Chapter 7: Transport Layer

Introduction to Networking

Cisco | Networking Academy®
Mind Wide Open™

1

Chapter 7: Objectives

- Describe the purpose of the transport layer in managing the transportation of data in end-to-end communication.
- Describe characteristics of the TCP and UDP protocols, including port numbers and their uses.
- Explain how TCP session establishment and termination processes facilitate reliable communication.
- Explain how TCP protocol data units are transmitted and acknowledged to guarantee delivery.
- Explain the UDP client processes to establish communication with a server.
- Determine whether high-reliability TCP transmissions, or non-guaranteed UDP transmissions, are best suited for common applications.

2

7.1: Transport Layer Protocols

Cisco Networking Academy®
Mind Wide Open™

3

Role of the Transport Layer

The transport layer is responsible for establishing a temporary communication session between two applications and delivering data between them.

TCP/IP uses two protocols to achieve this:

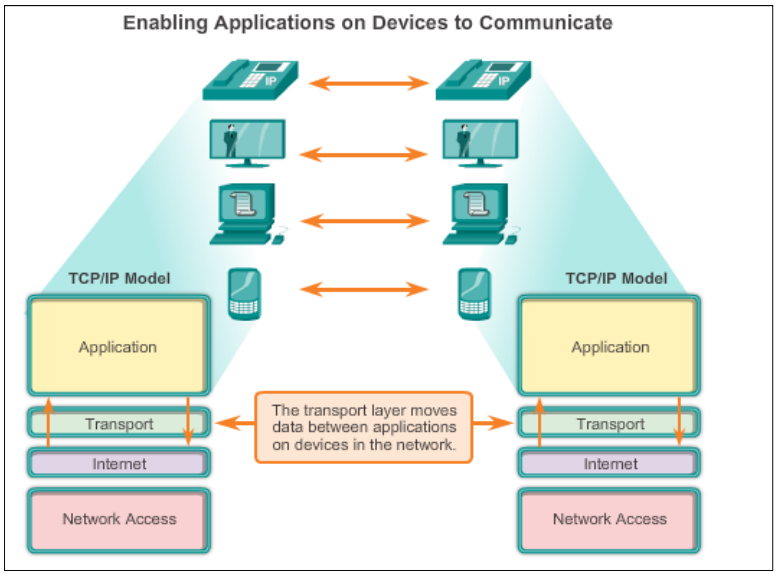
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Primary Responsibilities of Transport Layer Protocols

- Tracking the individual communication between applications on the source and destination hosts
- Segmenting data for manageability and reassembling segmented data into streams of application data at the destination
- Identifying the proper application for each communication stream

4

Role of the Transport Layer (Cont.)



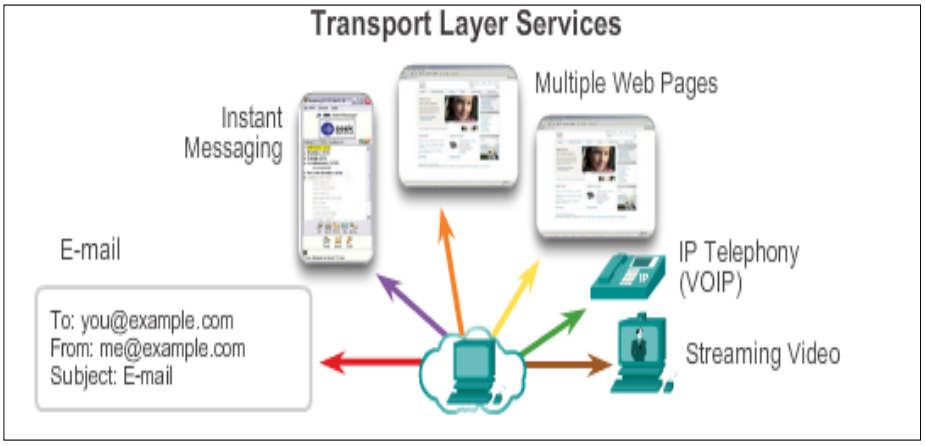
Tracking Individual Conversations

- At the transport layer, each particular set of data flowing between a source application and a destination application.
- A host may have multiple applications that are communicating across the network simultaneously.
- Each of these applications communicates with one or more applications on one or more remote hosts.
- It is the responsibility of the transport layer to maintain and track these multiple conversations.

Segmenting the Data

- Most networks have a limitation on the amount of data that can be included in a single packet.
- Transport layer protocols have services that segment the application data into blocks of data that are an appropriate size.
- Header added to each segment to identify it. This header is used to track the data stream.
- At the destination, the transport layer must be able to reconstruct the pieces of data into a complete data stream.

Conversation Multiplexing (Cont.)

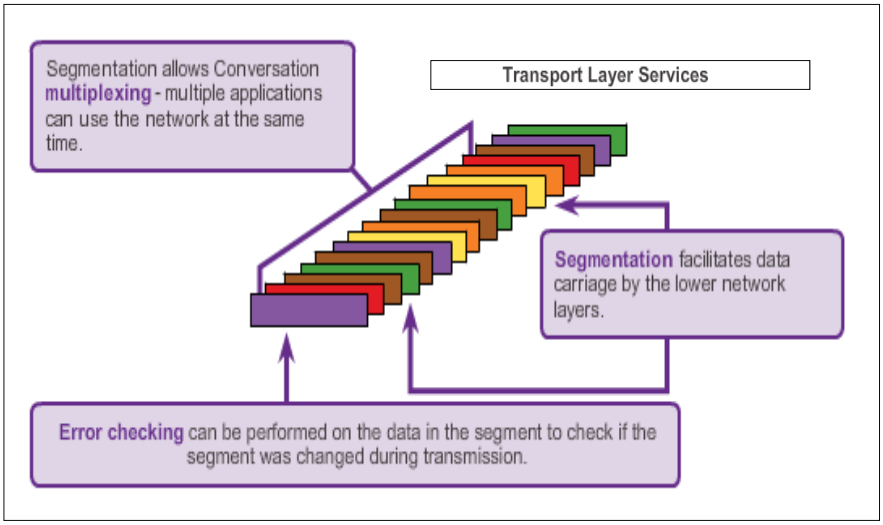


Conversation Multiplexing

- There may be many applications or services running on each host in the network.
- Enables many different communications, from many different users, to be interleaved (multiplexed) on the same network, at the same time.
- Provides the means to both send and receive data when running multiple applications.
- To pass data streams to the proper applications, the transport layer must identify the target application.
- To accomplish this, the transport layer assigns each application an identifier. This identifier is called a ***port number***.

9

Conversation Multiplexing (Cont.)



10

Transport Layer Reliability

TCP/IP provides two transport layer protocols, **TCP and UDP.**

TCP

- Provides reliable delivery ensuring that all of the data arrives at the destination.
- Uses acknowledged delivery and other processes to ensure delivery
- Makes larger demands on the network – more overhead.

UDP

- Provides just the basic functions for delivery – no reliability.
- Less overhead.

TCP or UDP

- There is a trade-off between the value of reliability and the burden it places on the network.
- Application developers choose the transport protocol based on the requirements of their applications.

11

Introducing TCP

- Connection-oriented – Creates a session between the source and destination
- Reliable delivery – Retransmits lost or corrupt data
- Ordered data reconstruction – Reconstructs numbering and sequencing of segments
- Flow control – Regulates the amount of data transmitted
- Stateful protocol – Tracks the session

TCP Segment

Bit(0)	Bit(15)	Bit(16)	Bit(31)	
Source Port (16)		Destination Port (16)		20 Bytes
Sequence Number (32)				
Acknowledgement Number (32)				
Header Length(4) Reserved(6) Code Bits(6)		Window (16)		
Checksum (16)		Urgent (16)		
Options (0 or 32 if any)				
APPLICATION LAYER DATA (Size varies)				

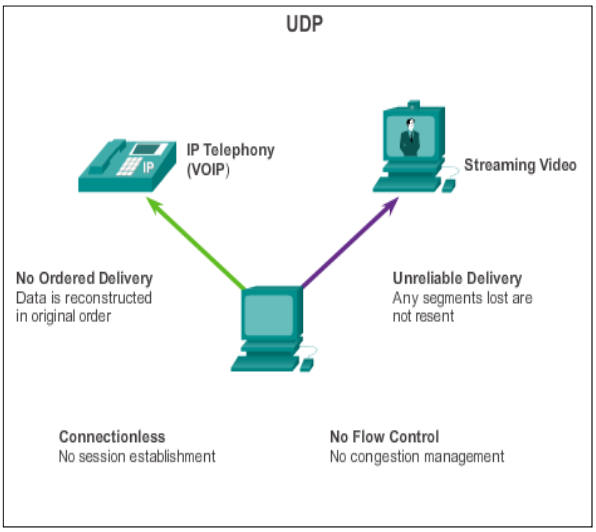
12

Introducing UDP

- Connectionless
- Unreliable delivery
- No ordered data reconstruction
- No flow control
- Stateless protocol

Applications that use UDP:

- Domain Name System (DNS)
- Video Streaming
- VoIP

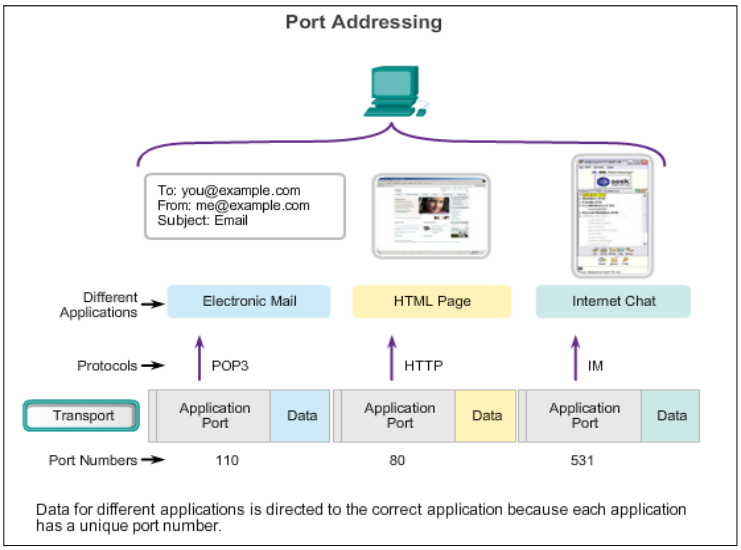


13

Separating Multiple Communications

TCP and UDP use port numbers to differentiate between applications.

.



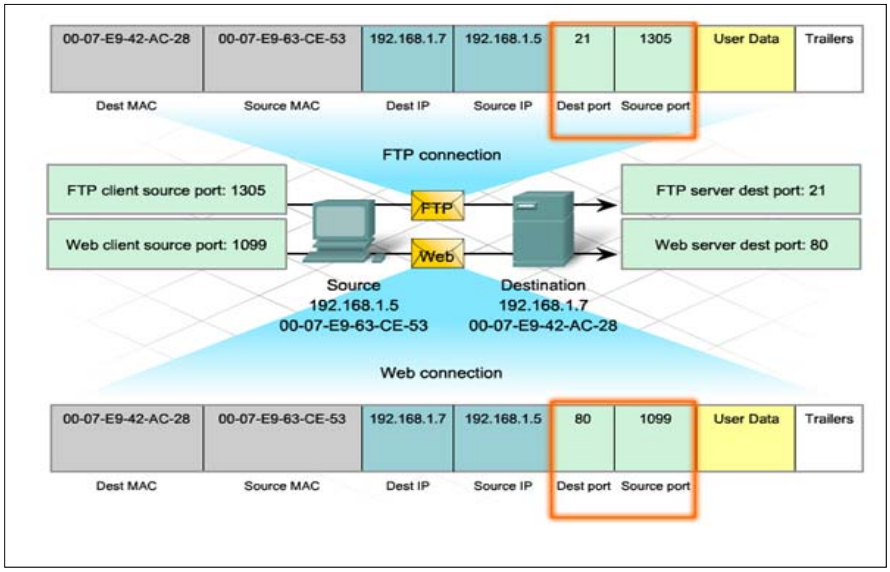
14

TCP and UDP Port Addressing

- When a message is delivered using either TCP or UDP, the protocols and services requested are identified by a port number.
- A port is a numeric identifier within each segment that is used to keep track of specific conversations and destination services requested.
- Every message that a host sends contains both a **source** and **destination** port. The source and destination ports are placed within the segment.
- The segments are then encapsulated within an IP packet. The IP packet contains the IP address of the source and destination.
- The combination of the source and destination IP addresses and the source and destination port numbers is known as a **socket**.
- The socket is used to identify the server and service being requested by the client.

15

TCP and UDP Port Addressing



16

Types of Port Numbers

There are different types of port numbers, as shown in Figure:

Well-known Ports (Numbers 0 to 1023)

- These numbers are reserved for services and applications.
- They are commonly used for applications such as HTTP (web server), Internet Message Access Protocol (IMAP)/Simple Mail Transfer Protocol (SMTP) (email server) and Telnet.

Registered Ports (Numbers 1024 to 49151)

- These port numbers are assigned to user processes or applications.

Dynamic or Private Ports (Numbers 49152 to 65535)

- Also known as ephemeral ports, these are usually assigned dynamically to client applications when the client initiates a connection to a service.

17

TCP and UDP Port Addressing (Cont.)

Port Numbers

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65533	Private and/or Dynamic Ports

Registered TCP Ports:

1863	MSN Messenger
2000	Cisco SCCP (VoIP)
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)

18

TCP and UDP Port Addressing (Cont.)

Registered UDP Ports:
1812 RADIUS Authentication Protocol
5004 RTP (Voice and Video Transport Protocol)
5040 SIP (VoIP)

Well Known UDP Ports:
69 TFTP
520 RIP

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

19

TCP and UDP Port Addressing (Cont.)

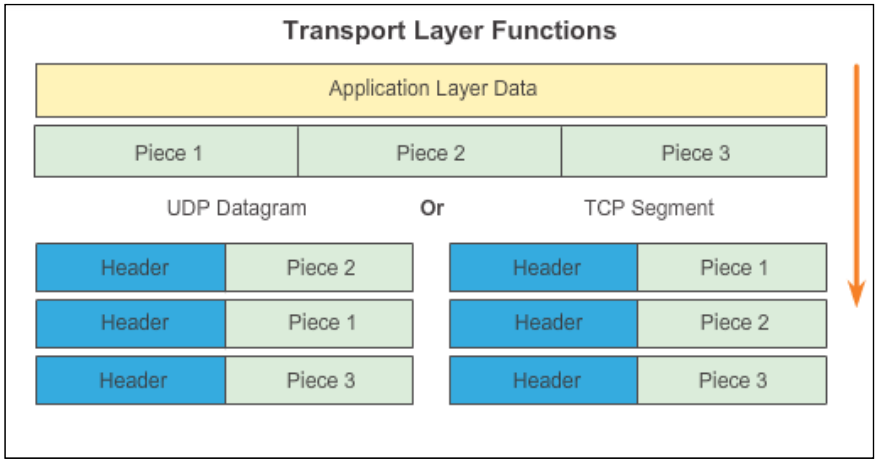
Netstat is used to examine TCP connections that are open and running on a networked host.

```
C:\>netstat

Active Connections

Proto   Local Address    Foreign Address          State
TCP     kenpc:3126       192.168.0.2:netbios-ssn  ESTABLISHED
TCP     kenpc:3158       207.138.126.152:http     ESTABLISHED
TCP     kenpc:3159       207.138.126.169:http     ESTABLISHED
TCP     kenpc:3160       207.138.126.169:http     ESTABLISHED
TCP     kenpc:3161       sc.msn.com:http          ESTABLISHED
TCP     kenpc:3166       www.cisco.com:http       ESTABLISHED

C:\>
```

20

TCP and UDP Segmentation

The transport layer divides the data into pieces and adds a header for delivery over the network



21

CISCO

7.2 TCP and UDP

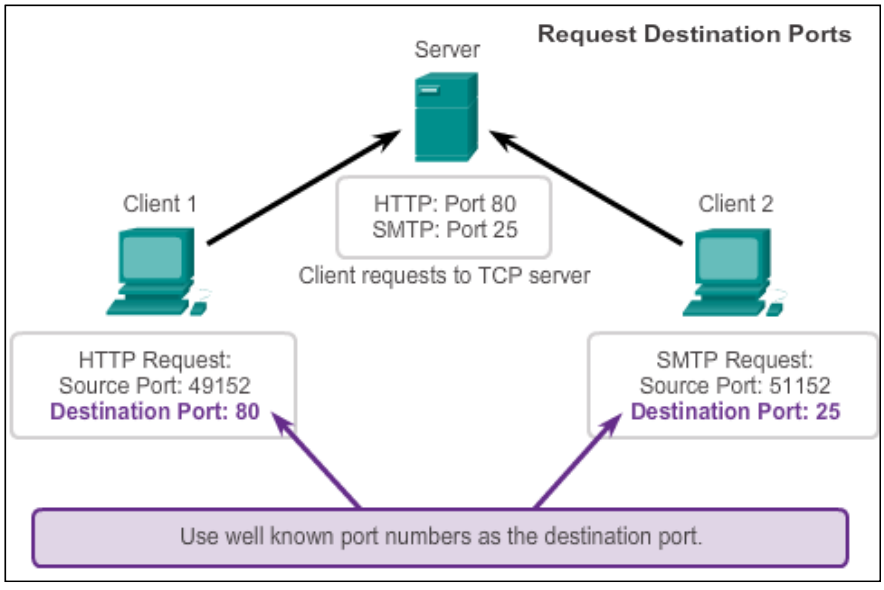
Cisco Networking Academy®
Mind Wide Open™

22

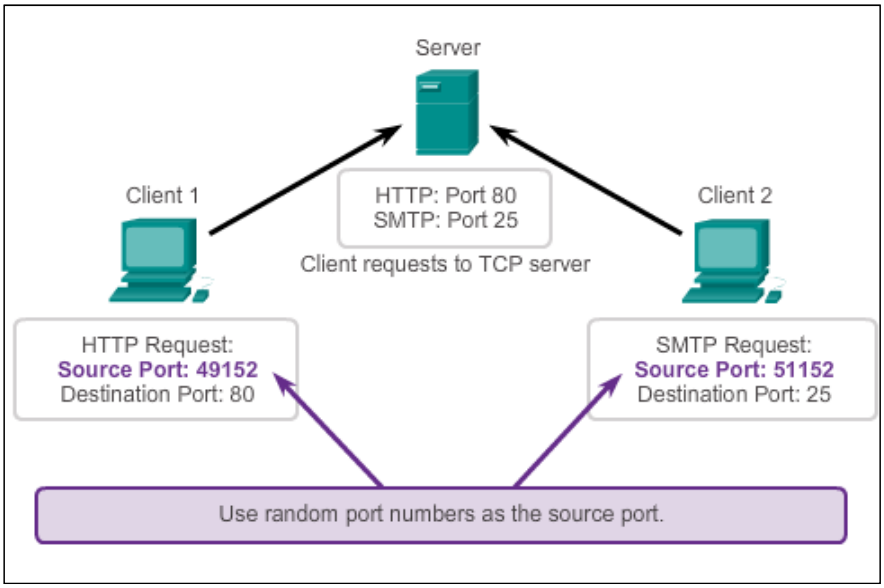
TCP Server Processes

- Each application process running on the server is configured to use a port number, either by default or manually by a system administrator.
- An individual server cannot have two services assigned to the same port number within the same transport layer services.
- There can be many simultaneous ports open on a server, one for each active server application.

TCP Server Processes



TCP Server Processes (Cont.)



25

TCP Connection, Establishment and Termination

Three-Way Handshake

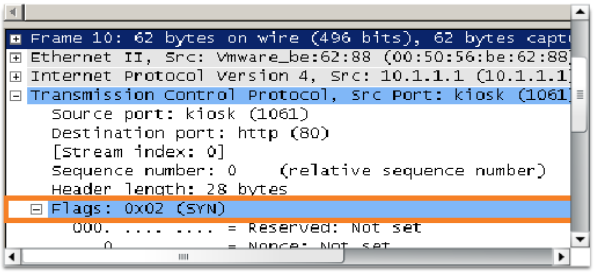
- Establishes that the destination device is present on the network.
- Verifies that the destination device has an active service and is accepting requests on the destination port number that the initiating client intends to use for the session.
- Informs the destination device that the source client intends to establish a communication session on that port number.

26

TCP Three-Way Handshake – Step 1

Step 1: The initiating client requests a client-to-server communication session with the server

TCP 3-Way Handshake (SYN)



A protocol analyzer shows initial client request for session in frame 10

TCP segment in this frame shows:

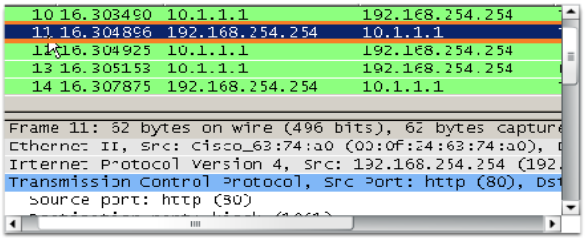
- SYN flag set to validate an Initial Sequence Number
- Randomized sequence number valid (relative value is 0)
- Random source port 1061
- Well-known destination port is 80 (HTTP port) indicates web server (httpd)

27

TCP Three-Way Handshake – Step 2

Step 2: The server acknowledges the client-to-server communication session and requests a server-to-client communication session.

TCP 3-Way Handshake (SYN, ACK)



A protocol analyzer shows server response in frame 11

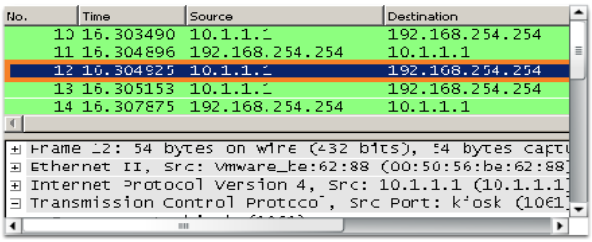
- ACK flag set to indicate a valid Acknowledgement number
- Acknowledgement number response to initial sequence number as relative value of 1
- SYN flag set to indicate the Initial Sequence Number for the server to client session
- Destination port number of 1061 to corresponding to the clients source port
- Source port number of 80 (HTTP) indicating the web server service (httpd)

28

TCP Three-Way Handshake – Step 3

Step 3: The initiating client acknowledges the server-to-client communication session.

TCP 3-Way Handshake (ACK)



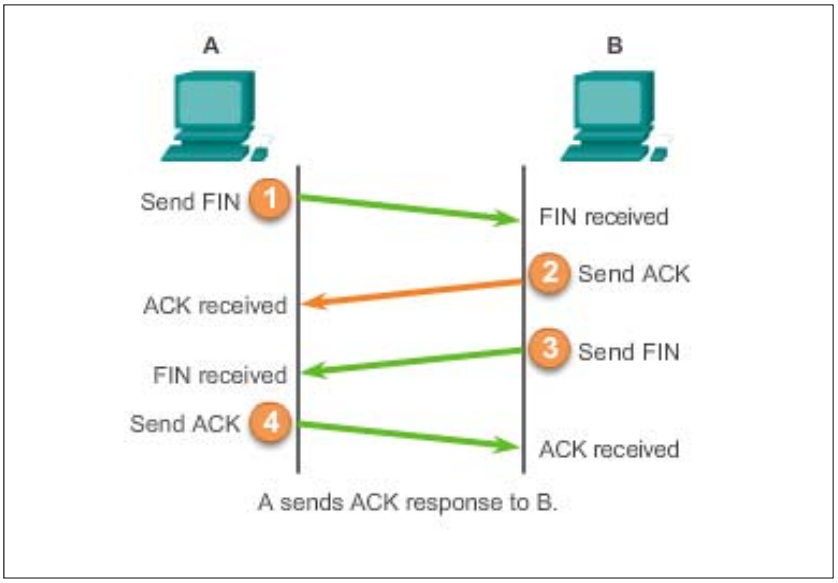
A protocol analyzer shows client response to session in frame 12

The TCP segment in this frame shows:

- ACK flag set to indicate a valid Acknowledgement number
- Acknowledgement number response to initial sequence number as relative value of 1
- Source port number of 1061 to corresponding
- Destination port number of 80 (HTTP) indicating the web server service (httpd)

29

TCP Session Termination



30

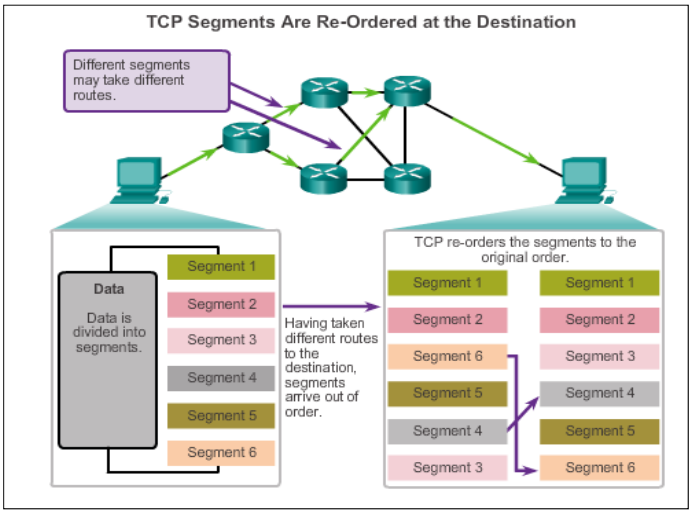
TCP Reliability – Ordered Delivery

- When services send data using TCP, segments may arrive at their destination out of order.
- During session setup, an initial sequence number (ISN) is set.
- As data is transmitted during the session, the sequence number is incremented by the number of bytes that have been transmitted.
- This data byte tracking enables each segment to be uniquely identified and acknowledged. Missing segments can be identified.
- Segment sequence numbers enable the reliability by indicating how to reassemble and reorder received segments, as shown in the figure.

31

TCP Reliability – Ordered Delivery

Sequence numbers are used to reassemble segments into their original order.



32

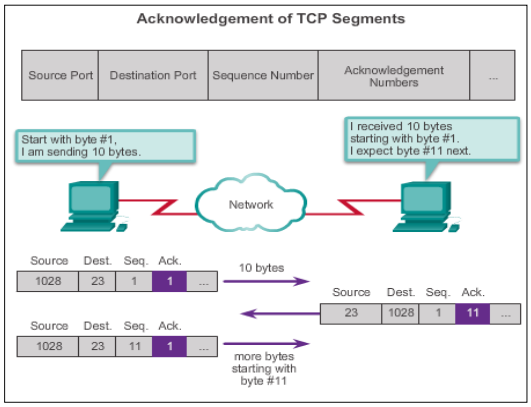
Acknowledgement and Window size

- TCP ensures that each segment reaches its destination.
  The sequence (SEQ) number and acknowledgement (ACK) number are used together to confirm receipt of the bytes of data.
- The SEQ number indicates the relative number of bytes that have been transmitted in this session.
- TCP uses the ACK number sent back to the source to indicate the next byte that the receiver expects to receive.
- The source is informed that the destination has received all bytes indicated by the ACK number.
- The amount of data that a source can transmit before an acknowledgement must be received is called the **window size**, which enables the management of lost data and flow control.

33

Acknowledgement and Window Size

The sequence number and acknowledgement number are used together to confirm receipt.



The window size is the amount of data that a source can transmit before an acknowledgement must be received.

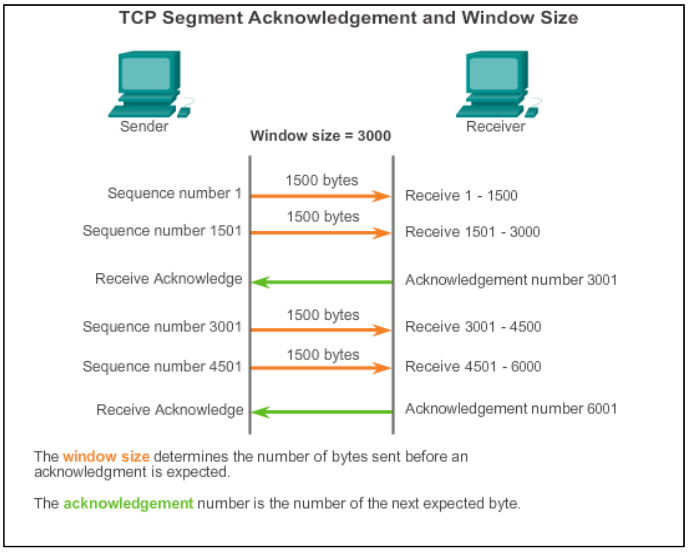
34

TCP Flow Control- Window Size and Acknowledgements

- Flow control helps maintain the reliability of TCP transmission by adjusting the rate of data flow between source and destination for a given session.
- Flow control is accomplished by limiting the amount of data segments forwarded by requiring acknowledgments of receipt prior to sending more.
- TCP uses window sizes to attempt to manage the rate of transmission to the maximum flow that the network and destination device can support, while minimizing loss and retransmissions.

35

TCP Flow Control- Window Size and Acknowledgements

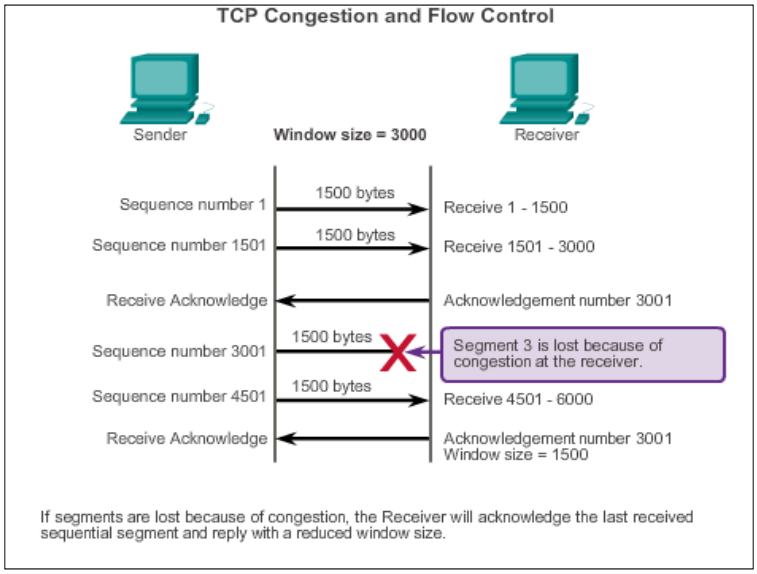


36

TCP Flow Control – Congestion Avoidance

- When network resources are constrained, TCP can reduce the window size to require that received segments be acknowledged more frequently.
- As shown in the figure, if a receiving host has congestion, it may respond to the sending host with a segment that specifies a reduced window size.
- After a period of transmission with no data losses or constrained resources, the receiver begins to increase the window field, which reduces the overhead on the network.

TCP Flow Control – Congestion Avoidance



UDP Low Overhead vs. Reliability

UDP

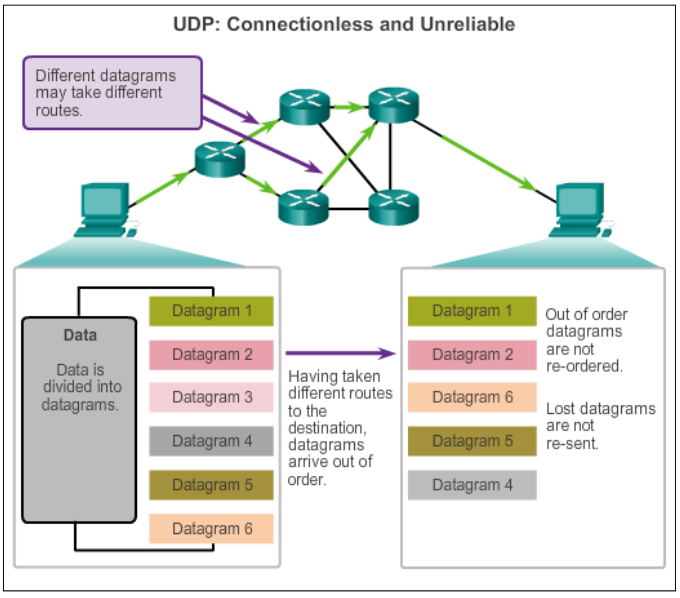
- Simple protocol that provides the basic transport layer function
- Used by applications that can tolerate small loss of data
- Used by applications that cannot tolerate delay

Used by

- DNS
- Simple Network Management Protocol (SNMP)
- Dynamic Host Configuration Protocol (DHCP)
- Trivial File Transfer Protocol (TFTP)
- IP telephony or VoIP
- Online games

39

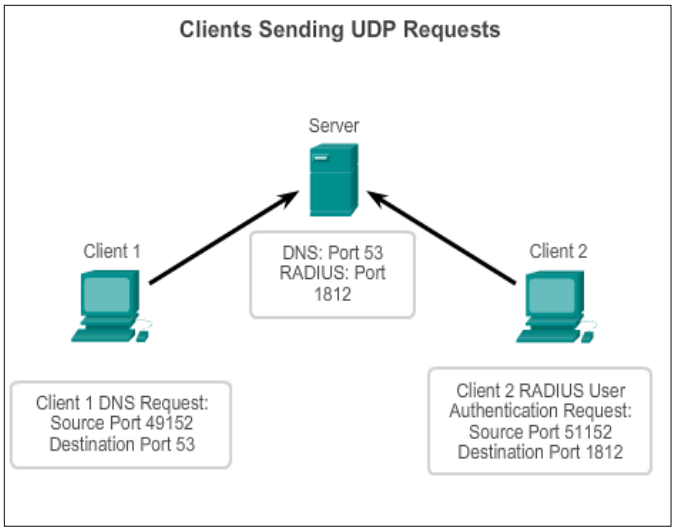
Datagram Reassembly



40

UDP Server and Client Processes

- UDP-based server applications are assigned well-known or registered port numbers.
- UDP client process randomly selects port number from range of dynamic port numbers as the source port.



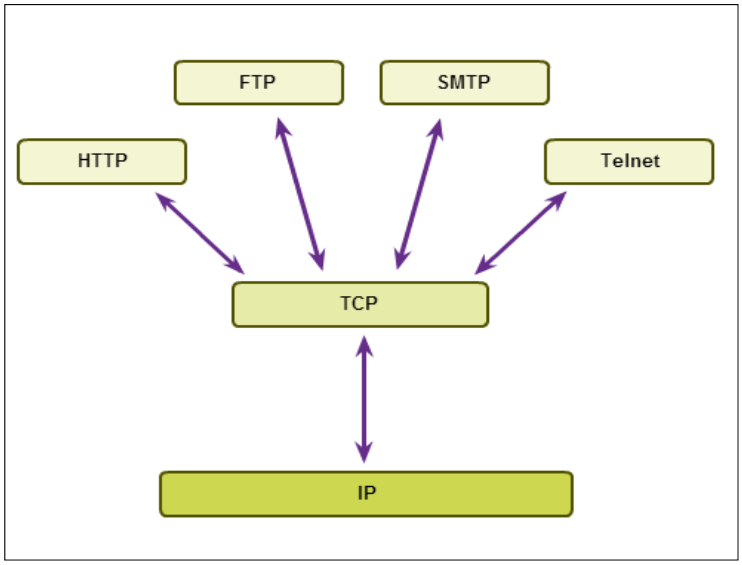
41

Applications that use TCP

- TCP best suited for applications that need reliable transport and can tolerate some delay.
- As shown in the figure, some examples of well-known applications that use TCP include:
  - Hypertext Transfer Protocol (HTTP)
  - File Transfer Protocol (FTP)
  - Simple Mail Transfer Protocol (SMTP)
  - Telnet

42

Applications that use TCP



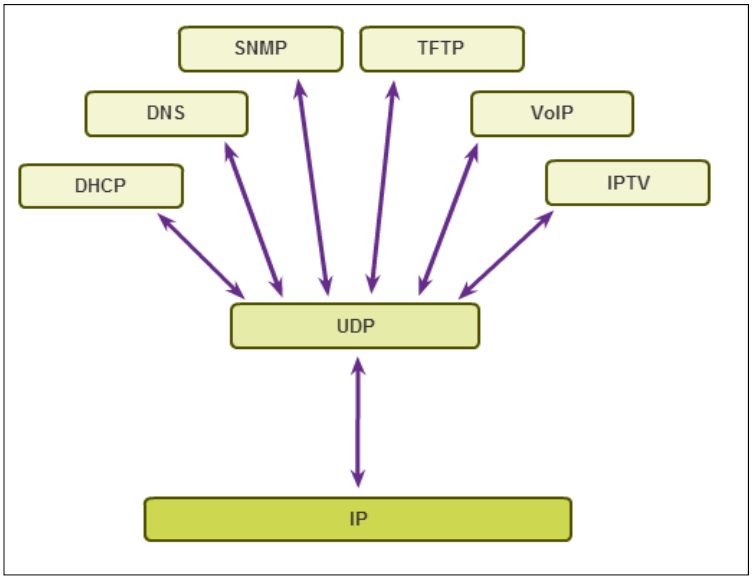
43

Applications that use UDP

- UDP are best suited applications that can tolerate some data loss, but require little or no delay; where reliability is not required or can be handled by the application.
- Many video and multimedia applications, such as VoIP and Internet Protocol Television (IPTV) use UDP.
  - These applications can tolerate some data loss with little or no noticeable effect.
- Some applications handle reliability themselves. TFTP is one example of this type of protocol. TFTP has its own mechanisms for flow control, error detection, acknowledgements and error recovery.
- Other types of applications well suited for UDP are those that use simple request and reply transactions.
  - This is where a host sends a request and may not receive a reply.
  - One example of this type of application is DHCP.

44

Applications That Use UDP



45

Chapter 7: Summary

In this chapter, you learned:

- The role of the transport layer is to provide three main services: multiplexing, segmentation and reassembly, and error checking. It does this by:
  - Dividing data received from an application into segments.
  - Adding a header to identify and manage each segment.
  - Using the header information to reassemble the segments back into application data.
  - Passing the assembled data to the correct application.
- How TCP and UDP operate and which popular applications use each protocol.
- Ports provide a “tunnel” for data to get from the transport layer to the appropriate application at the destination.

46