UNIT 2 TRANSPORT LAYER

Structure **Page No.**

2.0 INTRODUCTION

The transport layer supports two protocols in TCP/IP protocol suite. One is Transmission Control Protocol (TCP). TCP is connection oriented that provides reliable end-to-end transmission. Another protocol is User Datagram Protocol (UDP). UDP is simple and provides well sequenced transport function when reliability and serving are less important than size and speed. Transport layer services are implemented by transport protocols used between two transport entities. Transport layer services are similar to the data link services. Data link layer is designed to provide its services within a single network, while the transport layer provides services across an inter network made up of many networks. There are seven categories of services provided by the transport layer. These services are End to end delivery, Addressing, Reliable delivery, Flow control, Connection management, Multiplexing and Congestion Control.

2.1 OBJECTIVE

After going through this unit, you should be able to:

- Know the Functions and Services of transport layer
- Understand the Working of transport layer
- Understand the TCP Window management
- Know the different transport layer design issues

End to End Delivery

As shown in figure 1, network layer answers the end to end delivery of individual packets from a machine to another machine in different network, but does not see any relationship between those packets. It treats each as an independent entity. Further, the packet needs to be delivered to the last participating entity, i.e. a process engaged in data exchanged. But the transport layer makes sure that the entire message (not a single packets receives) is delivered to a process that is the end (last) entity participating in message exchange. So it provides process-to-process or end-to-end delivery of an entire message.

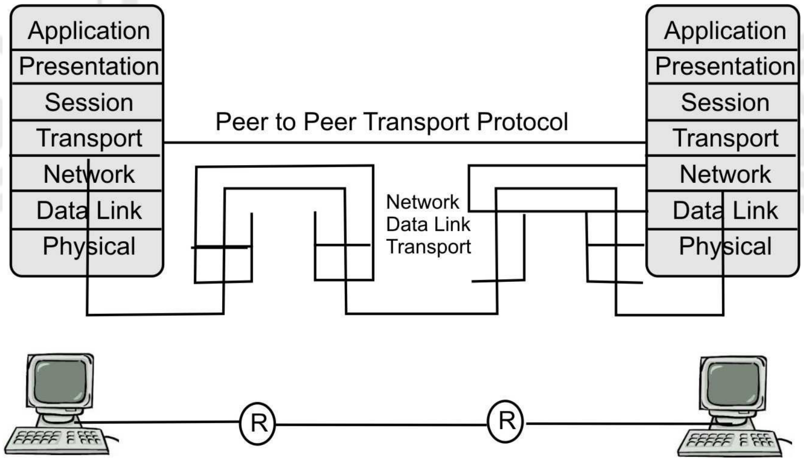

Figure 1: End to End delivery of packets

2.2 ADDRESSING

Transport layer interact with the functions of session layer. Many protocols combine session, presentation and application level protocols into a single package called an application. In these cases delivery to the session layer functions is, in effect delivery to the application. So communication occurs not just from end machine to end machine but from end application to end application. Data generated by an application on one machine must be received not just by other machines but by the correct application on that machine.

In most cases, we end up with the communication between many to many entities, called service access points as shown in figure 2 given below. To ensure accurate delivery from service access point to service access point we used another level of addressing in addition to the network and data link level.

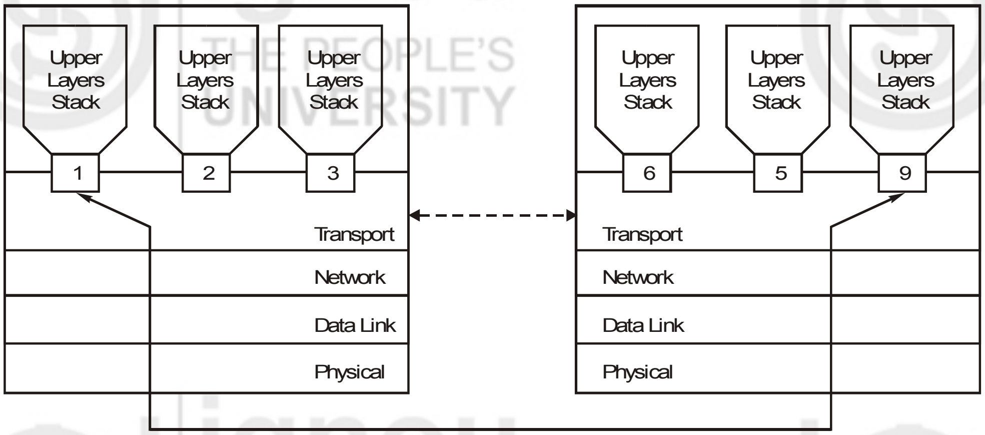

Figure 2: Service Access Points

2.3 RELIABLE DELIVERY

It is responsible for reliable delivery of data by providing the following methods also shown in figure 3:

- Error control
- Sequence control
- Loss control

- Duplication control

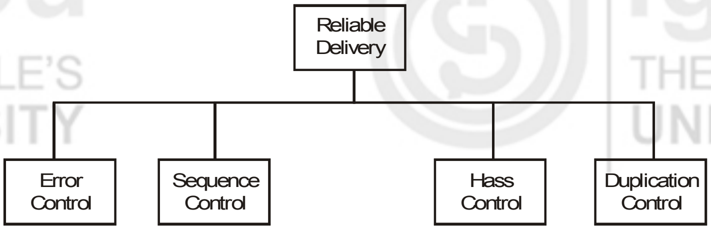

Figure 3: Methods of reliable delivery of packets

a) **Error Control:** When transferring data the primary goal of reliability if occur control. Data must be delivered to their destination. Exactly as they originated from the source. The reality of physical data transport are that while 100 per cent error free delivery is probably impossible, transport layer protocols are designed to come as close as possible.

Mechanisms full errors handling at this layer are based on error detection and retransmission. With the error handling, performed using algorithms implemented in software such as checksum “error detection and correction”.

b) **Sequence Control:** Second aspect of reliability implemented at the transport layer is sequence control. On the sending end, the transport layer is responsible for ensuring that data with received from the upper layers are usable by the lower layers. ON the receiving end it is responsible for ensuring that the various pieces of a transmission are correctly reassembled.

Segmentation: When the size of the data units received from the upper layer is too long for the network layer datagram and data link layer frame to handle, the transport layer divides it into smaller usable blocks. This dividing process is called segmentation.

Concatenation: When the sizes of the data units belonging to a single session are so small that several can fit together into a single data queue are frame, the transport protocol combines them into a single data unit. This combining process is called concatenation.

Sequence Number: Most transport layer services add sequence number at the end of each segment.

If a longer data unit has been segmented the sequence number indicate the reassembly.

If several shorter units have been concatenated the numbers indicate the end of each subunit and allow them to be separated accurately at the destination.

c) **Loss Control:** The third aspect of reliability covered by the transport layer is loss control as depicted in figure 4. The transport layer ensures all pieces of the transmission arrive at the destination, not just some of them. When data have been segmented for delivery, some segments may be lost in transmit. Sequence number allows the receiver’s transport layer protocol to identify any missing segment and request in delivery.

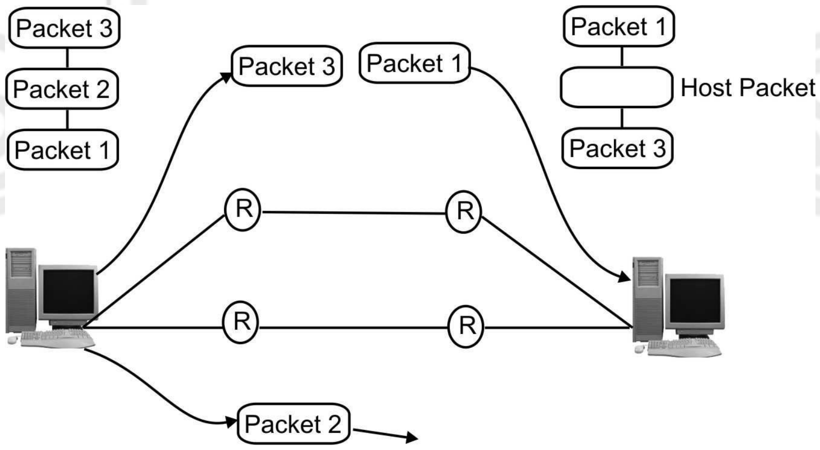

Figure 4: Loss Control

d) **Duplication Controls:** The fourth aspect of reliability by the transport layer is duplication control as shown in figure 5. Transport layer functions must guarantee that no places of data arrive at the receiving system duplicated. As they allow identification of last packets, sequence no. allows the receiver to identify and discard duplicate segments.

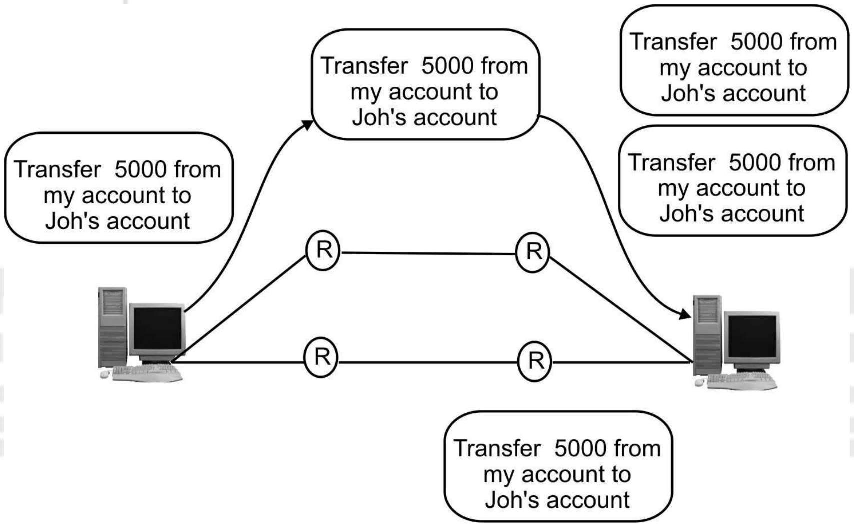
Transfer 5000 from my account to Joh's account
Transfer 5000 from my account to Joh's account
Transfer 5000 from my account to Joh's account
Transfer 5000 from my account to Joh's account
R
R
R
R
Transfer 5000 from my account to Joh's account

Figure 5: Duplication Control

☞ Check Your Progress 1

1. Which layer ensures the process-to-process or end-to-end delivery of an entire message? Explain.

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

2. List the methods or mechanism provided by Transport layer for reliable delivery of data.

……………………………………………………………………………………

……………………………………………………………………………………

2.4 FLOW CONTROL

Like the data link layer transport layer is responsible for flow control. Flow control is performed end to end rather than across a single link. Transport layer flow control uses a sliding window protocol. The window at the transport layer can vary in size to accommodate buffer occupancy as depicted in figure 6 given below.

Sliding window is used to make data transmission more efficient as well as to control the flow of data so that the receiver does not become overwhelmed. Sliding window used at the transport layer are usually byte oriented rather than frame oriented.

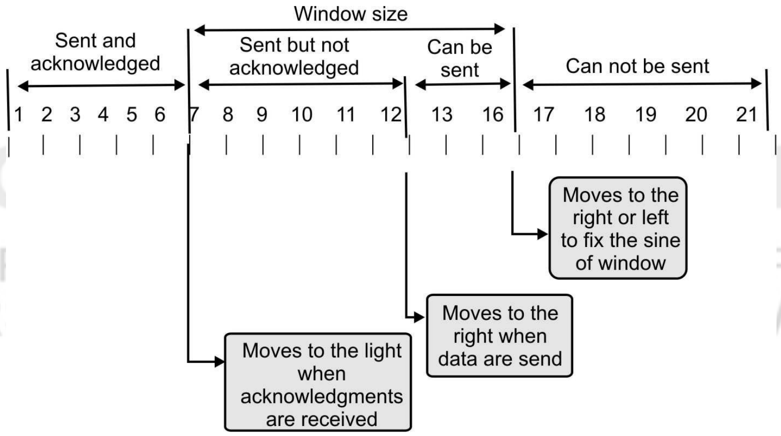
Window size
Sent and acknowledged
Sent but not acknowledged
Can be sent
Can not be sent
1 2 3 4 5 6 7 8 9 10 11 12 13 16 17 18 19 20 21
Moves to the right or left to fix the sine of window
Moves to the right when data are send
Moves to the light when acknowledgments are received

Figure 6: Sliding Window for Flow Control

Some points about sliding window at the transport layer are as follows:

1. Sender does not have to send a full window's worth of data.
2. An acknowledgement can expand the size of the window based on the sequence number of the acknowledged data segment.
3. The size of the window can be increased as decreased by the receiver.
4. The receiver can send acknowledgement at anytime.

2.5 CONNECTION MANAGEMENT

End to end delivery can be accomplished in two ways connection oriented and connectionless. The connection oriented mode is most commonly used from both two modes. A connection oriented protocol establishes a virtual circuited or pathway through the internal between sender and receiver. All of the packets belonging to a message are then sent over this same path. Using a single pathway for the entire message facilitates the acknowledgement process and retransmission of damaged and lost frames connection oriented services is generally considered reliable.

Connection Oriented transmission has three stages:

1. Connection establishment.
2. Data transfer
3. Connection termination.

Connection Establishment: Before communicating device can send data to the other, the initializing device must first determine the availability of the other to exchange data and a pathway must be found through the network by which the data can be sent.

This step is called connection establishment. Connection establishment requires three actions called three way handshake as shown in figure 7 given below.

- The computer requesting the connection sends a connection request packet to the intended receiver.
- The responding computer returns a confirmation packet to the requesting computer.
- The requesting computing returns a packet acknowledging the confirmation.

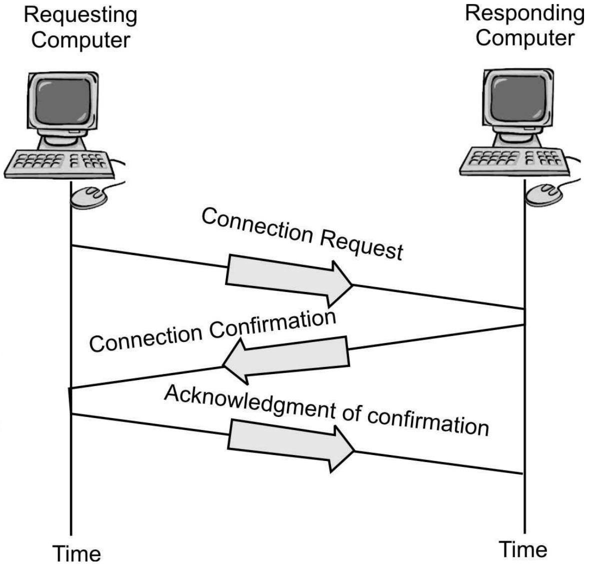

Figure 7: Connection Establishment

Connection Termination: Once all of the data have been transferred, the connection must be terminated.

Connection termination also requires three way handshake as shown in figure 8:

- Requesting computer sends a disconnection request packet.
- Responding computers confirms the disconnection request.
- The requesting computer acknowledges the confirmation.

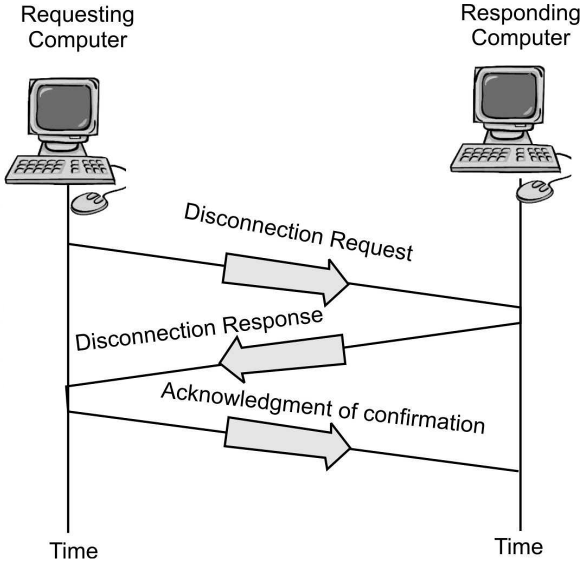

Figure 8: Connection Termination

2.6 MULTIPLEXING

To improve transmission efficiency, the transport layer has the option of multiplexing. Multiplexing at this layer occurs in two ways as shown in figure 9:

1. **Upward** -meaning that multiple transport layer connections use the same network connection

2. **Downward-** (meaning that one transport layer connection uses multiple network connections.

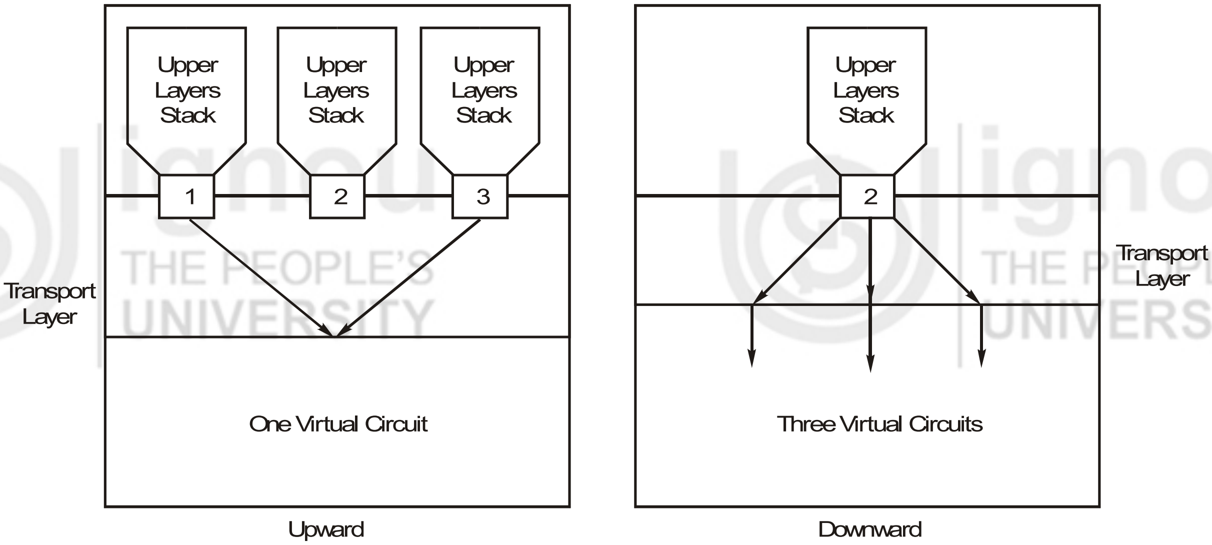

Figure 9: Upward and downward Multiplexing

Upward Multiplexing: Transport layer uses virtual circuits based on the services of the lower three layers. The underlying network charge for each virtual circuit connection. To make well cost-effective use of an established circuit, the transport layer sends several transmissions based for the same destination along the same path by upward multiplexing.

Downward Multiplexing: It allows the transport layer to split a single connection among several different paths to improve throughput (speed of delivery) as shown in figure 9. This option is useful when the underlying networks have low or slow capacity. For example, some network layer protocols have restriction on the sequence number that can be handled .X.25 uses a three bit numbering code, so sequence number are restricted to the range of 0 to 7. In this case throughput can be unacceptably low. To counteract this problem the transport layer can use more than one virtual circuit at the network layer to improve throughput by sending several data segment at once delivery is faster.

2.7 CONGESTION CONTROL

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. Congestion control mechanism is divided into two broad categories:

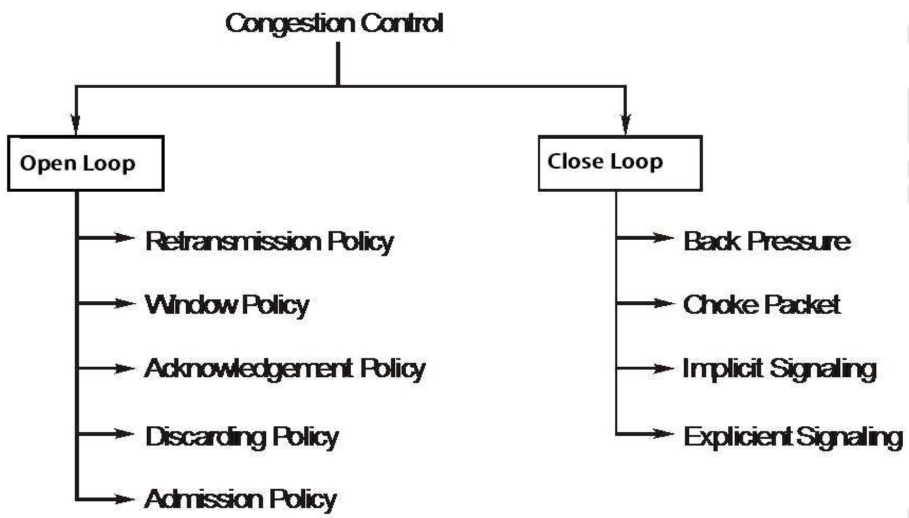

Figure 10:

Open Loop: Polices are applied to prevent congestion before it happens.

Closed Loop: Mechanism try to alleviate congestions after it happens.

Congestion Control Policy

TCP's general policy for handling congestion is based on 3 phases:

- Slow start,
- Congestion avoidance, and
- Congestion detection

The sender starts with a very slow rate of transmission but increases the rate rapidly to reach a threshold. When threshold is reached, the data rate is reduced to avoid congestion.

Congestion Avoidance: Additive Increase

In the congestion avoidance algorithm, the size of the congestion window increases additively until congestion is detected.

Congestion Detection: Multiplicative Decrease

An implementation reacts to congestion detection in one of the following ways:

1. If detection is by time-out, a new slow-start phase starts.
2. If detection is by three (acknowledgements) Acks, a new congestion Avoidance phase starts.

Slow Start: Exponential Increase

At the beginning of the connection, set the congestion window size to the maximum segment size. For each segment that is acknowledged, increase the size of the congestion window by one maximum size until you reach a threshold of half the allowable window size. This is some timer called slow start which is totally misleading because the process is not slow at all.

The size of the congestion window increase exponentially. The sender sends one segment receives one acknowledgement, increases the size to two segments, sends two segments, receives ack for two segments; increase the size to four and so on.

In other words, after receiving the 3rd ACK, the size of the window has been increased to eight segments (i.e. $2^3 = 8$). To avoid congestion before it happen one must slow down this exponential growth. After the size reaches the threshold, the size is increased one segment for each acknowledgement even if as ACK is for several segment.

Multiplicative Decrease

If the congestion occurs the congestion windows size must be decreased. The only way the sender causes that connection has occur through a lost segment. If the sender does not receive an acknowledgement for a segment before it transmission timer matured, it assumes that there is congestion.

The strategy says if a time out occurs, the threshold must be set to half of the congestion window size and the congestion or size should start from one again.

2.8 QUALITY OF SERVICES (QOS)

A stream of packet, from a source to a destination is called a flow. In connection oriented network, all the packets belonging to a flow the same route. But in connection less network they may follow different routes. The need of each flow can be characterized by four primary parameters:

1. Reliability: Reliability is the ability of a system to perform and maintain its functions in normal conditions as well as under unexpected conditions.

2. Delay is defined as the time interval elapsed between the departures of data from the source to its arrival at the destination.

3. Jitter**:** Jitter refers to the variation in time between packets arriving at the destination.

4. Bandwidth refers to the data rate supported by a network connection or interface.

Techniques to Improve QOS

1. **Over Provisioning:** An easy solution is to provide so much router capacity, buffer space and bandwidth that the packets just fly through costly.

2. **Buffering:** Flows can be buffered on the receiving side before being delivered. Buffering those does not affected the reliability of bandwidth and increases the delay but it smoothes out the jitter.

3. **Scheduling:** Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner.

   Here we discuss some scheduling techniques to improve the quality of service such as:

   i) FIFO Queuing

   ii) Priority Queuing

   iii) Weighted Fair Queuing

FIFO Queuing

In FIFO Queuing packets wait in a buffer (Queue) until the node is ready to process them. If the average rate is higher than the average processing rate, the queue will fill up and new packets will be discarded.

Priority Queuing

In this packets are first assigned to **priority class**. Each priority class has its own Queue. The packets in the highest priority Queue are processed first. But if there is a continuous flow in high-priority Queue, the packets in the low priority Queues will never have a chance to be processes.

Weighted Fair Queuing

In this method, the packets are still assigned to different classes and admitted to different queues. The queues, however, are weighted based on the priority of the queues; higher priority means a higher weight.

The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight.

For example: If the weights are 3, 2 and 1, three packets are processed from the first queue, two from the second queue and one from the third queue.

If the system does not impose priority on the classes, all weights can be equal. In thus way, we have fair queuing with priority.

4. Traffic Shaping
   a) Leaky bucket
   b) Token bucket

2.9 TCP WINDOW MANAGEMENT

TCP uses two buffers and one window, to control the flow of data coming from the sending application program. The application program creates data and writes it to the buffer. The sender imposes a window on this buffer and sends the segments as long as the size of the window is not zero. The TCP receiver has buffer also. It receives data, checks them, and stores them in buffer to be consumed by the receiving application program.

A sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become over whelmed with data. TCP's sliding window is byte oriented

TCP's Sliding Window

The source does not have to send a full window's worth of data. The size of the window can be increased or decreased by the destination. The destination can send an acknowledgement at any time.

Silly Window Syndrome

Silly Window Syndrome is another problem that can degrade the TCP performance. This problem occurs when the sender transmits data in large blocks, but an interactive application on the receiver side reads data 1 byte at a time as shown in figure 11.

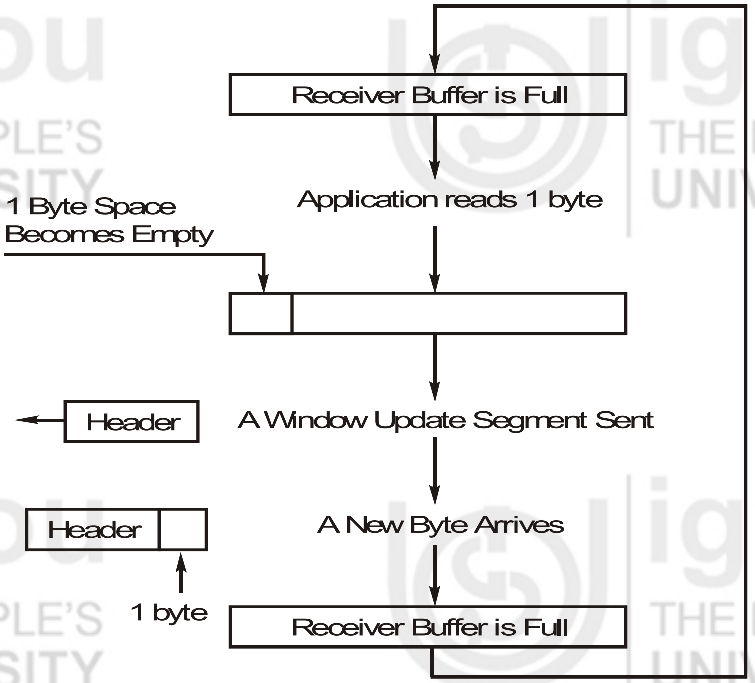

Figure 11: Silly Window Syndrome

1. Initially the receiver's buffer is full so it sends a window size 0 to block the sender.

2. But the interactive application reads one byte from the buffer, so one byte space becomes empty.

3. The receiving TCP sends a window update to the sender informing that it can send 1 byte.

4. The sender sends 1-new byte.

5. The buffer is full again and the window size is 0. The behavior can continue forever's.

Solution to Silly Window Syndrome

A solution to silly window syndrome was suggested as follows:
It was suggested that the **receiver should not send** a window update for 1 byte. Instead it must wait until it has a substantial amount of buffer space available and then sends the window update. To be specific, the receiver should not send a window update until it can handle the maximum window size, it has advertised at the time of establishing a connection or its buffer is half empty, whichever is smaller.

The sender can also help to improve the situation. It should not send tiny segments instead it must wait and send a full segment or at least one containing half of the receivers buffers size.

2.10 PORTS

In computer networking of connection-based communication port is like a medium through which, an application establish a connection with another application by binding a socket by a port number. Addressing the information and the port number, accompanied the data transfer over the network.

The Ports are used by TCP and UDP to deliver the data to the right application, are identified by a 16-bit number present in the header of a data packet. Ports are typically used to map data to a particular process running on a client. If we consider a letter (data packet) sent to a particular apartment (IP) with house no. (Port no), at this time the port no. is the most important part for the delivery of the letter. In order for the delivery to work, the sender needs to include a house number along with the address to ensure the letter gets to the right destination.

Do you know?

TCP and UDP ports are 16 bit number

The TCP and UDP protocols use ports to map incoming data to a particular process running on a computer.

- Port is represented by a positive (16-bit) integer value
- Some ports have been reserved to support common/well known services:

o	FTP	21/TCP
o	TELNET	23/TCP
o	SMTP	25/TCP
o	LOGIN	513/TCP

- User level process/services generally use port number value >= 1024

Types of Port

1. Well known port (0 to 1023)-They are controlled by IANA
2. Registered Port (1024-49159)
3. Dynamic port (49152-65535)

If we consider the client-server architecture, a server application binds a socket to a specific port number in connection-based communication. It registered the server with the system where all the data destined for that port.

Do you know?

Port number permits unique identification of several simultaneous processes using TCP/UDP

Now we are aware of the importance of the port number. In the same order there are some ports which are predefine and called reserved ports. Some of them are given in Table 1given below:

Table 1: Reserved Port Numbers.

Service	Port no.
ECHO	7
DAYTIME	13
FTP	21
TELNET	23
SMTP	25
FINGER	79
HTTP	80
POP3	110

Do you know?

If we consider the range of the port numbers, there are 0 to 65,535 ports available.

Tips

The port numbers ranging from 0 - 1023 are reserved ports or we can say that are restricted ports. All the 0 to 1023 ports are reserved for use by well-known services such as FTP, telnet and http and other system services. These ports are called well-known ports.

☞ **Check Your Progress 2**

1. List three stages of Connection Oriented transmission.

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

2. Compare and contrast between Upward and Downward Multiplexing.

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

2.11 SUMMARY

Transport layer is mainly responsible for end to end reliable delivery, segmentation and concatenation. Two main protocols that operate on transport layer are TCP and UDP.TCP provides reliable connection oriented service while UDP provides unreliable connectionless service. The data link and transport layer perform many of the same duties .The data link layer function in a single network, while the transport layer operates across an internet. Flow control at the transport layer is handles by three walled sliding window. Multiplexing can be downward of upward in transport layer. Connection establishment and termination can be done by using three way handshakes. Transport layer works on port address. To know more about the Transport layer and its protocols (TCP and UDP), student may please refer to the course material of BCS-61TCP/IP Programming or BCS-54 Network Programming.

2.12 REFERENCES/FURTHER READING

1. Computer Networks, A. S. Tanenbaum 4th Edition, Practice Hall of India, New Delhi. 2003.

2. Introduction to Data Communication & Networking, 3rd Edition, Behrouz Forouzan, Tata McGraw Hill.

3. Douglas E. Comer, *Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture* (4th Edition).

4. James F. Kurose, *Computer Networking*: A Top-Down Approach Featuring the Internet (3rd Edition).

5. Larry L. Peterson, *Computer Networks*: A Systems Approach, 3rd Edition (The Morgan Kaufmann Series in Networking).

6. W. Richard Stevens, *The Protocols (TCP/IP Illustrated, Volume 1)*.

7. www.wikipedia.org

8. William Stallings, *Data and Computer Communications*, Seventh Edition.

2.13 SOLUTION /ANSWERS

☞ Check Your Progress 1

1. Transport layer makes sure that the entire message (not a single packets receives) is delivered to a process that is the end (last) entity participating in message exchange. So it provides process-to-process or end-to-end delivery of an entire message.

2. Transport Layer is responsible for reliable delivery of data, it provide following methods to provide reliable delivery of data.

   - Error control
   - Sequence control
   - Loss control
   - Duplication control

☞ Check Your Progress 2

1. Connection Oriented transmission has following three stages:

   i) Connection establishment.

   ii) Data transfer

   iii) Connection termination.

2. **Upward Multiplexing:** Transport layer uses virtual circuits based on the services of the lower three layers. The underlying network charge for each virtual circuit connection. To make well cost-effective use of an established circuit, the transport layer sends several transmissions based for the same destination along the same path by upward multiplexing. **Downward Multiplexing:** Allows the transport layer to split a single connection among several different path to improve throughput (speed of delivery). This option is useful when the underlying networks have low or slow capacity. For example, some network layer protocols have restriction on the sequence number that can be handled .X.25 uses a three bit numbering code, so sequence number are restricted to the range of 0 to 7. In this case throughput can be unacceptably low. To counteract this problem, the transport layer can use more than one virtual circuit at the network layer to improve throughput by sending several data segment at once delivery is faster.