6.1 The Purpose of the Transport Layer

The transport layer is the core of the OSI model. Protocols at this layer oversee the delivery of data from an application program on one device to an application program on another device. More important, they act as a liaison between the upper-layer protocols (session, presentation, and application) and the services provided by the lower layers (network, data link, and physical). The upper layers can use the services of the transport layer to interact with the network without ever having to interact directly with or even be aware of the existence of the lower layers. To make this separation possible, the transport layer itself is independent of the physical network.

To better understand the role of the transport layer, it is helpful to visualize an Internet made up of a variety of different physical networks such as the LANs, MANs, and WANs shown in figure (1). These networks are connected to enable the transport of data from a computer on one network to a computer on another network. As a transmission moves from network to network, the data may be encapsulated in different types and lengths of packets. One network's network or data link functions may chop it into smaller segments to fit a more restricted packet or frame size, while another network's peer functions may link several segments together in a single large packet. The data may even share a frame with other, nonrelated data segments. No matter what transformations they must go through on the way, however, the data must arrive at their destination in their original form.

The upper-layer protocols are kept unaware of the intricacy of the physical networks; so that only one set of upper-layer software has to be developed. To the upper-layers, the individual physical networks are a simple homogenous cloud that somehow takes data and delivers it to its destination safe and sound.

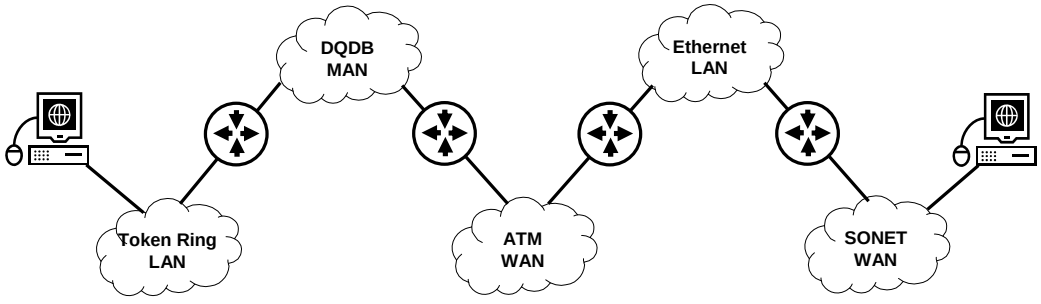


Figure (1) An Internetwork.

The Transport Layer segments the data and manages the separation of data for different applications. Multiple applications running on a device receive the correct data to the following points:

1. Tracking Individual Conversations
2. Segmenting Data (Source)
3. Reassembling Segments (destination)
4. Identifying the Applications (port No.)

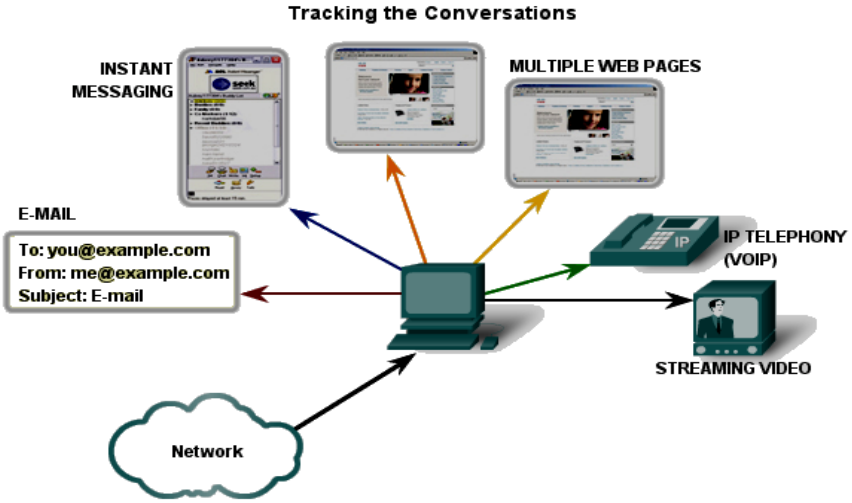


Figure (2) Transport Layer Applications.

6.2 Duties (Primary Functions) of the Transport Layer

Transport services are implemented by transport protocol used between two transport entities as shown in figure (3):

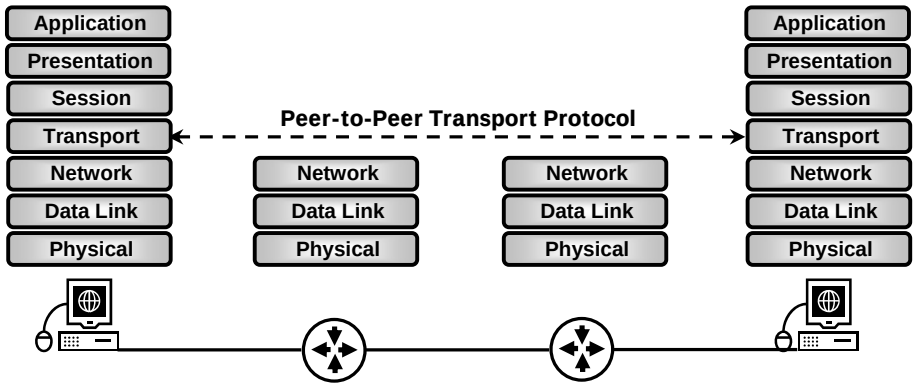


Figure (3) Transport Layer Concept.

The services provided are similar to those of the data link layer. The data link layer, however, is designed to provide its services within a single network, while the transport layer provides these services across an internetwork made of many networks.

The data link layer controls the physical layer, while the transport layer controls all three of the lower layers as shown in figure (4).

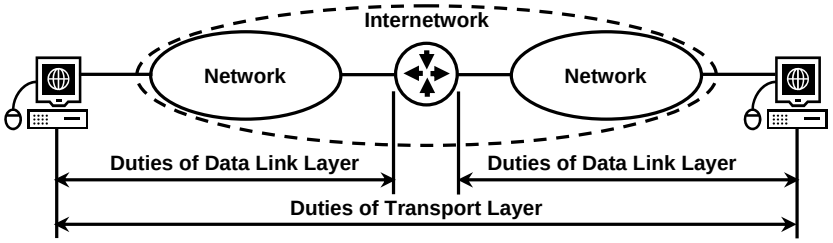


Figure (4) Transport Layer Compared with Data Link Layer.

Some Services Provided by Transport Layer Protocols:

1. **End-to-End Delivery:** The transport layer makes sure that the entire message (not just a single packet) arrives intact, thus, it oversees the end-to-end (source-to-destination) delivery of an entire message.

2. **Addressing:** To ensure accurate delivery from service point, we need another level of addressing. At the transport level, the protocol needs to know which upper-layer protocols are communicating.

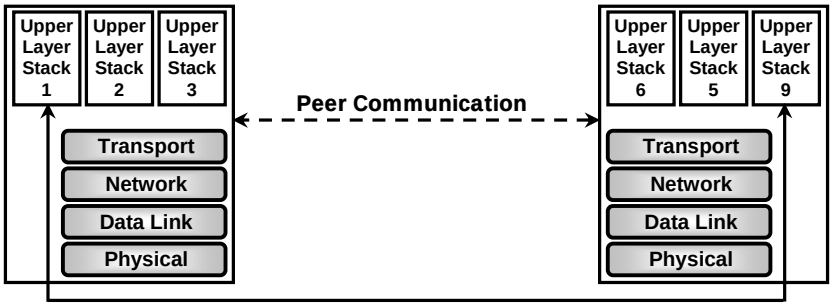


Figure (5) Service Points.

3. **Reliable Delivery:** Means lost segments are resent so the data is received complete. At the transport layer, reliable delivery has four aspects: error control, sequence control, loss control, and duplication control.
   a. **Error Control:** Mechanisms for error handling at this layer are based on error detection and retransmission with the error handling usually performed using algorithms implemented in software, such as checksum.

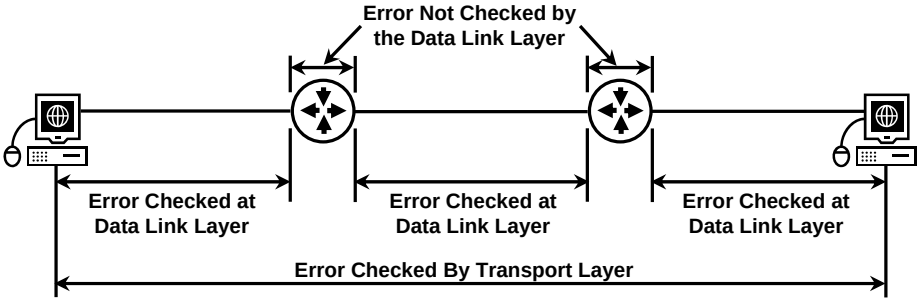


Figure (6) Transport and Data Link Layers Error Control.

b. Sequence Control: Ensures data is delivered sequentially in the same order delivery as it was sent. On the receiving end, it is responsible for ensuring that the previous pieces of a transmission arrive in the order intended.

Segmentation and Concatenation: When the size of the data is too long the transport layer protocol divides it into smaller, usable blocks, where, the dividing process is called segmentation. On the other hand, several small data can fit together into a single datagram or frame, the transport layer protocol combines them into a single data unit, where the combining process is called concatenation.

Sequence Numbers: Most transport layer services add a sequence number at the end of each segment. If a longer data unit has been segmented, the numbers indicate the order for reassembly.

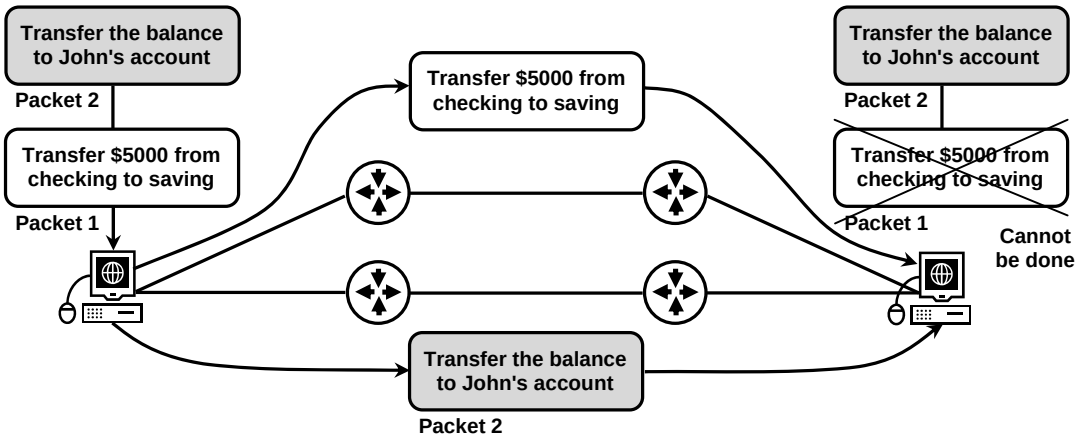


Figure (7) Transport Layer Sequence Control.

c. Loss Control: The transport layer ensures that all pieces of a transmission arrive at the destination, not just some of them. When data have been segmented for delivery, some segments may be lost in transit.

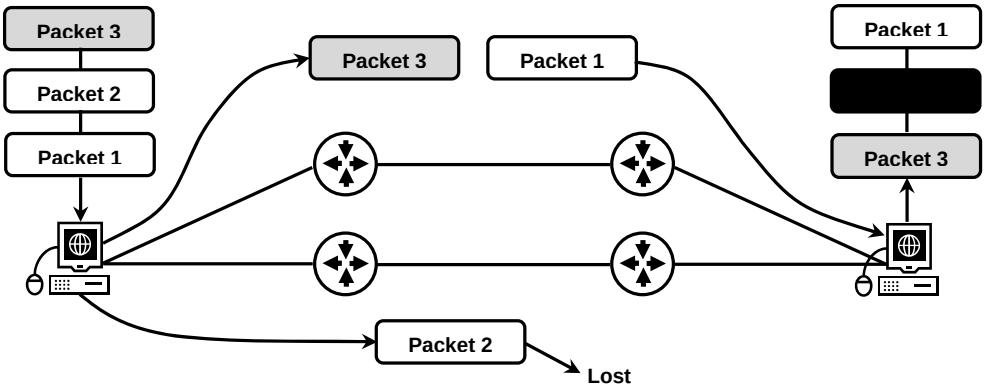


Figure (8) Transport Layer Loss Control.

d. Duplication Control: Transport layer functions must guarantee that no pieces of data arrive at the receiving system duplicated.

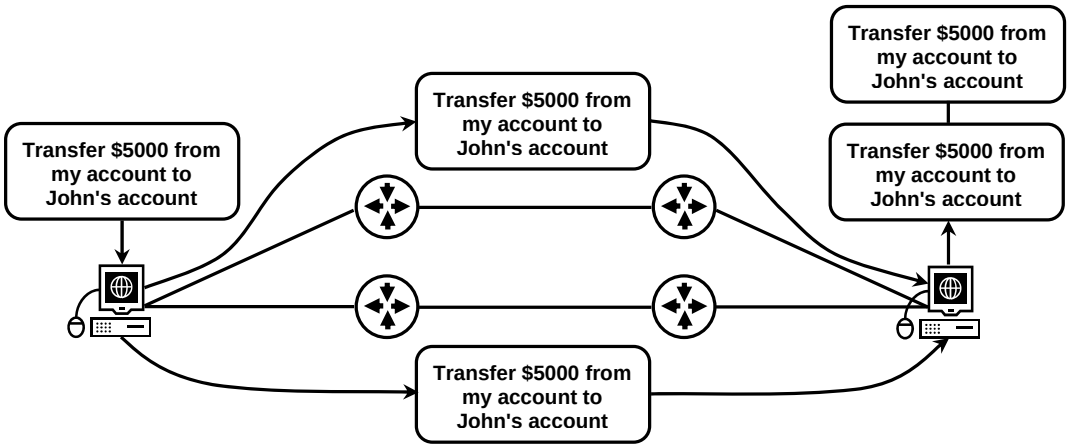


Figure (9) Transport Layer Duplication Control.

4. Flow Control: Manages data delivery if there is congestion on the host. Flow control at this layer is performed end-to-end rather than across a single link. Transport layer flow control also uses a sliding window protocol. However, the window at the transport layer can vary in size to accommodate buffer occupancy.

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

- The sender does not have to send a full window's worth of dat.
- An acknowledgment can expand the size of the window based on the sequence number of the acknowledged data segment.
- The size of the window can be increased or decreased by the receiver.
- The receiver can send an acknowledgment anytime it wants to.

5. Multiplexing: To improve transmission efficiency, the transport layer has the option of multiplexing, which occurs in two ways: upward or downward.

Upward: Meaning that many transport layers use the same network connection. The transport layer uses virtual circuits based on the services of the lower three layers. To make more cost-effective use of an established circuit, the transport layer can send

several transmissions bound for the same destination along the same path by multiplexing.

Downward: Meaning that one transport layer uses many network connections. Downward multiplexing allows the transport layer to split a single transmission among a number of different paths to improve throughput (speed of delivery). This option is useful when the underlying networks have low or slow capacity.

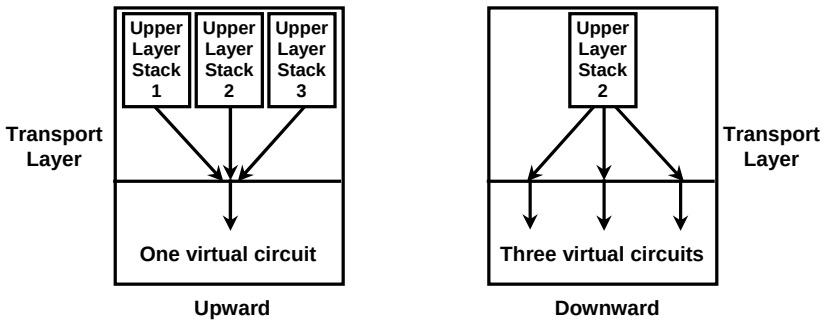


Figure (10) Transport Layer Multiplexing.

6.3 Transport Layer Connection

End-to-end delivery can be accomplished in either of two modes: connection-oriented or connectionless. Between these two, the connection-oriented mode is the more commonly used. A connection-oriented protocol establishes a virtual circuit or pathway through the Internet between the 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 acknowledgment process and retransmission of damaged or lost frames.

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

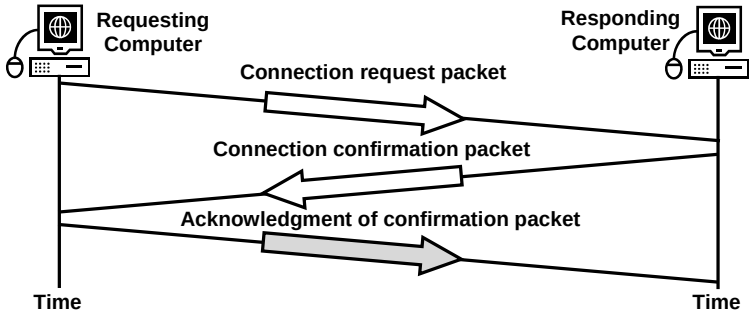


Figure (11) Transport Layer Connection Establishment.

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

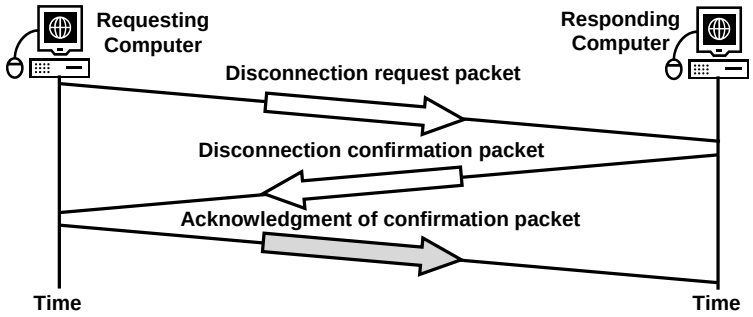


Figure (12) Transport Layer Connection Termination.

6.4 The OSI Transport Layer Protocol

1. Transport Classes: To avoid redundant services, the OSI defines five types of transport classes:

- **TP0.** Simple class.
- **TP1.** Basic error recovery class.
- **TP2.** Multiplexing class.
- **TP3.** Error recovery and multiplexing class.
- **TP4.** Error detection and recovery class.

Which class is used depends on the type of service required by the upper layers. The transport layer tries to match these requests to the available network services:

- **TP0 & TP2** are used with perfect network layers. A perfect network layer is one in which the number of packets that are lost or damaged is almost zero.
- **TP1 & TP3** are used with residual-error network layers. A residual-error network layer is one in which some percentage of errors are never corrected.
- **TP4** is used with unreliable network layers. TP4 provides fully reliable, full-duplex, connection-oriented services similar to TCP in the TCP/IP protocol suite.

2. Transport Protocol Data Unit (TPDU): The format of a transport protocol data unit (TPDU) is shown in Figure (13).

Length	Fixed parameters	Variable parameters	Data

Figure (13) Transport Protocol Data Unit (TPDU).

Length: The length field occupies the first byte and indicates the total number of bytes (excluding the length field itself) in the TPDU.

Fixed Parameters: The fixed parameters field contains parameters, or control fields, that are commonly present in all transport layer packets.

- **Code.** The code identifies the type of the data unit:

CR: Connection request.	**CC:** Connection confirm.
DR: Disconnect request.	**DC:** Disconnect confirm.
DT: Data.	**ED:** Expedited data.
AK: Data acknowledge.	**EA:** Expedited data acknowledge.
RJ: Reject.	**ER:** Error.

- **Source and Destination Reference.** The source and destination refence fields contain the addresses of the original sender and the ultimate destination of the packet.
- **Sequence Number.** Each segment is given a number that identifies its place in the sequence. Sequence numbers are used for acknowledgment, flow control, and recording of packets at the destination.
- **Credit Allocation.** Credit allocation enables a receiving station to tell the sender how many more data units may be sent before the sender must wait for an acknowledgment.

Variable Parameters: The variable parameters field contains parameters that occur infrequently. These control codes are used mostly for management (e.g., testing the reliability of a router).

Data: The data section of a TPDU may contain regular data or expedited data coming from the upper layers. Expedited data consist of a high-priority message that must be handled out of sequence. An urgent request (such as an interrupt command to a remote login) can supersede the incoming queue of the receiver and be processed ahead of packets have been received before it.

6.5 The TCP/IP Transport Layer Protocol

The transport layer is represented in TCP/IP by two protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). UDP is the simpler one; it provides non-sequenced transport functionality when reliability and security are less important than size and speed. Most applications, however, require reliable end-to-end delivery and so make use of TCP.

The transport protocols of the TCP/IP suite define a set of conceptual connections to individual processes called protocol ports or, more simply, ports. A protocol port is a destination point (usually a buffer) for storing data for use by a particular process. The interface between processes and their corresponding ports is provided by the operating system of the host.

Each port is defined by a positive integer address carried in the header of transport layer packet. An IP datagram use the host's 32-bit Internet address. A frame at the transport level uses the process port address of 16-bits, enough to allow the support of up to 65,536 (0 to 65536) ports.

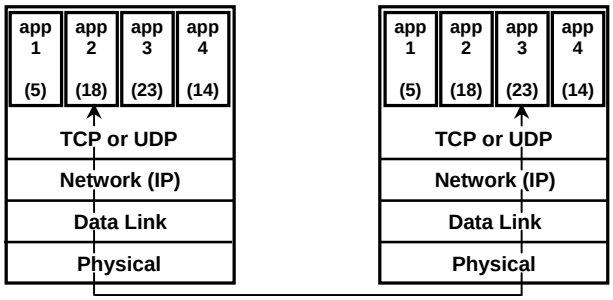


Figure (14) Port-to-Port Addresses.

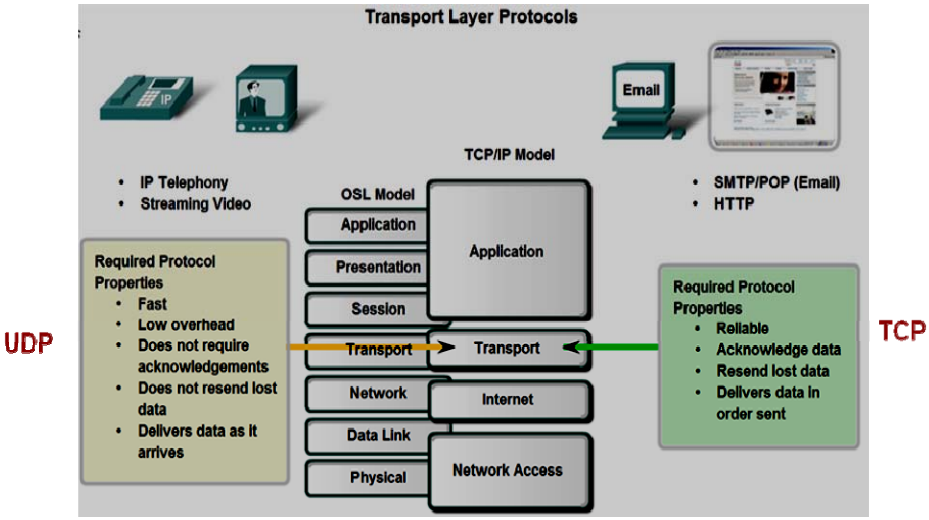


Figure (15) Application developers choose the appropriate Transport Layer protocol based on the nature of the application

1. **User Datagram Protocol (UDP):** It is an end-to-end transport level protocol that adds only port addresses, checksum error control, and information length to the data from the upper layer. The packet produced by the UDP is called a user datagram with the following fields:

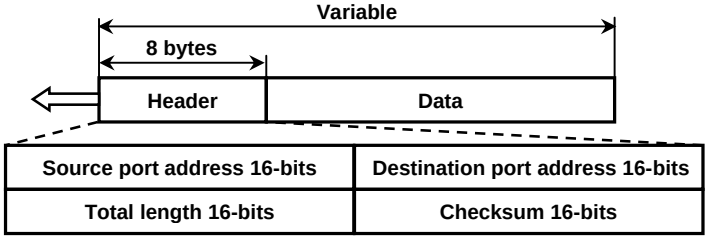


Figure (16) UDP Datagram Format.

- **Source Port Address.** The source port address is the address of the application program that has created the message.

- **Destination Port Address.** The destination port address is the address of the application program that will receive the message.
- **Total Length.** The total length field defines the total length of the user datagram in bytes.
- **Checksum.** The checksum is a 16-bit field used in error detection.

2. Transmission Control Protocol (TCP): TCP is a reliable stream transport port-to-port protocol. The term stream, in this context, means connection-oriented: a connection must be established between both ends of a transmission before either may transmit data. By creating this connection, TCP generates a virtual circuit between sender and receiver that is active for the duration of a transmission.

TCP is responsible for the reliable delivery of the entire stream of bits contained in the data unit originally generated by the sending application. Reliability is ensured by provision for error detection and retransmission of damaged frames; nall segments must be received and acknowledged before the transmission is considered complete and the virtual circuit is discarded.

The TCP Segment

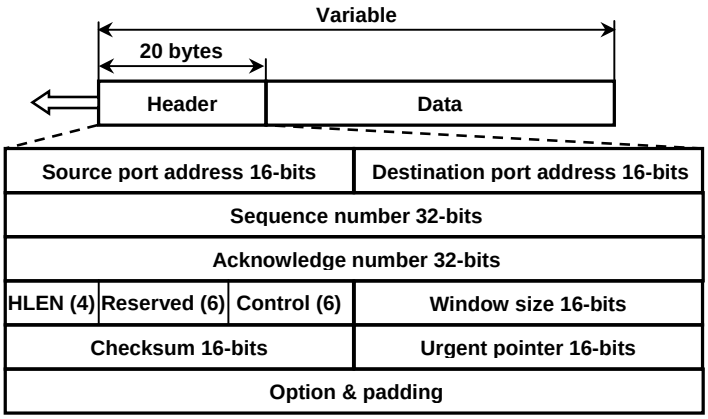


Figure (17) TCP Segment Format.

- **Source Port Address.** The source port address defines the application program in the source computer.
- **Destination Port Address.** The destination port address defines the application program in the destination computer.
- **Sequence Number.** A stream of data from the application program may be divided into two or more TCP segments. The sequence number field shows the position of the data in the original data stream.
- **Acknowledgment Number.** Its 32-bit acknowledgment number is used to acknowledge the receipt of data from the other communicating device.

- **Header length (HLEN).** The 4-bit HLEN field indicates the number of 32-bit (4-byte) words in the TCP header. The 4-bits can define a number up to 15. This is multiplied by 4 to give the total number of bytes in the header.
- **Reserved.** A 6-bit field is reserved for future use.
- **Control.** Each bit of the 6-bit control field functions individually and independently. A bit can either define the use of a segment or serve as a validity check for other fields (Urgent, ACK, PSH, Reset, SYN, & FIN).
- **Window Size.** The window is a 16-bit field that defines the size of sliding window.
- **Checksum.** The checksum is a 16-bit field used in error detection.
- **Urgent Pointer.** Its value is valid only if the URG bit in the control field is set.
- **Options and Padding.** They are used to convey additional information to the receiver or for alignment purposes.

6.6 Port Addressing

Identify how a port number is represented and describe the role port numbers play in the TCP and UDP protocols.

Notes:

1. The port numbers Identified communicated applications
2. The port numbers Identified conversation between hosts
3. Both TCP & UDP use source & destination port numbers

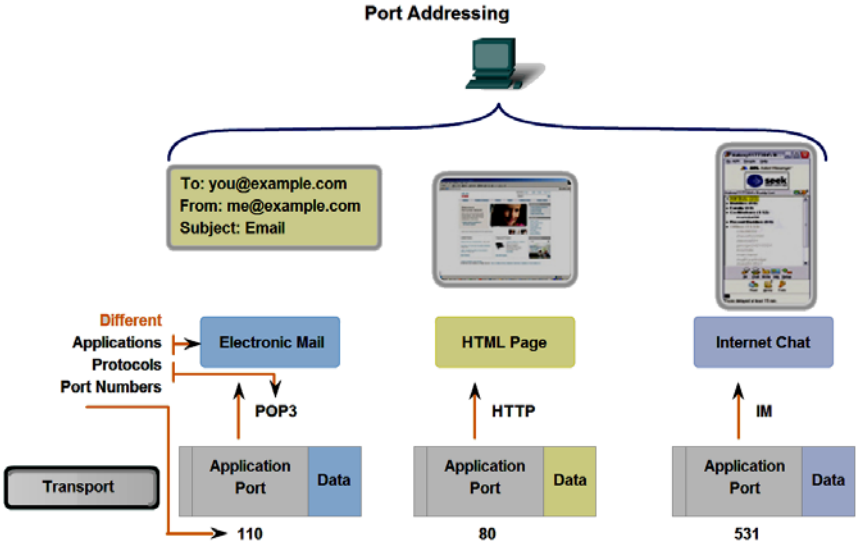


Figure (18) Data for different applications is directed to the correct application because each application has a unique port number

Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered TCP Ports:
1863 MSN Messenger
8008 Alternate HTTP
8080 Alternate HTTP

Well Known TCP Ports:
21 FTP
23 Telnet
25 SMTP
80 HTTP
110 POP3
194 Internet Relay Chat (IRC)
443 Secure HTTP (HTTPS)

Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered UDP Ports:
1812 RADIUS Authentication Protocol
2000 Cisco SCCP (VoIP)
5004 RTP (Voice and Video Transport Protocol)
5060 SIP (VoIP)

Well Known UDP Ports:
69 TFTP
520 RIP

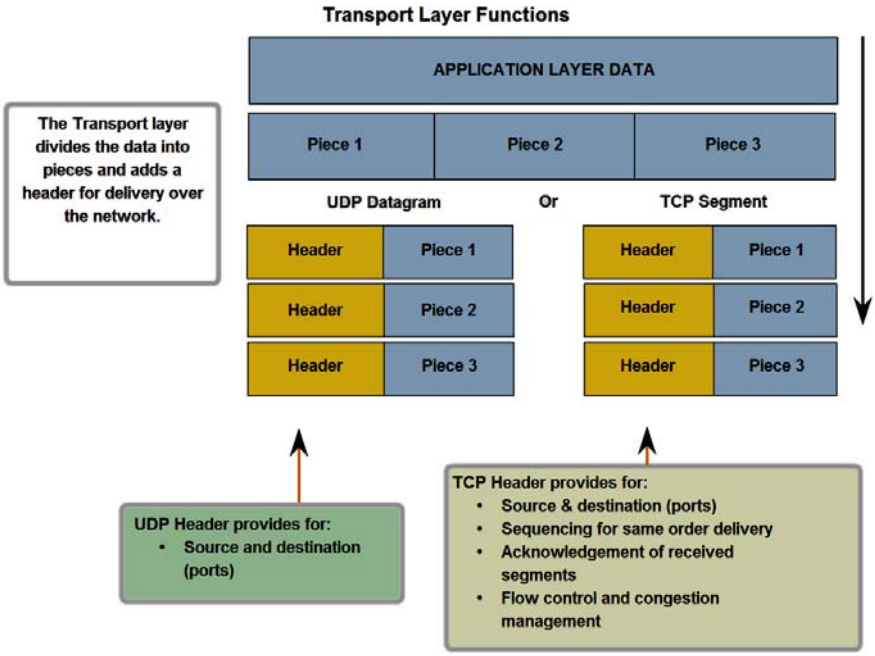
Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered TCP/UDP Common Ports:
1433 MS SQL
2948 WAP (MMS)

Well Known TCP/UDP Common Ports:
53 DNS
161 SNMP
531 AOL Instant Messenger, IRC

Dividing application data into pieces both ensures that data is transmitted within the limits of the media and that data from different applications can be multiplexed on to the media.



6.7 Application and Operation of TCP Mechanisms

1. Each application process running on the server is configured to use a port number, either by default or manually by a system administrator.
2. An individual server cannot have two services assigned to the same port number within the same Transport layer services

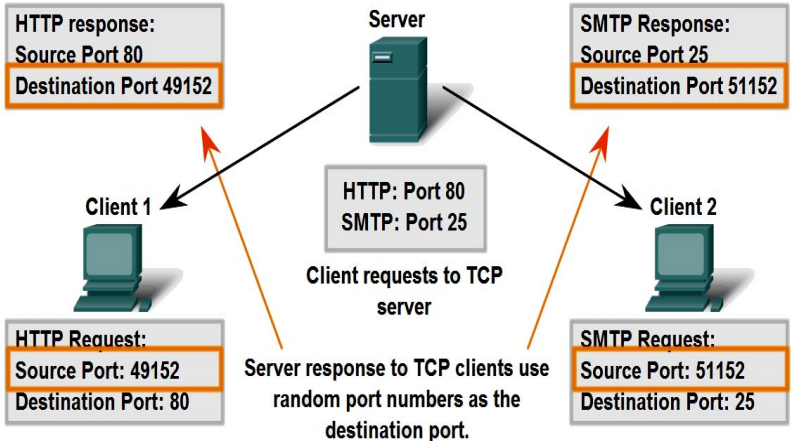


Figure (19) Clients sending TCP requests

6.8 Managing TCP Sessions

Describe how TCP sequence numbers are used to reconstruct the data stream with segments placed in the correct order.

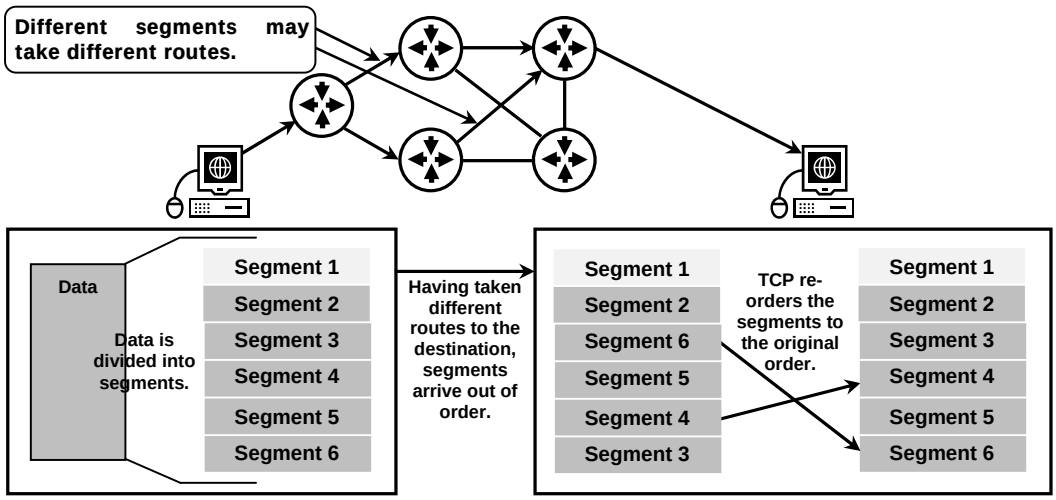


Figure (20) TCP segments are re-ordered at the destination

6.9 Confirming Receipt of Segments

The segment header sequence number and acknowledgement number are used together to confirm receipt of the bytes of data contained in the segments.

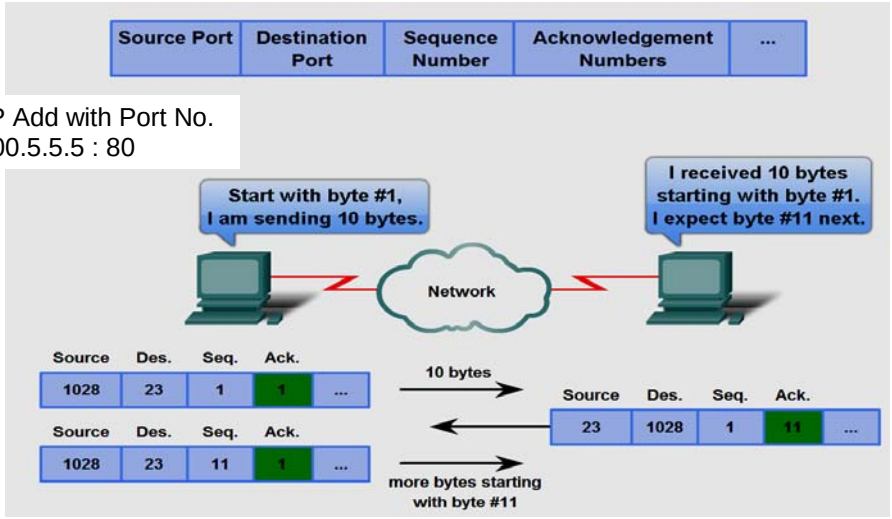
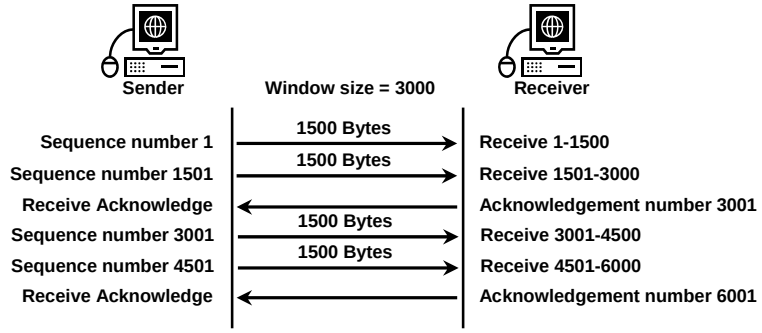


Figure (21) Acknowledgement of TCP segments

6.10 Flow Control

Flow control assists the reliability of TCP transmission by adjusting the effective rate of data flow between the two services in the session. Window Size field in the TCP header specifies the amount of data that can be transmitted before an acknowledgement must be received.

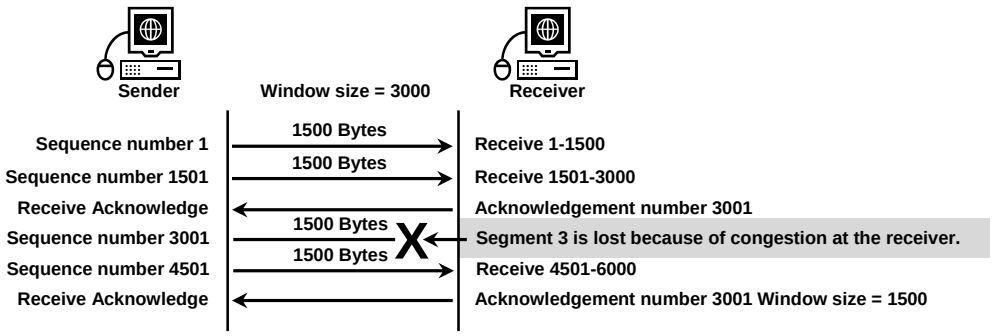


The window size determines the number of bytes sent before an acknowledgement is expected.
The acknowledgement number is the number of the next expected byte.

Figure (22) TCP segment acknowledgment and window size

6.11 Managing TCP Sessions

The relationship between window size, data loss and congestion during a session (when receiver is overloaded)



If segments are lost because of congestion, the receiver will acknowledge the last received sequential segment and reply with a reduced window size.

Figure (23) TCP congestion and flow control

- When source of data (server) reaches time out, it will resend the data that needs to be acknowledged.
- Connection Services use TCP
- Connectionless Services use UDP

6.12 UDP Protocol

Describe the characteristics of the UDP protocol and the types of communication for which it is best suited

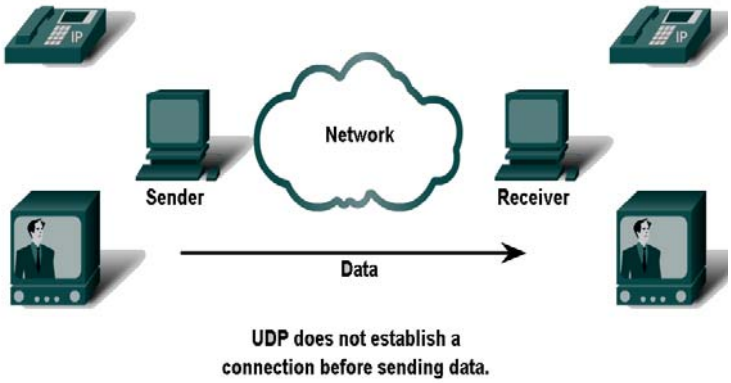


Figure (24) UDP low overhead data transport

Describe in detail the process specified by the UDP protocol to reassemble PDUs at the destination device

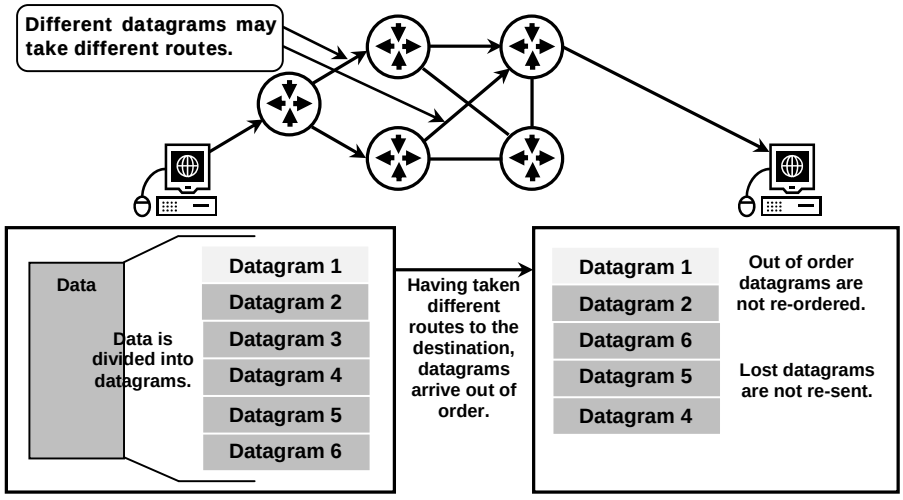
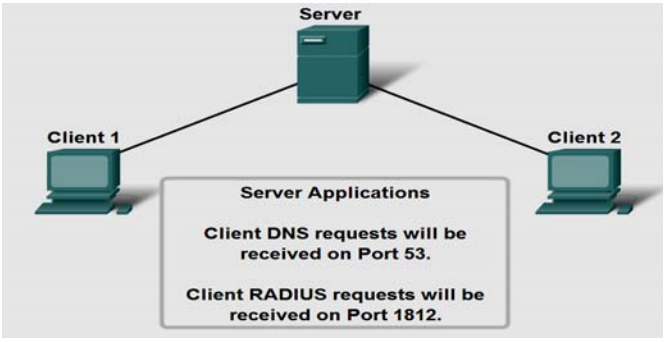


Figure (25) UDP: connectionless and unreliable

Describe how servers use port numbers to identify a specified application layer process and direct segments to the proper service or application



Client requests to servers have well known ports numbers as the destination port.

Figure (26) UDP server listening for requests

Trace the steps as the UDP protocol and port numbers are utilized in client-server communication.

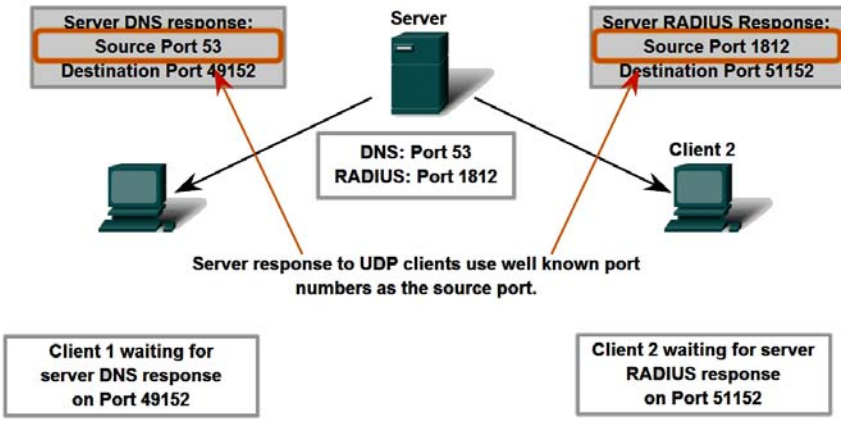


Figure (27) Client sending UDP requests

Although the total amount of UDP traffic found on a typical network is often relatively low, key Application layer protocols that use UDP include:

- Domain Name System (DNS)
- Simple Network Management Protocol (SNMP)
- Dynamic Host Configuration Protocol (DHCP)
- Routing Information Protocol (RIP)
- Trivial File Transfer Protocol (TFTP)
- Online games