ACK Received	m+3
K+3	m+3	m+3 ACK Received	m+4

Fig. 4.14. Additive Increase in CWND during Congestion Avoidance

3. **Multiplicative Decrease:** This phase is reached when some retransmission is required. Congestion is the most likely reason for a retransmission. Therefore, whenever a retransmission is done, the Slow Start Threshold (ssthresh) limit is reduced to one-half of current Window Size. This process is known as Multiplicative Decrease.

4.10. TCP TAHOE

TCP Tahoe is the simplest Congestion Control algorithm of TCP. Initially the TCP starts with a Slow Start Exponential Growth Congestion Window. There is an exponential increase in CWND after every successful acknowledgment phase. Once the limit of ssthresh is reached, the Congestion Window grows additively and not exponentially, as explained earlier. But in case a retransmission is required, the following actions are taken:

a. **In case of Retransmission due to 3 Duplicate ACKs:** If 3 Duplicate ACKs are received, it is an indication that some congestion is there in the network. But the probability of congestion is still low because 3 Duplicate ACKs may be due to loss of just 1 Segment, while other segments may have reached safely. **In this case the TCP takes the following actions:**
   1. Tahoe performs a fast Retransmission of the missing segment. This missing segment is known from the 3 Duplicate ACKs received.
   2. Tahoe sets the Slow Start Threshold limit (ssthresh) to half of the current window size. Assume if the current CWND size is 256, then ssthresh limit is reset to 256/2 = 128.
   3. CWND size is reset to 1.
   4. The Slow Start phase is started again.

b. **In case of Retransmission due to Retransmission Timeout:** If the retransmission is due to a timeout and not 3 Duplicate ACKs, it is a strong indication that some congestion is there in the network. Here the probability of congestion is very high unlike 3 Duplicate ACKs. However, the TCP Tahoe performs exactly in the same way as in the case of Retransmission due to 3 Duplicate ACKs. **The TCP Tahoe does not distinguish Retransmission due to 3 Duplicate ACKs and Retransmission due to Timeout.** It follows exactly the above 4 steps:

1. Tahoe performs a fast Retransmission of the missing segment. This missing segment is known from the 3 Duplicate ACKs received.
2. Tahoe sets the Slow Start Threshold limit (ssthresh) to half of the current window size. Assume if the current CWND size is 256, then ssthresh limit is reset to 256/2 = 128.
3. CWND size is reset to 1.
4. The Slow Start phase is started again.

4.11 TCP RENO

TCP Reno is the most widely used Congestion Control algorithm of TCP. TCP Reno operation is similar to TCP Tahoe, except that there is some difference when 3 Duplicate ACKs are received. The initial start is also same. Initially the TCP starts with a Slow Start and Exponential Growth Congestion Window. There is an exponential increase in CWND after every successful acknowledgment phase. Once the upper limit of ssthresh is reached, the Congestion Window grows additively and not exponentially. In case of a retransmission, the TCP Reno behaves differently depending on the fact that the Retransmission is due to 3 Duplicate ACKs or a Timeout. These two cases are explained below.

a. **In case of Retransmission due to 3 Duplicate ACKs:** If 3 Duplicate ACKs are received, there is a strong probability that one of the segments has been lost while other segments may have reached safely. Therefore, the Reno enters Congestion Avoidance and Fast Recovery mode. This behaviour is typically different from the TCP Tahoe, which enters a Slow Start phase again. In this case the TCP Reno takes the following actions:
   1. TCP Reno performs a fast Retransmission of the missing segment. This missing segment is known from the 3 Duplicate ACKs received.
   2. TCP Reno sets the Slow Start Threshold limit (ssthresh) to half of the current window size. Assume if the current CWND size is 256, then ssthresh limit is reset to 256/2 = 128.
   3. CWND size is reset to half of its previous value i.e CWND = CWND/2, which is equal to the new ssthresh value.
   4. TCP enters the Congestion Avoidance with Fast Recovery phase. When all the Bytes of Window are acknowledged, then the CWND is incremented by 1. This method is known as Fast Recovery because CWND is not reset to 1 like in Tahoe. It assumes that only one intermediate segment is lost, while others may have reached the destination.

b. **In case of Retransmission due to Retransmission Timeout**: If the retransmission is due to a timeout instead of 3 Duplicate ACKs, it is a strong indication that there is congestion in the network. Here, the TCP Reno responds in exactly the same way as TCP Tahoe. It resets the CWND window to 1 and goes to the Slow Start phase again. The steps of this process are :
   1. Reno performs a fast Retransmission of the missing segment.
   2. It resets the Slow Start Threshold limit (ssthresh) to half of the current window size. Assume if the current CWND size is 256, then ssthresh limit is reset to 256/2 = 128.
   3. CWND size is reset to 1.
   4. The Slow Start phase is started again.

CHAPTER 5

Optical Networking

5.1 INTRODUCTION TO OPTICAL NETWORKING

Optical fibers are very thin glass cylinders or filaments which carry optical signals i.e. the signals are in the form of light.

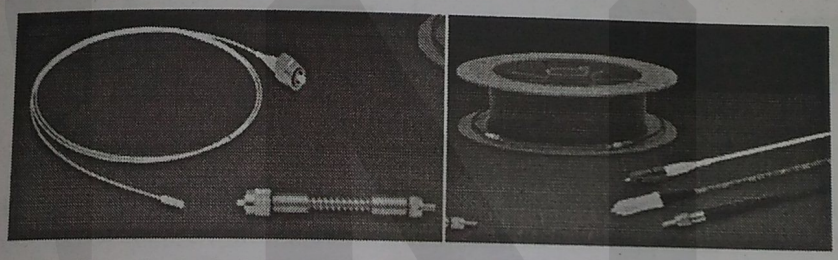


Fig. 5.1. Fiber Optic Cable and its attachments. : Image Courtesy ThorLabs[ltd]

5.1.1 What is an Optical Network?

An optical network connects multiple computers and other peripheral devices which can store and generate data in electrical form. During data communication, an optical network utilizes optical devices to generate electrical signals, amplify and recover the signals that have been transmitted over the network and route the electrical signals.

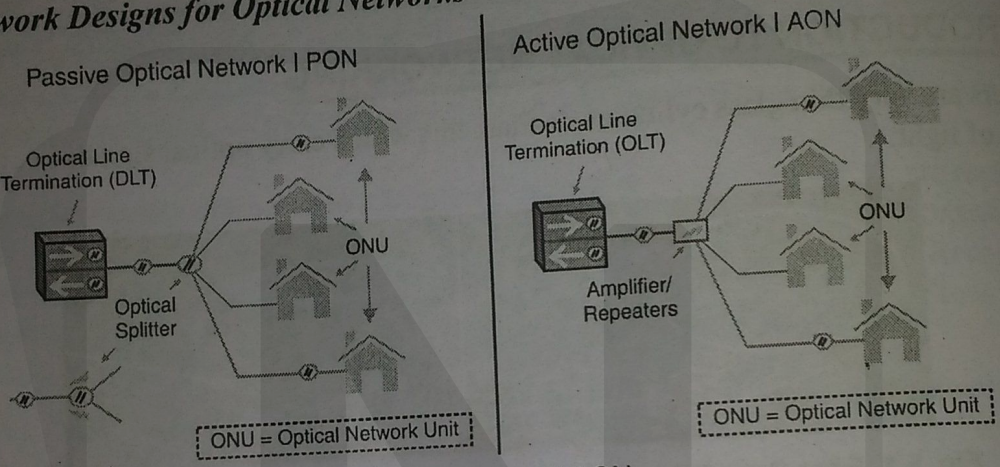
Optical networks have found widespread acceptance in the data communication field due to its high bandwidth which is upto 50 tera-bits per second. This means that it is theoretically possible to send 50×1012 bits per second using a single fiber.

S.No	Optical Fiber	Copper Wires
1	Very Expensive hence need optimized installation in network	Not costly
2	Multiple optical signals may be transmitted by one single optical fiber (WDM-wavelength division multiplexing)	Only one signal is transmitted
3	Speed of transmission of optical signal is far greater than the current data processing speed of the end terminals.	Speed of transmission matches the data processing capabilities of the end terminals.
4	Not easy to maintain	Maintenance is easy

5.1.2. Characterstics of Optical Network

- **Low signal attenuation**: As a signal propagates through fibers, the signal strength goes down at a low rate (0.2 db/km). This means that the number of optical amplifiers needed is relatively small.
- **Low signal distortion:** As a signal is sent along a fiber optic network, it degrades with respect to shape and phase. Signal regenerators are needed to restore the shape and timing. Low signal distortion means that signal regeneration is needed infrequently.
  - Low power requirement.
  - Low material usage.
  - Small space requirements.
  - Low cost.

Types of Network Designs for Optical Networks



Fg. 5.2. PON and AON

There are two types of network designs that may be used in designing fiber optic broadband connections. These may be classified as follows:

S.No.	AON	PON
1	Active Optical Network	Passive Optical Network
2	It requires electricity powered switching equipment like routers or a switch aggregator to manage signal distribution and direction to the correct end users. These switches open and close to ensure that the outgoing and incoming messages are going in the right direction.	It does not require electrically powered equipment and rather makes use of optical splitters to separate and collect optical signals that move through the network.
3	Subscribers have a dedicated fiber optic strand	Shares fiber optic strands for a portion of the network
4	Uses active components like amplifiers, repeaters, or shaping circuits to manage signal distribution	Uses optical splitters to separate and aggregate the signal
5	AON networks can cover a range to about 100 km.	Every time the signal is split two ways, half the power goes one way and half goes the other therefore PON networks have a shorter range

(Contd...)

		of coverage limited by signal strength.A PON is typically limited to fiber cable runs of up to 20 km.
6	Flexible solution suitable for businesses	Rigid solution suitable for residential
7	Higher building cost as active networks requires more fiber	PONs have a low building cost with lower maintenance costs
8	It is easier to isolate a fault in AONs	PONs also makes it difficult to isolate a failure when they occur.
9	Types Of AON- NIL	Types of PON: • EPON – Ethernet PON (Symmetrical) • GPON – Gigabit PON (Asymmetrical)

NOTE: - Hybrid systems can also be formed that utilize both active and passive networks in some FTTH (Fiber to Home) systems.

5.2 BENEFITS & DISADVANTAGES

Fiber optic cable is one of the most popular mediums for both new cabling installations and upgrades, including backbone, horizontal, and even desktop applications. Fiber offers a number of advantages over copper.

1. **Greater bandwidth**

   Fiber provides more bandwidth than copper and has standardized performance up to 10 Gbps and beyond. More bandwidth means fiber can carry more information with greater fidelity than copper wire. Keep in mind that fiber speeds are dependent on the type of cable used. Single-mode fiber offers the greatest bandwidth and no bandwidth requirements. Laser-optimized OM3 50-micron cable has an EMB of 2000 MHz/km. Laser-optimized OM4 50-micron cables has an EMB of 4700 MHz/km.

Mutlimode Fiber Types and Standards

Industry Sandards				Attenuation Typical Cable Max. (dB/km)		Bandwidth (MHz/km): Overfilled Launch (OFL)		Bandwidth (MHz/km): Effective Model Bandwidth (EMB) (also known as laser BW)
ISO/IEC	IEC	TIA	Fiber Type (μm)	850 nm	1300 nm	850 nm	1300 nm	850 nm
OM1	A1b	492-AAAA	62.5/125	3.5	1.5	200	500	—
OM2	A1a.1	492-AAAB	50/125	3.5	1.5	500	500	–
OM3	A1a.2	492-AAAC	50/25	3.5	1.5	1500	500	2000
OM4	A1a.3	492-AAAD	50/125	3.5	1.5	3500	500	4700

2. Speed and distance

Because the fiber optic signal is made of light, very little signal loss occurs during transmission, and data can move at higher speeds and greater distances. Fiber does not have the 100-meter (328-ft.) distance limitation of unshielded twisted pair copper (without a booster). Fiber distances depend on the style of cable, wavelength and network. Distances can range from 550 meters (984.2 ft.) for 10-Gbps multimode and up to 40 kilometers (24.8 mi.) for single-mode cable.

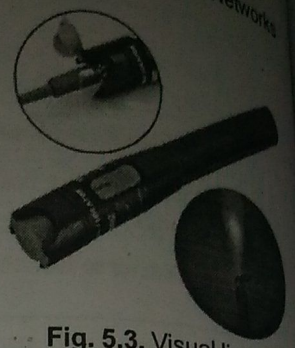


Fig. 5.3. Visual light source pen

Fiber Ethernet Standard

Netowrk	Standard	IEEE	Media	Speed	Distance
Ethernet	10BASE,-F FB, FL FP	802.3	Fiber	10 Mbps	2000 m/ 500m
Fast Ethernet	100BASE=FX	802.3u	MM Fiber	100Mbps	400 m half-duplex 2 km full-duplex
Gigabit Ethernet	1000BASE-LX	802.3z	MM, SM Fiber	1000 Mbps	550 m/2 km
	1000BASE-LX-10		SM Fiber	10 Gbps	10 km
	1000BASE-CX4				
10-Gigabit Ethernet	10GBASE-SR,-LR, -EW. -ER, -SW, -LW, -EW 10GBASE-CX4		MM, SM Fiber	10 Gbps	65m to 40 km
	10-GBASE-LX4		MM, SM Fiber	10 Gbps	400 m/10 km
	10GBASE-LR		SM Fiber	10 Gbps	10 km
	10GBASE-ER		SM Fiber	10 Gbps	40 km
	10GBASE-SR		OM3MMF	10 Gbps	26–52 m
	10GBASE-KRN	802.3aq	500-MHz MMF	10 Gbps	220 m
40-Gigabit Ethernet	40GBASE-SR4	802.3-bm	MMF	40 Gbps	100m
	40GBASE-SR4		(8) OM3 lanes	40 Gbps	125 m
	—		SM Fiber	40 Gbps	10 km
	40GBASE-FR		SM Fiber	40 Gbps	2 km
	40GBASE-LR4		SMF	40 Gbps	10 km
	40GBASE-FR		SMF	40 Gbps	2 km
	100GBASE-SR10		(10) OM3 MM pairs	100 Gbps	100 m
100-Gigabit Ethernet	—	(10) OM4 MM	100 Gbps pairs	150 m	
	100GBPS-LR4		(4) SMF lanes	100 Gbps	10 km
	100GBASE-ER4		(4) SMF lanes	100 Gbps	40 km

3. **Security :** It doesn't radiate signals and is extremely difficult to tap. If the cable is tapped, it's very easy to monitor because the cable leaks light, causing the entire system to fail. If an attempt is made to break the physical security of your fiber system, you'll know it. Fiber networks also enable you to put all your electronics and hardware in one central location, instead of having wiring closets with equipment throughout the building.

4. **Immunity and reliability :** Fiber provides extremely reliable data transmission. It's completely immune to many environmental factors that affect copper cable. The core is made of glass, which is an insulator, so no electric current can flow through. It's immune to electrometric interference and radio-frequency interference (EMI/RFI), crosstalk, impedance problems, and more. You can run fiber cable next to industrial equipment without worry. Fiber is also less susceptible to temperature fluctuations than copper and can be submerged in water.

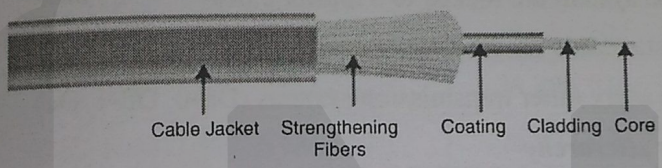


Fig. 5.4. Structure of Optical Cable

5. **Design :** Fiber is lightweight, thin, and more durable than copper cable. To get higher speeds using copper cable, you need to use a higher grade of cable, which typically have larger outside diameters, weight more, and take up more space in cable trays. With fiber cable, there is very little difference in diameter or weight. Additionally, fiber optic cable has pulling specifications that are up to 10 times greater than copper cable, depending on the specific cable. Its small size makes it easier to handle, and it takes up much less space in cabling ducts. And, fiber is easier to test than copper cable.

6. **Migration :** The proliferation and lower costs of media converters are making copper to fiber migration much easier. The converters provide seamless links and enable the use of existing hardware. Fiber can be incorporated into network in planned upgrades. In addition, with the advent of 12- and 24-strand MPO cassettes, cables, and hardware, planning for future 40- and 100-GbE networks is easier.

7. **Field termination :** Although fiber is still more difficult to terminate than copper, advancements in fiber tools have made terminating and using fiber in the field easier. Quick fusion splicers enables with auto-alignments enable fast splicing in the field. Auto-aligning pins ensure accuracy. And the use of pig-tails and pre-terminated cable make field connections quick and easy.

8. **Cost :** The cost for fiber cable, components, and hardware has steadily decreased. Overall, fiber cable is more expensive than copper cable in the short run, but it may be less expensive in the long run. Fiber typically costs less to maintain, has less downtime, and requires less networking hardware. In addition, advances in field termination technology have reduced the cost of fiber installation as well.

5.2.1. Drawbacks of Optical Networks

1. Higher initial cost in installation

2. High cost of connector and interfacing requires specialized and sophisticated tools & maintenance and repairing.

5.3 SONET ARCHITECTURE

- In the 1980's Bellcore developed the Synchronous Optical Network (SONET) standard. Digital transmission standards for fiber-optic cable
- Independently developed in USA & Europe
- SONET(Synchronous Optical Network) by ANSI
- SDH(Synchronous Digital Hierarchy) by ITU-T
- Synchronous network using synchronous TDM multiplexing
- All clocks in the system are locked to a master clock
- It contains the standards for fiber-optic equipments
- Very flexible to carry other transmission systems (DS-0, DS-1, etc)
- ***PCM T-Carrier Hierarchy***

Digital Signal Designation	Line rate	Channels (DS0s)	Line
DS0	64 kbit/s	1	
DS1	1.544 Mbit/s	24	T1
DS1C	3.152 Mbit/s	48	T1C
DS2	6.312 Mbit/s	96	T2
DS3	44.736 Mbit/s	672	T3
DS4	274.176 Mbit/s	4032	T4
DS5	400.352 Mbit/s	5760	T5

- The system that was used before 1980 was called Digital carrier systems
  - It worked by establishing a hierarchy of digital signals that the telephone network uses.
  - In it Trunks and access links were organized in **DS (digital signal) hierarchy**
  - The main drawbacks of Digital Carrier System were that the data rates were not multiples of each other.
- Before SONET **ISDN** and **BISDN** were also used for optimizing the network.

The SONET architecture is as follows:

- Architecture of a SONET system: signals, devices, and connections
- Signals: SONET(SDH) defines a hierarchy of electrical signaling levels called STSs(Synchronous Transport Signals, (STMs)). Corresponding optical signals are called OCs(Optical Carriers)

Table 5.1. Sonets Transmission Rate

SONET Optical Carrier Level	SONET Frame Formal	SDH Level and Frame	Payload Bandwidth (kbit/s)	Line Rate (Kbit/s)
OC-1	STS-1	STM-0	48,960	51,840
OC-3	STS-3	STM-1	150,336	155,520
OC-12	STS-12	STM-4	601,344	622,080
OC-24	STS-24	STM-8	1,202,688	1,244,160
OC-48	STS-48	STM-16	2,405,376	2,488,320
OC-96	STS-96	STM-32	4,810,752	4,976,640
OC-192	STS-192	STM-64	9,621,504	9,953,280
OC-768	STS-768	STM-256	38,486,016	39,813,120
OC-1536	STS-1536	STM-512	76,972,032	79,626,120
OC-3072	STS-3072	STM-1024	153,944,064	159,252,240

- SONET devices: STS multiplexer/ demultiplexer, regenerator, add/drop multiplexer, terminals

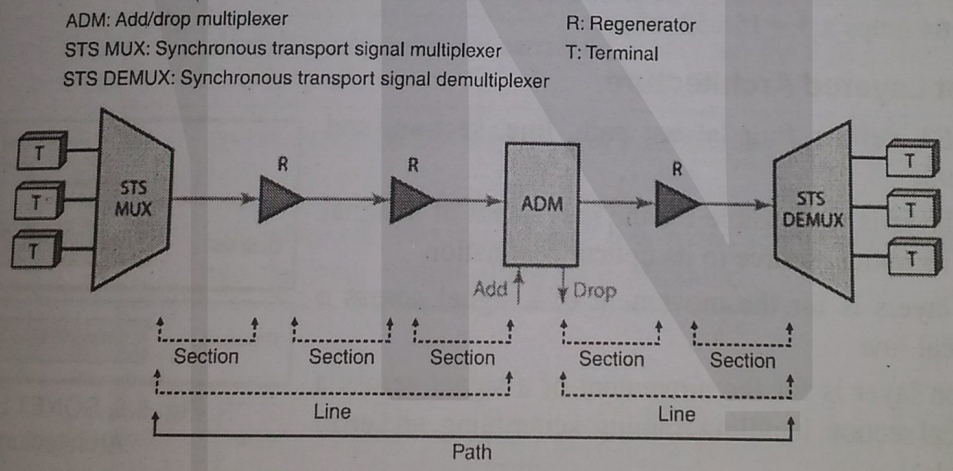


Fig. 5.5: SONET Architecture

- Connections: SONET devices are connected using *sections*, *lines*, and *paths*
- *Section:* optical link connecting two neighbor devices: mux to mux, mux to regenerator, or regenerator to regenerator
- *Lines:* portion of network between two multiplexers
- *Paths:* end-to-end portion of the network between two STS multiplexer

SONET Add-Drop Multiplexor

- Complex system of pointers locates channels within a payload, payloads within a frame.
- Facilitates rapid access, removal and insertion of data without regenerating SPE.

SONET Pointers frame SPE Payload which can slip or "float" within STS frames
- POH pointer describes payload channels
- LOH pointer describes entire payload
- Each SONET node recalculates pointers to determine exact payload starting point

Payload can "straddle" more than one STS frame
- Pointers in LOH indicate start of payload in each frame

Transportation of T-Carrier Services within SONET STS-1 frame (51.84 mbps)
- STS-1 designed for DS3 (44.736 mbps) tributary
- SONET Virtual Tributaries (VT) carry lower data rate signals, e.g. DS1, DS2, E1
- SONET VT resembles STS frame structure
  - VT carries its own Transport and PO
- VTs may be locked or float within STS frame
- VTs byte interleaved within respective SPE
- Uses STS-1 SPE

High data rate or broadband tributary signals may require data rates greater than STS-1 SPE capacity (51.84 mbps)
- High capacity STS-N frames assembled from multiple byte interleaved STS-1 frames
- High data rate signals mapped directly into STS-Nc (concatenated frames)
- 3 STS-1 frames concatenated form STS-3c or OC-3c
- 51.84 mbps x 3 = 155.52 mbps

5.3.1. Sonet Layered Architecture

- SONET defines four layers: path, line, section, and photonic
- Path layer is responsible for the movement of a signal from its optical source to its optical destination
- Line layers is for the movement of a signal across a physical line
- Section layer is for the movement of a signal across a physical section, handling framing, scrambling, and error control
- Photonic layer corresponds to the physical layer of OSI model

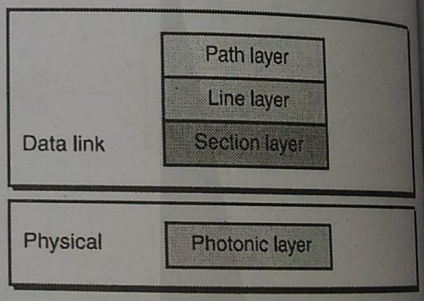


Fig. 5.6. SONET Layerad Architecture

Layer	Functions
Photonic	This layer is responsible for conversion between STS signal and OC signals
Section	This layer implements Framing, Scrambling, Error Monitoring, Section Maintenance
Line	Synchronization, Multiplexing, Error monitoring, Line Maintenance, Protection Switching are the functions of Line Layer
Path	Path Layer maps the signals into a format required by the line layer. It also reads, interprets, and modifies the path overhead for performance monitoring and automatic protection switching.

Fig. 5.7. Summarized Functions of Sonet Layers

Device-Layer Relationship in SONET

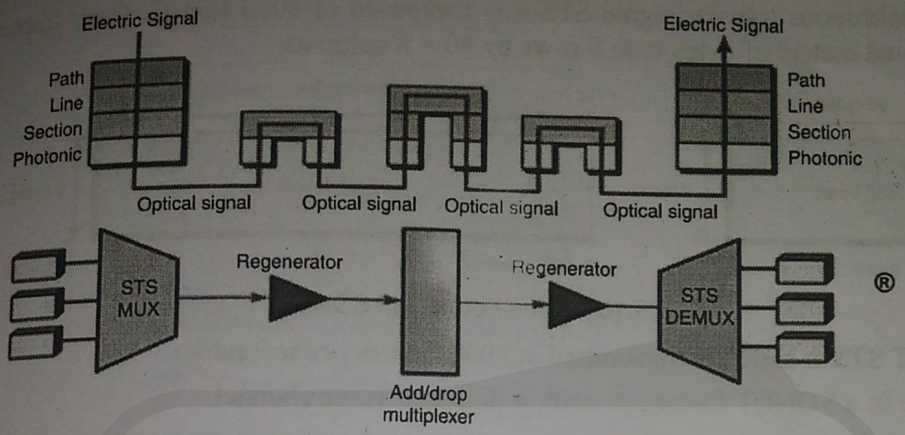
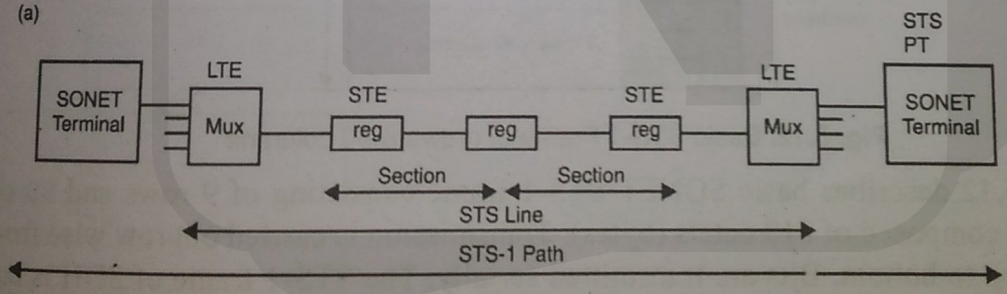


Fig. 5.8. Device Layer relationship in SONET

Element	Functions
Terminal multiplexer	Acts as a concentrator of DS1s as well as other tributary signals.
Regenerator	Acts as amplifier to enhance the signal level.
Add/Drop Multiplexer (ADM)	Can multiplex various inputs into an OC-N signal. At an add/drop site, only those signals that need to be accessed are dropped or inserted.
Wideband Digital Cross-Connects	Accepts various optical carrier rates, accesses the STS-1 signals, and switches at this level.
Broadband Digital Cross-Connect	Accesses the STS-1 signals, and switches at this level
Digital Loop Carrier	Considered as a concentrator of low-speed services before they are brought into the local central office for distribution.

(a)



STE: Section Terminating Equipment, e.g. a repeater
LTE: Line Terminating Equipment, e.g. a STS-1 to STS-3 multiplexer
PTE: Path Terminating Equipment, e.g. an STS-1 multiplexer

(b)

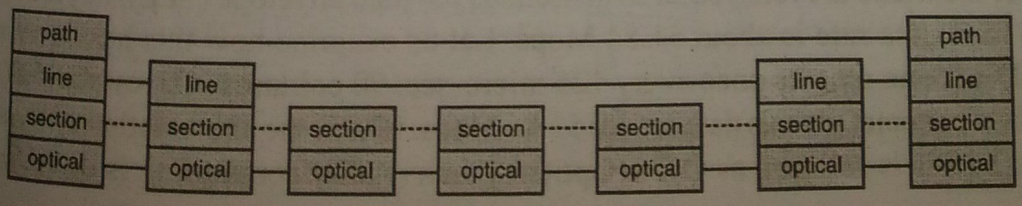


Fig. 5.9. Layer hierarchy in SONET

5.4 SONET FRAME FORMAT

- Each synchronous transfer signal STS-n is composed of 8000 frames. Each frame is a two-dimensional matrix of bytes with 9 rows by 90 × n columns.

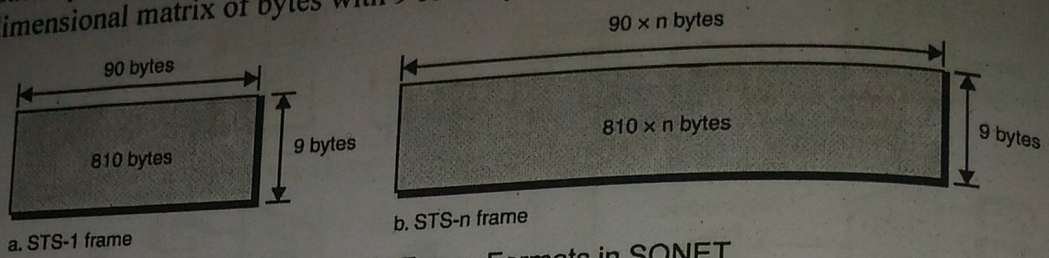


Fig. 5.10. Frame Formats in SONET

- A SONET STS-n signal is transmitted at 8000 frames per second
- Each byte in a SONET frame can carry a digitized voice channel

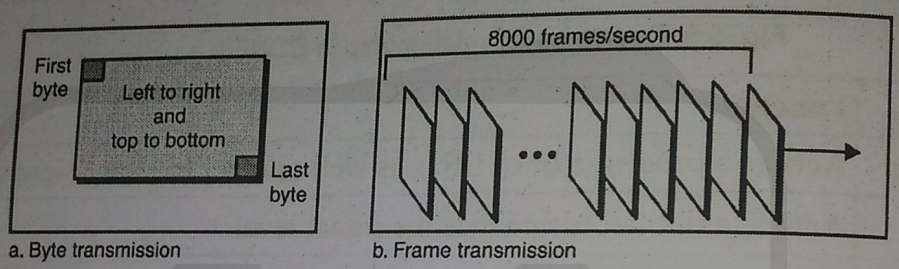


Fig. 5.11. Byte transmission & Frame transmission

- In SONET, the data rate of an STS-n signal is *n* times the data rate of an STS-1 signal
- In SONET, the duration of any frame is 125 is

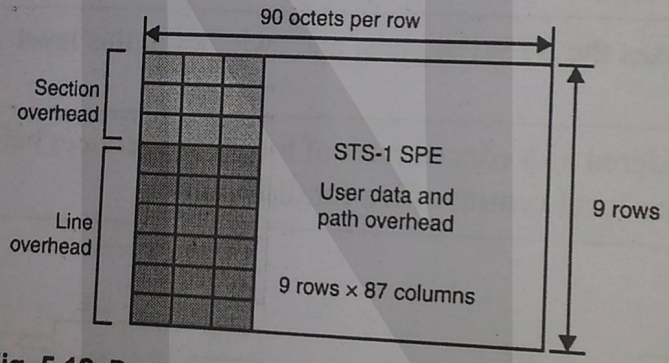


Fig. 5.12. Basic STS-1 Frame of grows & 90 columns

The figure 5.12 describes basic SONET STS-1 frame consisting of 9 rows and 90 columns. SONET frame is composed of 810 octets (bytes). Transmission is carried out row wise from left to right and from top to bottom. Bits are transmitted serially. The STS-1 frame of SDH is composed of section overhead, transport overhead, payload overhead and data part. The frame starts with fixed A1/A2 bit pattern of 0xf628 used for bit/octet synchronization. SONET/SDH is referred as octet synchronous. The first three columns of SONET frame is referred as transport overhead. The next 87 columns of the frame are referred as Synchronous payload envelope (SPE). Payload overhead is part of SPE. STS-1 data rate is about 51.84 Mbps. Let us examine how this has been achieved. Every SONET/SDH frame repeats once every 125 micro-sec. 90 columns in 9 rows and 8000 times per second and 8 bits per octet give us data rate of 51.84 Mbps. STS is the abbreviation of Synchronous Transport Signal. STS-1 is referred as OC-1(Optical Carrier) after scrambling is done on STS-1.

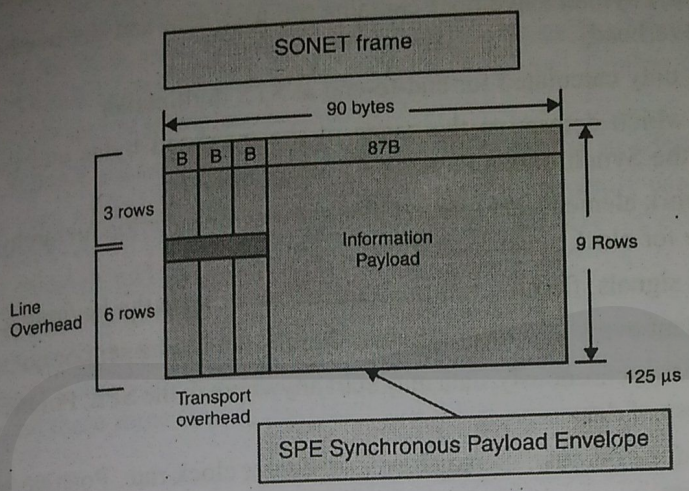


Fig. 5.13. Payload Envelope

The following Table details the different SONET overhead information:

Overhead	Function
Section Overhead	Performance monitoring (STS-N signal), Local orderwire, Data communication channels to carry information for OAM&P, Framing
Line Overhead	Locating the SPE in the frame, Multiplexing or concatenating signals, Performance monitoring, Automatic protection switching, and Line maintenance.
STS Path Overhead	Performance monitoring of the STS SPE, Signal label (the content of the STS SPE, including status of mapped payloads), Path status, Path trace
VT Path Overhead	Provides communication between the point of creation of a VT SPE and its point of disassembly, error checking, signal label, and path status.

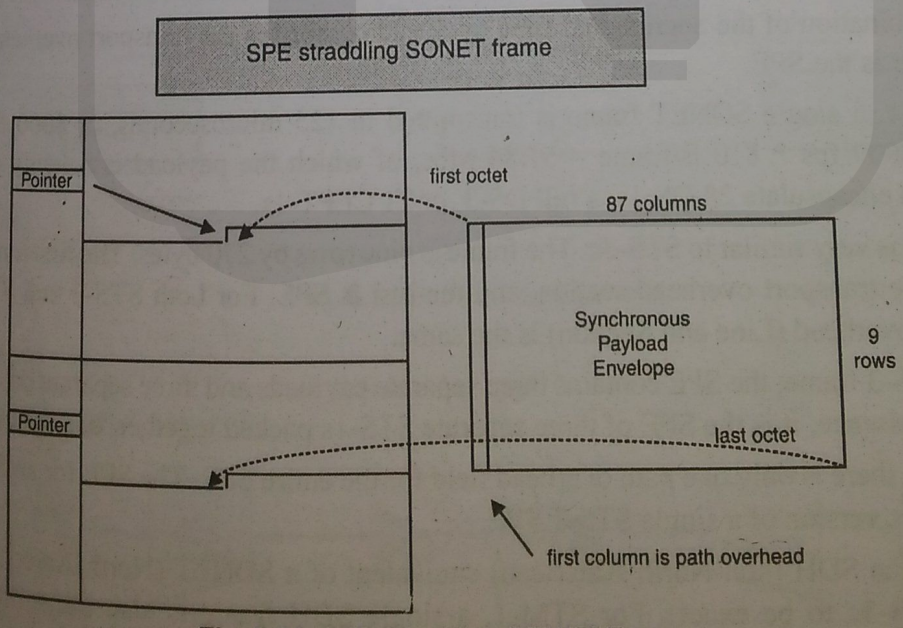


Fig. 5.14. SPE Straddling SONET Frame

- SPE(Synchronous Payload Envelope) contains the user data and the overhead related to the user data (path overhead)
- Path overhead is only calculated for end-to-end at STS multiplexer

We list the factors, which determine the usage of the overhead bytes and the ways the input signals are mapped into the Synchronous Payload Envelope (SPE).

- All SONET network elements are integrated into a synchronization hierarchy. There is no need to send preamble for clock synchronization.
- Similar to digital signals, framing bits are required to indicate the beginning of a frame.
- STS-N frame is sent every 125 msec whether there is data to be sent or not.
- Since data arrives asynchronously, data may start anywhere in the SPE. Pointer is used to indicate the starting address of data.
- The input data and the output data may have a different clock rate. Positive/negative stuffing is used for adjustment.
- SONET functions map closely into the physical layer of the OSI seven-layer stack. Error checking is not required in this layer. However, error checking is done in SONET for equipment monitoring and automatic protection switching.
- SONET integrates OAM&P in the network; overhead channels are established for administrative functions and communication.
- SONET has a fix size SPE. In order to accommodate different signal rates, bit stuffing is needed to map various signal into the SPE.
- A standard STS-1 frame is nine rows by 90 bytes. The first three bytes of each row represent the Section and Line overhead. These overhead bits comprise framing bits and pointers to different parts of the SONET frame.
- There is one column of bytes in the payload that represents the STS path overhead. This column frequently "floats" throughout the frame. Its location in the frame is determined by a pointer in the Section and Line overhead.
- The combination of the Section and Line overhead comprises the transport overhead, and the remainder is the SPE.
- For STS-1, a single SONET frame is transmitted in 125 microseconds, or 8000 frames per second. 8000 fps * 810 B/frame = 51.84 Mbs, of which the payload is roughly 49.5 Mbs, enough to encapsulate 28 DS-1s, a full DS-3, or 21 CEPT-1s.
- An STS-3 is very similar to STS-3c. The frame is nine rows by 270 bytes. The first nine columns contain the transport overhead section, and the rest is SPE. For both STS-3 and STS-3c, the transport overhead (Line and Section) is the same.
- For an STS-3 frame, the SPE contains three separate payloads and three separate path overhead fields. In essence, it is the SPE of three separate STS-1s packed together, one after another.
- In STS-3c, there is only one path overhead field for the entire SPE. The SPE for an STS-3c is a much larger version of a single STS-1 SPE.
- STM-1 is the SDH (non-North American) equivalent of a SONET (North American) STS-3 frame (STS-3c to be exact). For STM-1, a single SDH frame is also transmitted in 125

microseconds, but the frame is 270 bytes long by nine rows wide, or 155.52 Mbs, with a nine-byte header for each row. The nine-byte header contains the Multiplexer and Regenerator overhead. This is nearly identical to the STS-3c Line and Section overhead. In fact, this is where the SDH and SONET standards differ.

- SDH and SONET are not directly compatible, but only differ in a few overhead bytes.
- SONET is very widely deployed in telco space, and is frequently used in a ring configuration. Devices such as ADMs sit on the ring and behave as LTE-layer devices; these devices strip off individual channels and pass them along to the PTE layer.
- **Interleaving in SONET/SDH**

STS-3 frame is formed using three STS-1 frames with the help of interleaving technique. The interleaving is octet type i.e. A1 octet from 1st,2nd and 3rd STS-1 frame is taken first then A2 octet from all these three frames are taken and transmitted.

Scrambling IN SONET

SONET uses NRZ coding. 1 = Light On, 0 = Light Off.

Too many 1's or 0's result in Loss of bit clocking information

All bytes (except some overhead bytes) are scrambled

Polynomial $1 + x^6 + x^7$ with a seed of 1111111 is used to generate a pseudorandom sequence, which is XOR'ed to incoming bits.

1111 1110-0000 0100-0001 1000-0101 0001-1110 0100- 0101 1001-1101 0100-1111 1010-0001 1100-0100 1001- 1011 0101-1011 1101-1000 1101-0010 1110-1110 0110- 0101 010

If user data is identical to (or complement of) the pseudorandom sequence, the result will be all 0's or 1's.

5.5 SONET CONFIGURATION

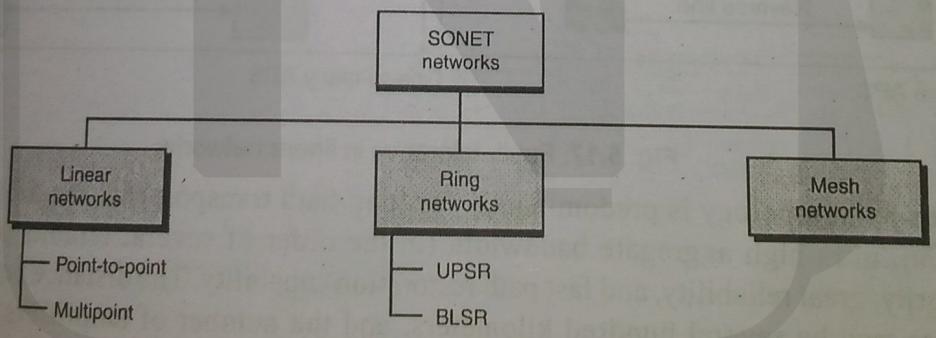
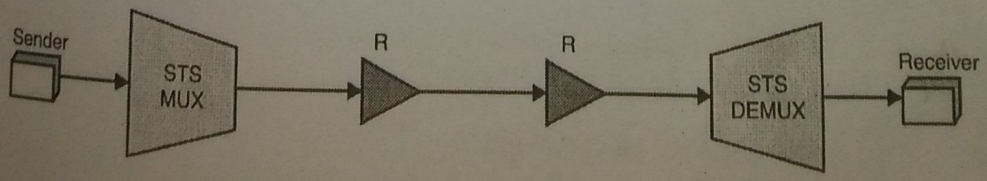


Fig. 5.15. SONET Configuration

5.5.1. SONET Topologies

- **Point-to-point network**



• **Multipoint network**

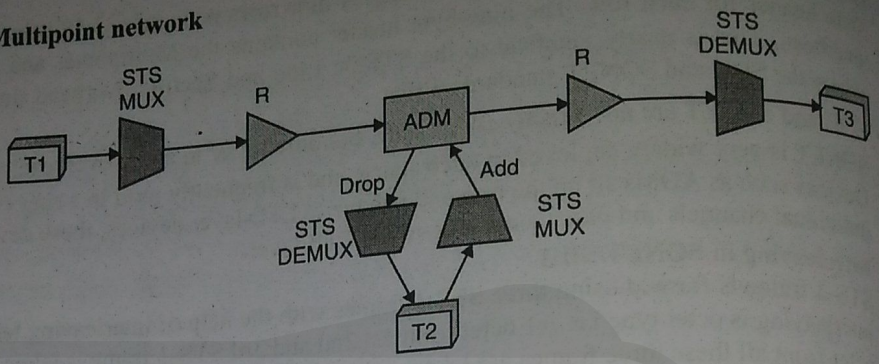


Fig. 5.16. SONET Topologies

• **To create protection against failure in linear networks**

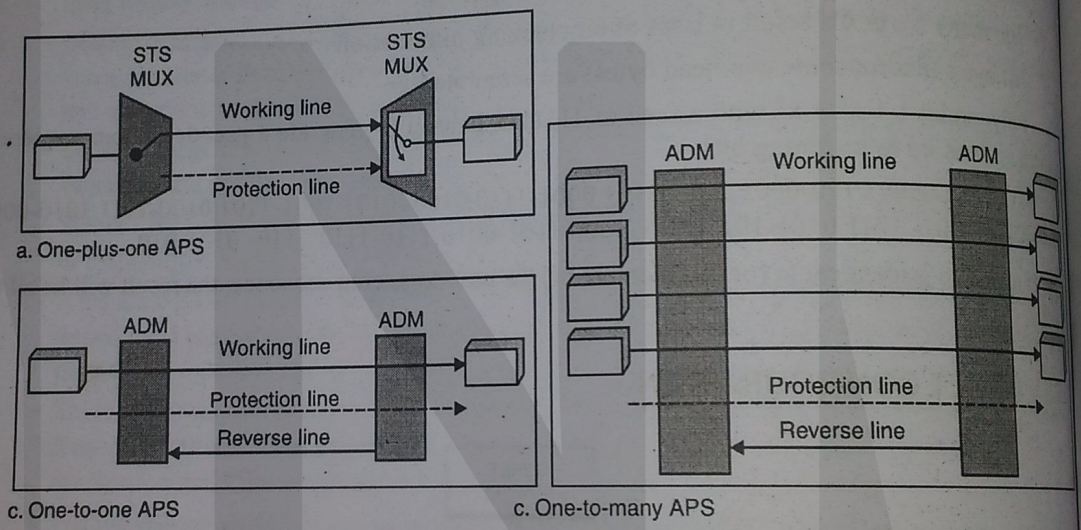


Fig. 5.17. Fault tolerance in linear networks

Point-to-point topology is predominantly for long-haul transport that requires ultrahigh spee (10-40 Gb/s), ultra high aggregate bandwidth (in the order of several terabits per second), hig signal integrity, great reliability, and fast path restoration capability. The distance between transmitte and receiver may be several hundred kilometers, and the number of amplifiers between the tw end points is typically less than 10 (as determined by power loss and signal distortion). Point-t point with add-drop multiplexing enables the system to drop and add channels along its path Number of channels, channel spacing, type of fiber, signal modulation method, and compone type selection are all important parameters in the calculation of the power budget.

- **Ring Network: UPSR (Unidirectional Path Switching Ring)**

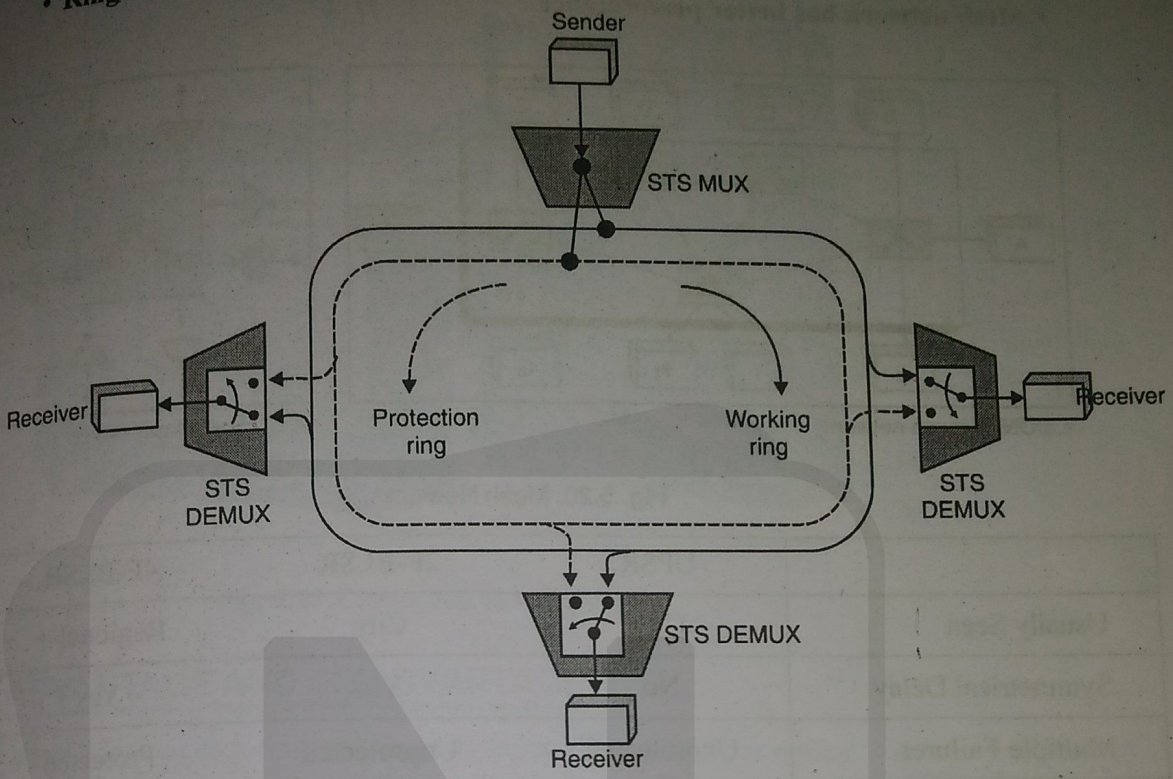


Fig. 5.18. UPSR ring Network

- **Ring Network: BLSR Bidirectional Line Switching Ring**

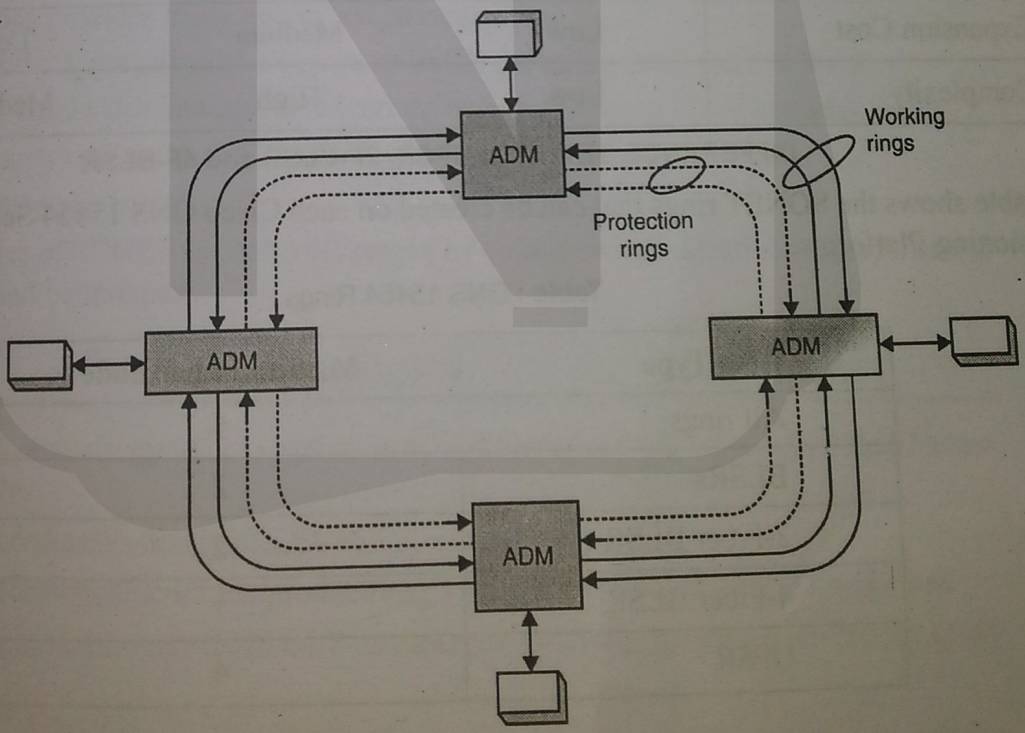
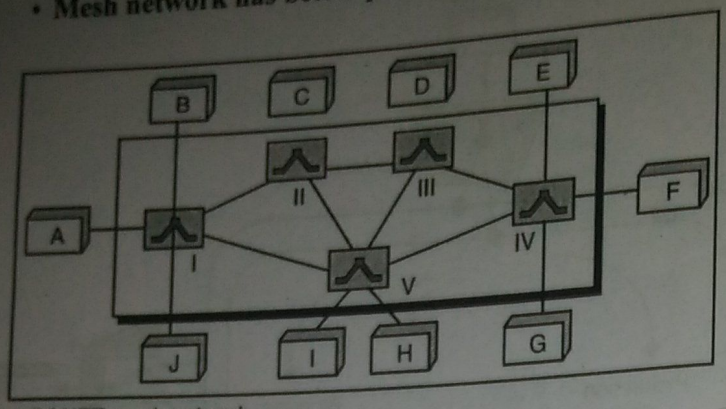
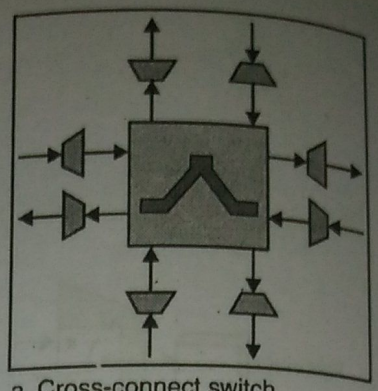


Fig. 5.19. BLSR Network

- Ring network has the lack of scalability
- Mesh network has better performance



a. SONET mesh network



a. Cross-connect switch

Fig. 5.20. Mesh Network

	UPSR	2F-BLSR	4F-BLSR
Usually Seen	City	City	Regional +
Symmetrical Delay	No	Yes	Yes
Multiple Failures	Unprotected	Unprotected	Protected
Bandwidth Efficiency	Medium	Medium	High
Intial Cost	Medium	Medium	High
Expansion Cost	Low	Medium	Low
Complexity	Low	High	Medium

DIFFERENCE BETWEEN UPSR, 2F-BLSR and 4F-BLSR

Table shows the SONET rings that can be created on each Cisco ONS 15454 Series Multiservice Provisioning Platforms.

Table : ONS 15454 Rings

Ring Type	Maximum per node
All rings	5
BLSRs	2
2-Fiber BLSR	2
4-Fiber BLSR	1
UPSR	4

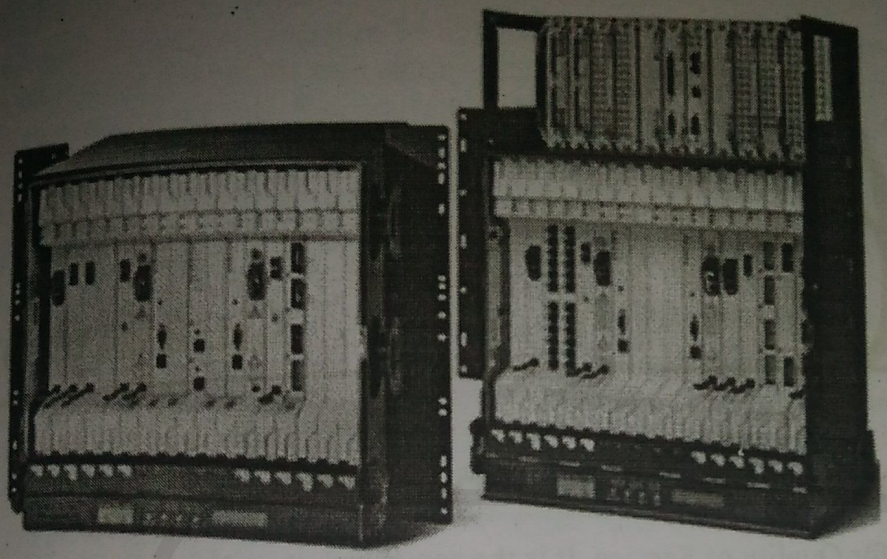


Fig. 5.21. Cisco ONS 15454 Series

5.6 SONET ADVANTAGE AND DISADVANTAGE

SONET offers a lot of advantages further to the advantages offered by fiber optics

- The fact that SONET is highly standardized offers the benefits of interconnectivity and interoperability between equipment of different manufacturers
- SONET/SDH is fully extendable to the customer premises
- SONET local loops provides many end to end advantages

SONET supports the aggregation of all kinds of traffic including data, voice and video

- It is attractive for bandwidth intensive applications and its resiliency is a major plus factor
- It offers high security due to the difficulty of tapping into the fiber optic links
- A major disadvantage is its high cost
- The full advantages of SONET are better leveraged by circuit-switched traffic in comparison to the packet-switched counterpart

REFERENCES

1. http://www.cisco.com/c/en/us/support/docs/optical/synchronous-optical-network-sonet/13567-sonet-tech-tips.html#example.
2. https://users.encs.concordia.ca/~dongyu/ELEC6851/lec06.pdf
3. http://docstore.mik.ua/univercd/cc/td/doc/product/ong/15400/r32docs/instoper/5432cnfg.htm.
4. http://blog.blackbox.com/technology/2015/04/8-advantages-to-choosing-fiber-over-copper-cable/

CHAPTER 6

Quality of Service

6.1 INTRODUCTION TO QUALITY OF SERVICE (QoS)

QoS stands for Quality of Service. It is the computer networks capability of providing better se... to prioritized or in other words selected network traffic over various underlying technologies... Ethernet, SONET, Frame Relay, ATM, IP and other networks which use IP routing. Thus Qo... also be defined as that feature of the computer network which enables the network to differe... between various classes of network traffic and treat them differently.

The various parameters to measure QoS of a network are as follows:

1. Service Availability : The reliability of users' connection to the internet device.
2. Delay : The time taken by a packet to travel through the network from one end to another...
3. Delay Jitter : The variation in the delay encountered by similar packets following the ... route through the network.
4. Throughput : The rate at which packets go through the network.
5. Packet loss rate : The rate at which packets are dropped, get lost or become corrupted (... bits are changed in the packet) while going through the network.

Any network design should try to maximize 1 and 4, reduce 2, and try to eliminate 3 & 5.

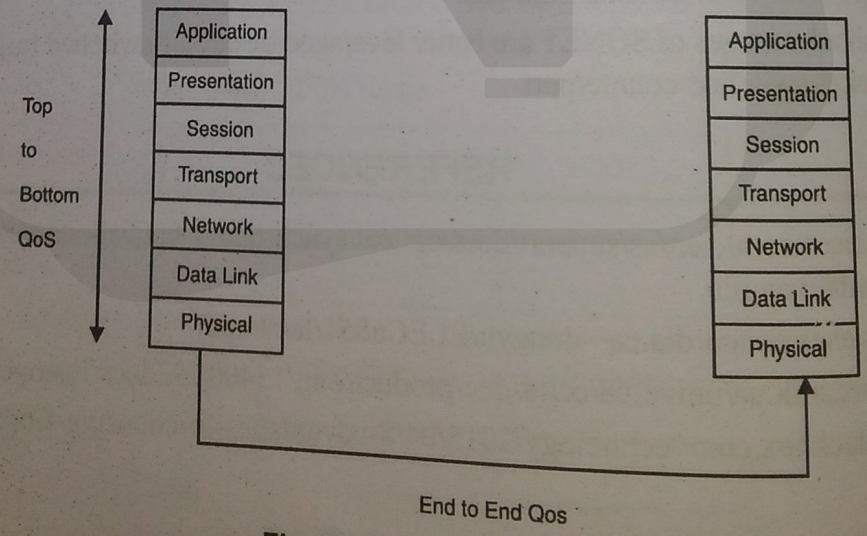


Fig. 6.1 : 2 levels of QoS

6.2 QUEUE ANALYSIS

Often in organization it is crucial to be able to foretell the effect of some change in the network design: either in the form of the load on a system, which is expected to increase or a design change is contemplated due to scaling of organization. For example, an organization supports a number of computer nodes on a 25-Mbps local area network (LAN). Another department in the building is to be added over onto the network. Two questions arise from this. First question is: Can the existing LAN handle the increased workload? Second question is, would it be better to provide a second LAN with a bridge between the two?

The answer involves estimating load generated by each PC since the major concern is system performance. Similarly in the case of interactive or real-time application, the parameter is response time. Likewise throughput can also be the primary issue for network designing. In any case, projections of performance are to be made on the basis of existing load information or on the basis of estimated load for a new environment. A number of approaches are possible:

1. Do an after-the-fact analysis based on actual values.
2. Make a simple projection by scaling up from existing experience to the expected future environment.
3. Develop an analytic model based on queuing theory.
4. Program and run a simulation model.

Option 1 is no option at all: we will wait and see what happens. This leads to unhappy users and to unwise purchases. Option 2 sounds more promising. The analyst may take the position that it is impossible to project future demand with any degree of certainty. Therefore, it is pointless to attempt some exact modeling procedure. Rather, a rough-and-ready projection will provide ballpark estimates. The problem with this approach is that the behaviour of most systems under a changing load is not what one would intuitively expect. If there is an environment in which there is a shared facility (e.g., a network, a transmission line, a time-sharing system), then the performance of that system typically responds in an exponential way to increases in demand.

Queueing network modelling, is a particular approach to computer system modelling in which the computer system is represented as a network of queues which is evaluated analytically. A network of queues is a collection of service centres, which represent system resources, and customers, which represent users or transactions. Analytic evaluation involves using software to solve efficiently a set of equations induced by the network of queues and its parameters.

For networking problems, analytic models based on queuing theory provide a reasonably good fit to reality. The disadvantage of queuing theory is that a number of simplifying assumptions must be made to derive equations for the parameters of interest. The final approach is a simulation model. Here, given a sufficiently powerful and flexible simulation programming language, the analyst can model reality in great detail and avoid making many of the assumptions required of queuing theory. However, in most cases, a simulation model is not needed or at least is not advisable as a first step in the analysis. For one thing, both existing measurements and projections of future load carry with them a certain margin of error. Thus, no matter how good the simulation model, the value of the results are limited by the quality of the input. For another, despite the many assumptions required of queuing theory, the results that are produced often come quite close to those that would be produced by a more careful simulation analysis. Furthermore, a queuing analysis can literally be accomplished in a matter of minutes for a well-defined problem, whereas simulation exercises can take days, weeks,

or longer to program and run. Accordingly, it behaves the analyst to master the basics of queuing analysis. There are two types of queuing models:

(*a*) Single server queuing models

(*b*) Multi server queuing models

6.3 QOS MECHANISMS

- **Classification:** Each class-oriented QoS mechanism has to support some type of classification
- **Marking:** Used to mark packets based on classification and/or metering
- **Congestion Management:** Each interface must have a queuing mechanism to prioritize transmission of packets
- **Traffic Shaping:** Used to enforce a rate limit based on the metering by delaying excess traffic
- **Compression:** Reduces serialization delay and bandwidth required to transmit data by reducing the size of packet headers or payloads
- **Link Efficiency:** Used to improve bandwidth efficiency through compression and link fragmentation and interleaving

6.4 QUEUE MANAGEMENT ALGORITHMS

Queue analysis is implemented in QoS for a network through various Congestion Management Techniques. Congestion control management techniques are techniques which are implemented in core network routers to support the various signalling protocols and provide different classes of service. They can be summarized as follows:

(*a*) First step in these techniques involve creating different queues for different classes of traffic.

(*b*) Next step involves implementing a algorithm for classifying incoming packets and assigning them to different queues.

(*c*) Third step involves scheduling packets out of the various network data queues and prepare them for transmission.

These queuing techniques can be implemented in following four ways:

(*a*) First in first out (FIFO) queues: In this, the data packets are transmitted in the order in which they arrive. The router keeps only one queue for all data packets. Data Packets are then stored in the queue when the network is congested and sent when there is no congestion. However, if the queue is full then the data packets are dropped.

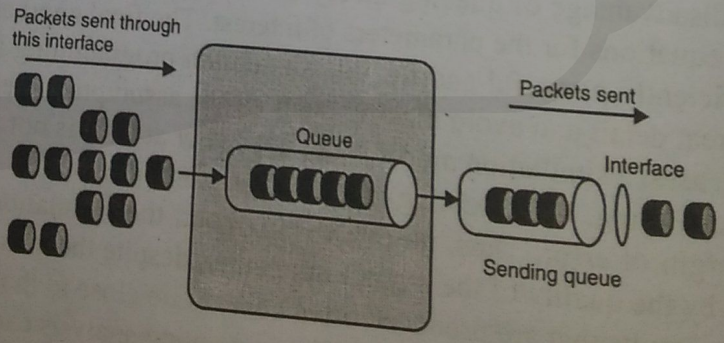


Fig. 6.2: FIFO Queue

(b) Weighted Fair Queuing: Packets are classified into different "conversation messages" by inspection of the ToS value, destination and source port number, destination and source IP address etc. One queue is maintained for each "conversation". Each queue has some priority value or weight assigned to it (once again calculated from header data). Low volume traffic is given higher priority over high volume traffic. e.g. telnet traffic over ftp traffic. After accounting for high priority traffic the remaining bandwidth is divided fairly among multiple queues (if any) of low priority traffic. WFQ also divides packet trains into separate packets so that bandwidth is shared fairly among individual conversations. The actual scheduling during periods of congestion is illustrated through the following example:

If there are 1 queue each of priority 7 to 0 respectively then the division of output bandwidth will be:

$$\text{total} = w_0 + w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 = Sw$$

priority 0 gets w0/Sw th of bandwidth, priority 1 gets w1/Sw th of bandwidth,
priority 2 gets w2/Sw th of bandwidth etc.

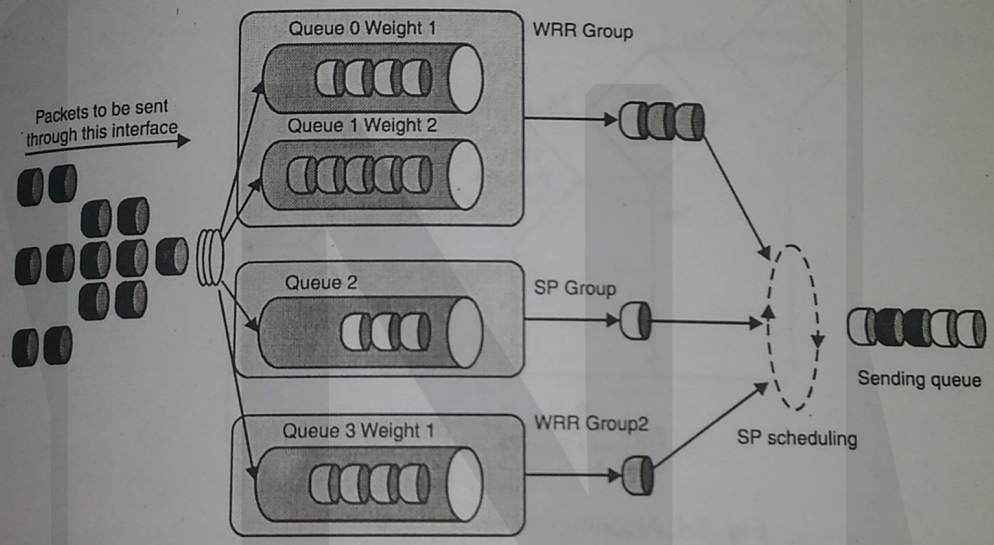


Fig. 6.3. Example of WFQ

The aim of WFQ is to ensure that low volume high priority traffic does get the service levels it expects. It also adapts itself whenever the network parameters change. WFQ cycles through the fair queues and picks up bytes proportional to the above calculation for transmission from each queue. "WFQ acts as a preparator for RSVP, setting up the packet classification and scheduling required for the reserved flows. Using WFQ, RSVP can deliver guaranteed service. RSVP uses the mean data rate, largest amount of data the router will keep in the queue and the minimum QoS to determine bandwidth reservation." During congestion periods ordinary data packets are dropped but messages which have control message data still continue to get enqueued.

(c) Custom Queuing: In this method separate queues are maintained for separate classes of traffic. The algorithm requires a byte count to be set per queue. That many bytes rounded of to the nearest packet is scheduled for delivery. This ensures that the minimum bandwidth requirement by the various classes of traffic is met. CQ round robins through the queues, picking the required number of packets from each. If a queue is of length 0 then the next queue is serviced. The byte counts are calculated as illustrated in the following example :

Suppose we want to allocate 20% for protocol A, 20% for protocol B, 20% for protocol C. Packet sizes for A is 1086 bytes, B is 291 bytes, C is 831 bytes.

Step 1. Calculate % / size ratio: 20/1086 , 60 / 291 , 20 / 831

Step 2. Normalize (by dividing by smallest number) : 1, .20619/.01842 , .02407 / .01842

Step 3. Round upto nearest integer : 1, 12 ,2

Step 4. Multiply each by corresponding byte size of packet : 1086 , 3492, 1662

Verify:

Step 5. Add them : 1086 + 3492 + 1662 = 6240

Step 6. 1086/6240, 3492/6240, 1662/6240 or 17.4, 56, 26.6 which are nearly equal to the ones at the top. CQ is a static strategy. It does not adapt to the network conditions. The system takes a longer while to switch packets since packets are classified by the processor card.

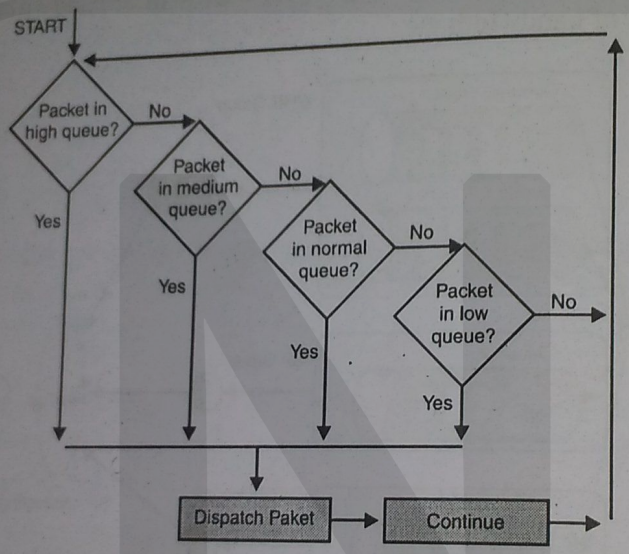


Fig. 6.4. Algorithm for priority queuing

***(d)* Priority Queuing** : We can define 4 traffic priorities - high, medium, normal and low. Incoming traffic is classified and enqueued in either of the 4 queues. Classification criteria are protocol type, incoming interface, packet size, fragments and access lists. Unclassified packets are put in the normal queue. The queues are emptied in the order of - high, medium, normal and low. In each queue, packets are in the FIFO order. During congestion, when a queue gets larger than a predetermined queue limit, packets get dropped. The advantage of priority queues is the absolute preferential treatment to high priority traffic so that mission critical traffic always get top priority treatment. The disadvantage is that it is a static scheme and does not adapt itself to network conditions and is not supported on any tunnels.

6.5 RESOURCE RESERVATION PROTOCOL

The Resource ReSerVation Protocol RSVP was designed to enable the senders, receivers and routers of communication sessions (either multicast or unicast) to communicate with each other in order to set up the necessary router state to support various router the services. RSVP is a novel signalling protocol in at least three ways:

1. It accommodates multicast, not just point-to-multipoint (one-to-many) reservations. To this end, the receiver driven request model permits heterogeneity, in principle, and the filter mechanism allows for calls that reserve resources efficiently for the aggregate traffic flow (e.g. for audio conferencing).
2. It uses soft state, which means that it is tolerant of temporary loss of function without entailing fate-sharing between the end systems and the network nodes. This means that QoS routing can be deployed separately (in more than one way!).
3. RSVP is quite straightforward in packet format and operations, and so is relatively low cost in terms of implementation in end systems and routers. One thing that RSVP is not is a routing protocol. RSVP does not support QoS-dependent routing itself (in other words, such routing is independent of RSVP, and could precede or follow reservations).

RSVP identifies a communication session by the combination of destination address, transport er protocol type and destination port number. It is important to note that each RSVP operation y applies to packets of a particular session and as such every RSVP message must include details the session to which it applies.

Although RSVP is applicable to both unicast and multicast sessions we concentrate here on the re complicated multicast case.

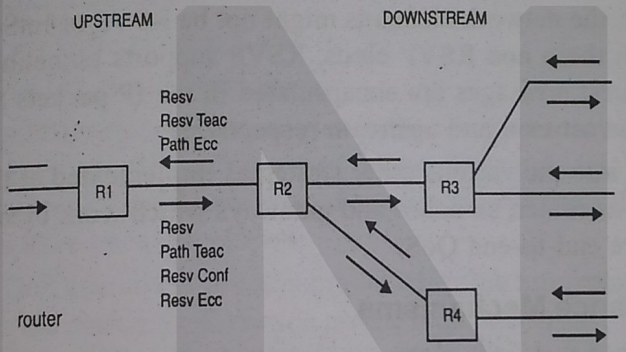


Fig. 6.5. Direction of RSVP messages

RSVP is not a routing protocol, it is a signalling protocol; it is merely used to reserve resources ng the existing route set up by whichever underlying routing protocol is in place. Figure 6.5 shows example of RSVP for a multicast Session involving one sender, S1 and three receivers, RCV1 - V3.

The primary messages used by RSVP are the Path message which originates from the traffic der and the Resv message which originates from the traffic receivers. The primary roles of the message are firstly, to install reverse routing state in each router along the path and secondly, to vide receivers with information about the characteristics of the sender traffic and end-to-end path hat they can make appropriate reservation requests. The primary role of the Resv message is to y reservation requests to the routers along the distribution tree between receivers and senders. rning now to Figure 6.5, as soon as S1 has data to send it begins periodically forwarding RSVP messages to the next hop, R1 down the distribution tree. RSVP messages can be transported by psulating within IP datagrams using protocol number 46 although hosts without this raw I/O bility may first encapsulate the RSVP messages within a UDP header.

6.5.1. RSVP Reservation Types

Reservation types initiated by a receiver may be of the following types :

1. **Distinct Reservation :** The receiver requests to reserve a portion of the bandwidth for each sender. In a multicast flow with multiple senders each senders flow can thus be protected from other senders' flow. This style is also called as the Fixed Filter Style.
2. **Shared Reservations :** Here the receiver requests the network elements to reserve common resources for all the sources in the multicast tree to share among themselves. This style is important for applications like video conferencing, where one sender transmits at a time, since it leads to optimum usage of resources at the routers. They are of 2 types
   - **2a Wildcard Filter Type :** The receiver requests resources to be reserved for all the sources in the multicast tree. Sources may come and go but they should share the same resources to send their traffic, so that the sink can receive from all of them.
   - **2b Shared Explicit Reservation :** This is exactly like the wildcard filter type except that the receiver chooses a fixed set of senders out of all available senders in the multicast flow to share the resources.

Tunneling

In many areas of the internet, the network elements might not be RSVP or IntServ capable. In order for RSVP to operate through these non RSVP clods, RSVP supports tunneling through the cloud. RSVP PATH and RESV request messages are encapsulated in the IP packets and forwarded to the next RSVP capable router downstream and upstream respectively.

Now we will be looking into the various other strategies implemented at the network elements and forwarding devices such as router, switches and gateways which work in tandem with signaling protocols like RSVP to ensure end-to-end QoS.

6.5.2. Congestion Avoidance Mechanisms

Whereas congestion management deals with strategies to control congestion once it has set in, congestion avoidance implements strategies to anticipate and avoid congestion in the first place. There are 2 popular strategies:

1. **Tail drop:** As usual at the output we have queues of packets waiting to be scheduled for delivery. Tail drop simply drops a incoming packet if the output queue for the packet is full. When congestion is eliminated queues have room and taildrop allows packets to be queued. The main disadvantage is the problem of TCP global synchronization where all the hosts send at the same time and stop at the same time. This can happen because taildrop can drop packets from many hosts at the same time.

2. **Random Early Dropping: RED**strategies should only be employed on top of reliable transport protocols like TCP. Only then they can act as congestion avoiders. RED starts dropping packets randomly when the average queue size is more than a threshold value. The rate of packet drop increases linearly as the average queue size increases until the average queue size reaches the maximum threshold. After that a certain fraction - designated as mark probability denominator - of packets are dropped once again randomly. The minimum threshold should be greater than some minimum value so that packets are not dropped unnecessarily. The difference between maximum and minimum threshold should be great enough to prevent global synchronization.

3. **Weighted Random Early Dropping (WRED):** WRED is a RED strategy where in addition it drops low priority packets over high priority ones when the output interface starts getting congested. For IntServ environments WRED drops non-RSVP-flow packets and for Diff Serv environments WRED looks at IP precedence bits to decide priorities and hence which ones to selectively dump. WRED is usually configured at the core routers since IP precedence is set only at the core-edge routers. WRED drops more packets from heavy users than meager users - so that sources which generate more traffic will be slowed down in times of congestion. Non IP packets have precedence 0 - that is highest probability to be dropped. The average queue size formula is :

average = (old_average * 2^ (-n)) + (current_queue_size * 2^(-n))

where n is the exponential weight factor configured by the user. A high values of n means a slow change in the "average" which implies a slow reaction of WRED to changing network conditions - it will be slow to start and stop dropping packets. A very high n implies no WRED effect. Low n means WRED will be more in synch with current queue size and will react sharply to congestion and decongest ion. But very low n means that WRED will overreact to temporary fluctuations and may drop packets unnecessarily.

6.6 DIFFSERV AND INTSERV

- **DiffServ = Diff**erentiated **Serv**ices VS
- **IntServ = Int**egrated **Serv**ices

The need for different service qualities in Internet protocol-based networks is growing stronger as the Internet protocol (IP) turns out to become the universal network architecture. Different applications require different qualities of service (QoS) from the network. To address this in IP, two approaches have been developed.

First, strict QoS guarantees are accomplished by the integrated services (IntServ) architecture in conjunction with the resource reservation protocol (RSVP) used for signaling. This framework allows reserving resources on a path through the network to achieve an end-to-end QoS guarantee, but it has shortcomings with regard to scalability since every router on that path has to maintain per-flow state information.

Second, the differentiated services (DiffServ) architecture, which gives a loose notion of QoS, enables the network to optimize the transport of data packets according to certain requirements. DiffServ only uses different per-hop behaviors (PHBs) for different classes of traffic rather than giving guarantees on these transport characteristics. Such PHBs are implemented on every DiffServ enabled router by mapping different traffic aggregates to different queues. These traffic aggregates are distinguished by the DiffServ codepoint (DSCP) in the IP header. Several main PHBs have already been defined - EF Traffic, AF Traffic, and BF Traffic:

- EF (Expedited forwarding) - a high priority service trying to achieve zero packet loss, minimal queuing delay, and minimal jitter
- AF (Assured Forwarding) - a group of several PHBs giving a variety of different forwarding assurances by defining four classes (distinguished by the resources available per class, namely buffer space and bandwidth) with three different drop precedences
- BE (Best Effort) - a low priority service equivalent to the service in DiffServ-unaware networks.

6.6.1. IntServ

A non-scalable (limited to smaller networks) reservation architecture which consists of *Flow Specs* and *RSVP*

IntServ or Integrated Services is an architecture, which specifies elements and allows them to request and receive "reservations", which act to guarantee quality of service (QoS) on networks.The two protocols of IntServ are:

- **Flow Specs** - describe the IntServ reservations
- **RSVP** - the IntServ signaling protocol to transmit the reservations across the network.

IntServ vs DiffServ

IntServ specifies a fine-grained QoS system, meaning there are many levels of QoS, which are defined and stored in the routers. DiffServ is the opposite - it is a coarse-grained control system, with only several QoS levels.

Flow Spec

There are two parts to a flow spec:

- What does the traffic look like? Done in the Traffic SPECification or TSPEC part.
- What guarantees does it need? Done in the service Request SPECification or RSPEC part.

TSPECs include token bucket algorithm parameters. The idea is that there is a token bucket which slowly fills up with tokens, arriving at a constant rate. Every packet which is sent requires a token, and if there are no tokens, then it cannot be sent. Thus, the rate at which tokens arrive dictates the average rate of traffic flow, while the depth of the bucket dictates how 'bursty' the traffic is allowed to be.

TSPECs typically just specify the token rate and the bucket depth. For example, a video with a refresh rate of 75 frames per second, with each frame taking 10 packets, might specify a token rate of 750Hz, and a bucket depth of only 10. The bucket depth would be sufficient to accommodate the 'burst' associated with sending an entire frame all at once. On the other hand, a conversation would need a lower token rate, but a much higher bucket depth. This is because there are often pauses in conversations, so they can make do with less tokens by not sending the gaps between words and sentences. However, this means the bucket depth needs increasing to compensate for the traffic being burstier.

RSPECs specify what requirements there are for the flow: it can be normal internet 'best effort', in which case no reservation is needed. This setting is likely to be used for webpages, FTP, and similar applications. The 'Controlled Load' setting mirrors the performance of a lightly loaded network: there may be occasional glitches when two people access the same resource by chance, but generally both delay and drop rate are fairly constant at the desired rate. This setting is likely to be used by soft QoS applications. The 'Guaranteed' setting gives an absolutely bounded service, where the delay is promised to never go above a desired amount, and packets never dropped, provided the traffic stays within spec.

6.6.2. RSVP (RFC 2205)

The Resource ReSerVation Protocol (RSVP) is described in RFC 2205. All machines on the network capable of sending QoS data send a PATH message every 30 seconds, which spreads out through the network. Those who want to listen to them send a corresponding RESV (short for "Reserve") message which then traces the path backwards to the sender. The RESV message contains the flow specs.

The routers between the sender and listener have to decide if they can support the reservation being requested, and if they cannot then send a reject message to let the listener know about it. Otherwise, once they accept the reservation they have to carry the traffic.

The routers then store the nature of the flow, and also police it. This is all done in soft state, so if nothing is heard for a certain length of time, then the reader will time out and the reservation will be cancelled. This solves the problem if either the sender or the receiver crash or are shut down incorrectly without first cancelling the reservation. The individual routers may, at their option, police the traffic to check that it conforms to the flow specs.

Why IntServ with its RSVP signaling Failed

The problem with IntServ is that many states must be stored in each router. As a result, IntServ works on a small-scale, but as you scale up to a system the size of the Internet, it is difficult to keep track of all of the reservations. As a result, IntServ is not very popular. Summarizing:

- Reservations in each device along the path are "soft," which means they need to be refreshed periodically, thereby adding to the traffic on the network and increasing the chance that the reservation may time out if refresh packets are lost. Though some mechanisms alleviate this problem, it adds to the complexity of the RSVP solution.
- Maintaining soft-states in each router, combined with admission control at each hop adds to the complexity of each network node along the path, along with increased memory requirements, to support large number of reservations.
- Since state information for each reservation needs to be maintained at every router along the path, scalability with hundreds of thousands of flows through a network core becomes an issue.

DiffServ

DiffServ (RFC 2474, 2475, and 3270), on the other hand, addresses the clear need for relatively simple and coarse methods of categorizing traffic into different classes, also called class of service (CoS), and applying QoS parameters to those classes. For making DiffServe a guanateed service it may be used with CAC- Connection admission Control.

To accomplish this, packets are first divided into classes by marking the type of service (ToS) byte in the IP header. A 6-bit bit-pattern (called the Differentiated Services Code Point [DSCP]) in the IPv4 ToS Octet or the IPv6 Traffic Class Octet is used.

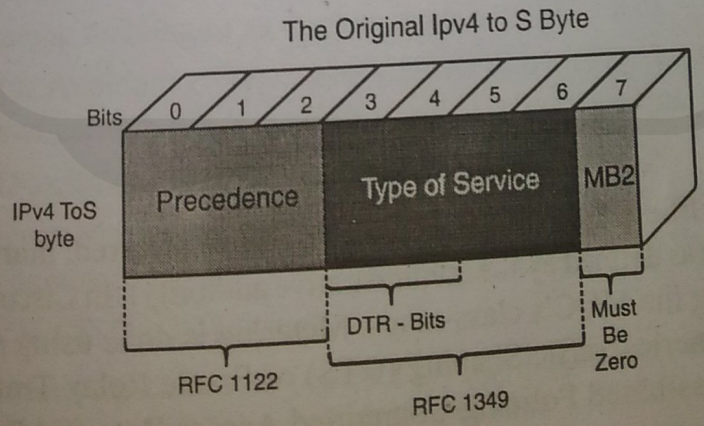


Fig. 6.6: IPv4 Traffic class octet

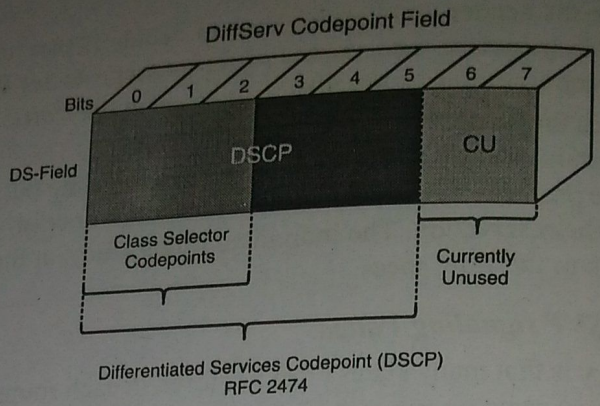


Fig. 6.7. Diffserv codepoint field

Unlike the IP-Precedence solution, the ToS byte is completely redefined. Six bits are now used to classify packets. The field is now called the DS (Differentiated Services) Field, with two of the bits unused (RFC-2474). The 6 bits replace the three IP-Precedence bits, and is called the Differentiated Services Codepoint (DSCP). With DSCP, in any given node, up to 64 different aggregates/classes can be supported (2^6). All classification and QoS revolves around the DSCP in the DiffServ model.

CoS levels can be applied as Premium, Gold, Silver, and Bronze by setting the DSCP to a corresponding value.

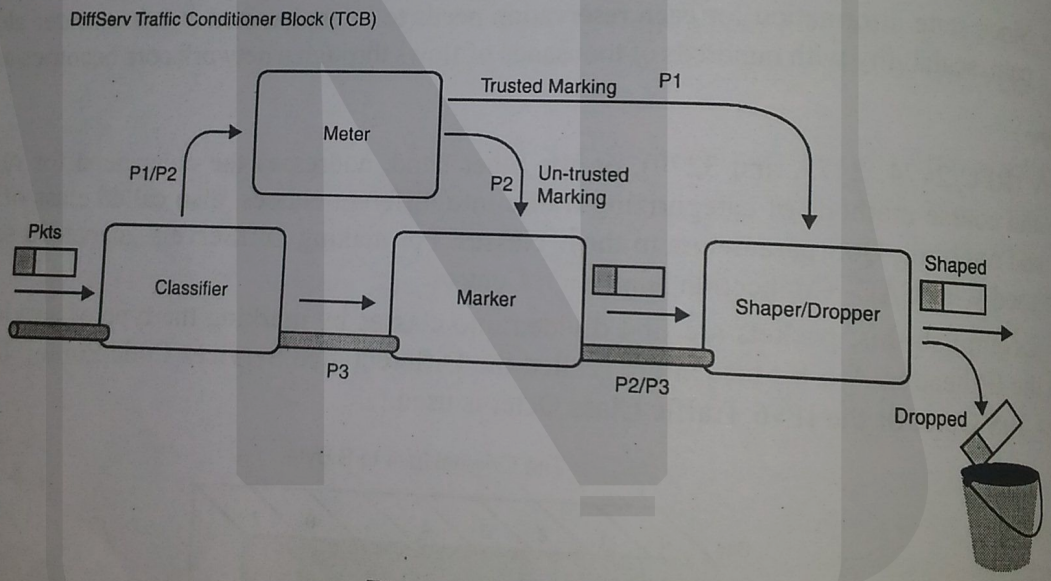


Fig. 6.8. Diffserv TCB

Packets entering a DiffServ Domain (DS-Domain) can be metered, marked, shaped, or policed to implement traffic policies (as defined by the administrative authority). In Cisco IOS software, classifying and marking is done using the MQC's class-maps. Metering is done using a token bucket algorithm, shaping is done using Generic Traffic Shaping (GTS) or Frame Relay Traffic Shaping (FRTS), and policing is done using class-based Policing/Committed Access Rate (CAR).

REFERENCES

1. **[nortel1]**IP QoS - A Bold New Network Nortel / Bay Networks White Paper, Sept 1999, 24 pages http://www.nortelnetworks.com/prd/isppp/collateral/ip_qos.pdf
   A tutorial on Nortel's view of future networks and the prevalent network architecture.
2. **[gqos]**Specification of Guaranteed Quality of Service (RFC 2212), 19 pages http://www.rfc-editor.org/in-notes/rfc2212.txt
   Describes the guaranteed quality of service class in intserv.
3. **[mplsa]**Multiprotocol Label Switching Architecture draft-ietf-mpls-arch-06.txt, August 1999, 60 pages
   Describes the entire mpls concept and the basic architecture.
4. **[nortel3]**IP Traffic Engineering using MPLS Explicit Routing in Carrier Networks, Nortel NetworksWhite Paper April 1999, 8 pages
   http://www.nortelnetworks.com/products/library/collateral/55046.25-10-99.pdf
   Describes various MPLS features and label distribution protocols.
5. **[clqos]**Specification of the Controlled-Load Network Element Service (RFC 2211), 16 pages http://www.rfc-editor.org/in-notes/rfc2211.txt
   The formal specification of controlled load service of intserv.
6. **[xiao99]**Internet QoS : the Big Picture Xipeng Xiao & Lionel M. Ni, IEEE Network, January 1999,25 pages.
   Describes the major QoS issues and protocols.
7. **[rsvp2]** Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification (RFC 2205), 110 pages http://www.rfc-editor.org/in-notes/rfc2205.txt
   This document gives the formal specification for RSVP.
8. **[ciscorsvp]**Resource Reservation Protocol (RSVP) CISCO White Papers, Jun 1999 , 15 pages
   A very concise and to the point summary of RSVP.
9. **[nortel2]** Preside Quality of Service Nortel Networks Position Paper, 11 pages
   Describes the future network architecture and support of QoS.
11. **[CISCO1]** Congestion Management Overview, CISCO White Papers, 1999, 14 pages http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart2/qcconman.htm
    Describes various queueing techniques as implemented in CISCO IOS.
12. **[CISCO2]** Congestion Avoidance Overview, CISCO White Papers, 1999, 16 pages http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart3/qcconavd.htm
    Describes various RED techniques as implemented in CISCO IOS.
13. **[CISCO3]** Link Efficiency Mechanisms, CISCO white papers, 1999, 8 pages http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart6/qclemech.htm
    LFI implementation as in CISO IOS product.
14. **[CISCO4]** QoS overview, CISCO white papers 1999. 24 pages http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcintro.htm
    A excellent overview of QoS capabilities in CISCO's IOS product + definitions.

15. **[diffserva]** An Architecture for Differentiated Services RFC 2475, 36 pages http://www.rfc-editor.org/in-notes/rfc2475.txt
This described the Diff Serv architecture and model in detail
16. **[afphb]** Assured Forwarding PHB Group RFC 2597, 10 pages http://www.rfc-editor.org/in-notes/rfc2597.txt
The AF PHB is described here
17. **[efphb]** Expedited Forwarding PHB Group RFC 2598, 10 pages http://www.rfc-editor.org/in-notes/rfc2598.txt
The EF PHB is described here
18. **[sbm]** A Protocol for RSVP-based Admission Control over IEEE 802-style networks draft-ietf-issll-is802-sbm-09.txt, 67 pages
The Subnet Bandwidth Manager is proposed here
19. **[stardust]** QoS Protocols and Architectures, Stardust White Paper, 17 pages http://www.stardust.com/qos/whitepapers/protocols.htm
A brief tutorial on the common QoS protocols is mentioned here.
20. **[rsvp]**D. Durham, R. Yavatkar, Inside the Internet's Resource Reservation Protocol, John Wiley and Sons, 1999, 351 pages
Provides excellent coverage of all aspects of RSVP and some topics of IntServ. Excellent figures.
21. **[rj99]**QoS over Data Networks, Raj Jain, CIS 788 handouts, The Ohio State University, Fall 99, 4 pages of slides (6 slides per page)http://www.cse.wustl.edu/~jain/cis788-99/h_6qos.htm
22. http://cse.csusb.edu/ykarant/courses/f2006/csci530/QueuingAnalysis.pdf
23. http://www.cl.cam.ac.uk/~jac22/books/mm/book/node52.html
Provides excellent overview of recent advances in the area of Quality of Service issues.

List of Acronyms

AF	Assured Forwarding
COPS	Common Open Policy Service
DiffServ	Differentiated Services
DSBM	Designated Subnet Bandwidth Manager
EF	Expedited Forwarding
IntServ	Integrated Services - IETF Standard
LDP	Label Distribution Protocol
LFI	Link Fragmentation and Interleaving
LSP	Label Switched Path
LSR	Label Switching Router
MPLS	Multi Protocol Label Switching
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHP	Per Hop Behaviour
QoS	Quality of Service

RED	Random Early Dropping
RSVP	Resource Reservation Protocol
SBM	Subnet Bandwidth Manager
SLA	Service Level Agreement
TCA	Traffic Conditioning Agreement
WFQ	Weighted Fair Queueing
WRED	Weighted Random Early Dropping