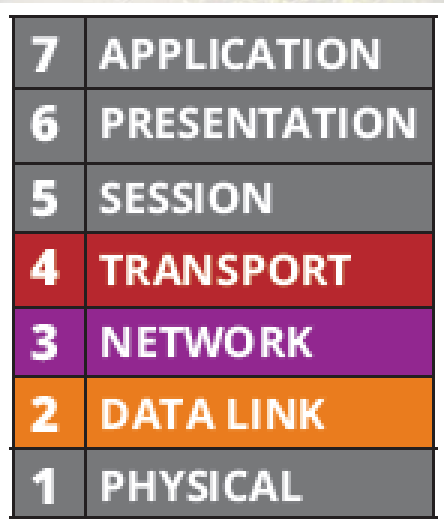
Network Protocols and Routing

Objectives

- Describe the functions of core TCP/IP protocols
- Identify how each protocol's information is formatted in a TCP/IP message
- Explain how routers manage internetwork communications
- Employ various TCP/IP utilities for network discovery and troubleshooting

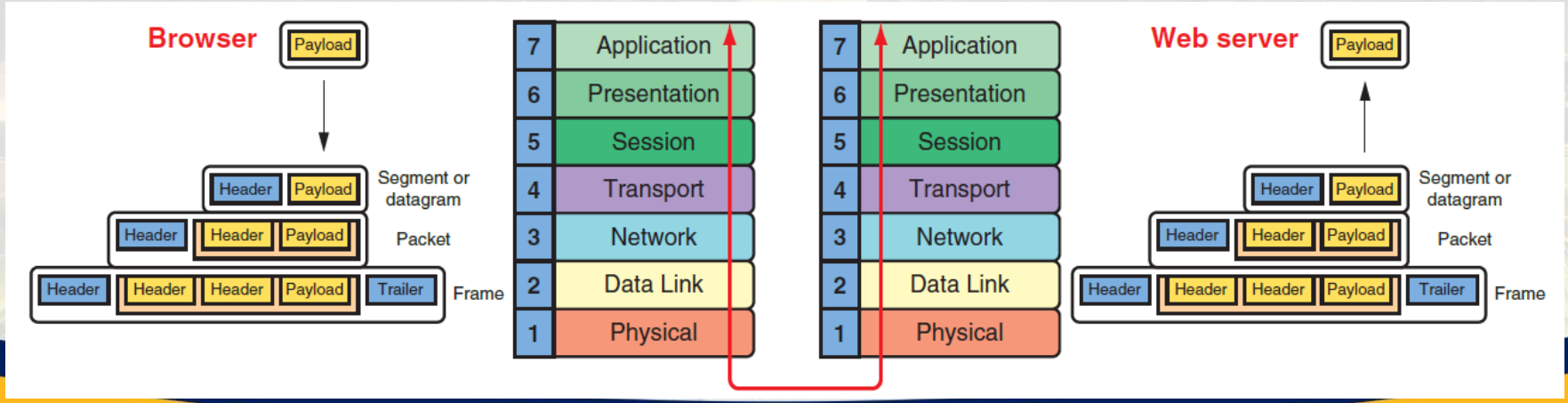
TCP/IP Core Protocols

- Explain the purposes and uses of ports and protocols.
- Explain the concepts and characteristics of routing and switching.
- Given a scenario, determine the appropriate placement of networking devices on a network and install/configure them.
- Explain the purposes of virtualization and network storage technologies.
- Given a scenario, use the appropriate tool.



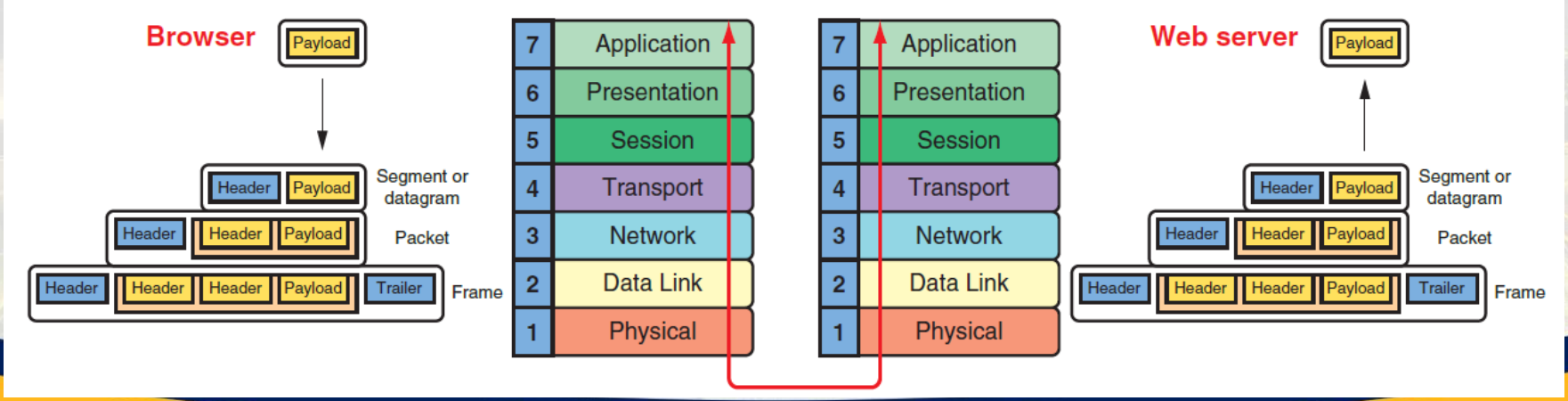
Recall: The OSI Model

- *Layers 7, 6, and 5*—Data and instructions, known as the payload, are generate
- *Layer 4*—A Transport layer protocol, usually either TCP or UDP, adds a header to the payload.
- *Layer 3*—The Network layer adds its own header to the passed-down segment or datagram.

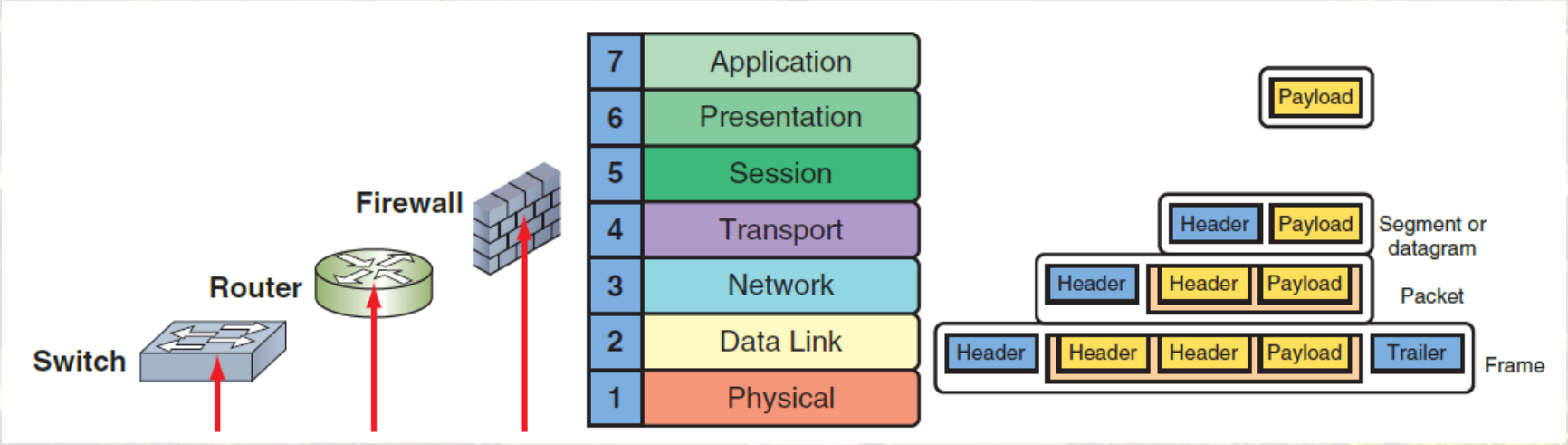


Recall: The OSI Model

- *Layer 2*—The packet is passed to the Data Link layer on the NIC, which encapsulates this data with its own header and trailer, creating a frame.
- *Layer 1*—The Physical layer on the NIC receives the frame and places the actual transmission on the network.



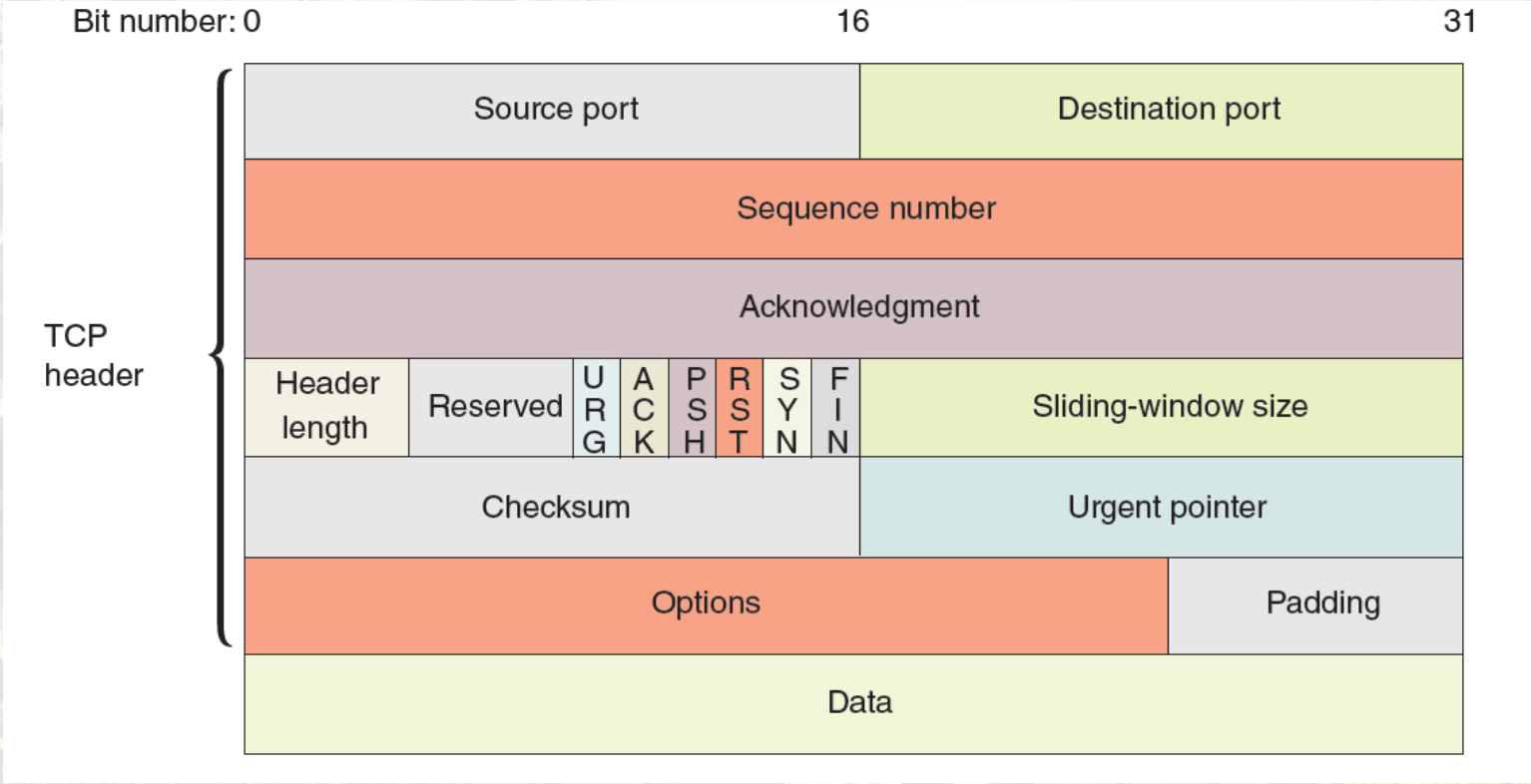
Recall: The OSI Model



TCP (Transmission Control Protocol)

- ***connection-oriented***—Before TCP transmits data, it ensures that a connection or session is established, similar to making sure someone is listening on the other end of a phone call before you start talking.
- ***sequencing and checksums***—In the analogy of a phone call, you might ask the other person if he can hear you clearly, and repeat a sentence as necessary.
- ***flow control***—You might slow down your talking over the phone if the other person needs a slower pace in order to hear every word and understand your message.

Fields in a TCP Segment



Fields in a TCP Segment

Table 4-1 Fields in a TCP segment

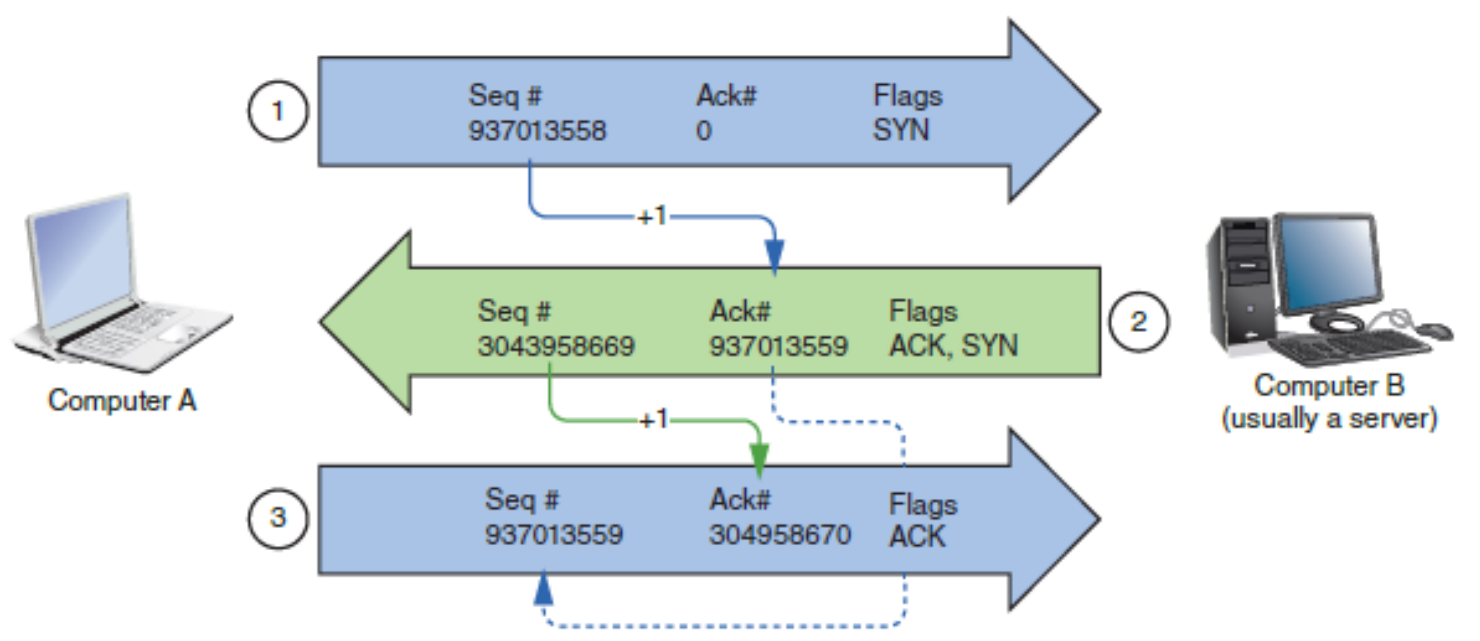
Field		Length	Function
Header	Source port	16 bits	Indicates the port at the source node. Recall that a port is the number that identifies a process on a host. The port allows a process to be available for incoming or outgoing data.
	Destination port	16 bits	Indicates the port at the destination node.
	Sequence number	32 bits	Identifies the data segment's position in the stream of data segments being sent.
	Acknowledgment number	32 bits	Confirms receipt of data via a return message to the sender.
	TCP header length	4 bits	Indicates the length of the TCP header in bytes. The header can be a minimum of 20 bytes to a maximum of 60 bytes in 4-byte increments. It's also called the Data offset field because it indicates the offset from the beginning of the segment until the start of the data carried by the segment.
	Reserved	6 bits	Indicates a field reserved for later use.
	Flags	6 bits	Identifies a collection of six 1-bit fields or flags that signal special conditions about other fields in the header. The following flags are available to the sender: • *URG*—If set to *1*, the Urgent pointer field later in the segment contains information for the receiver. If set to *0*, the receiver will ignore the Urgent pointer field. • *ACK*—If set to *1*, the Acknowledgment field earlier in the segment contains information for the receiver. If set to *0*, the receiver will ignore the Acknowledgment field. • *PSH*—If set to *1*, data should be sent to an application without buffering. • *RST*—If set to *1*, the sender is requesting that the connection be reset.

Fields in a TCP Segment

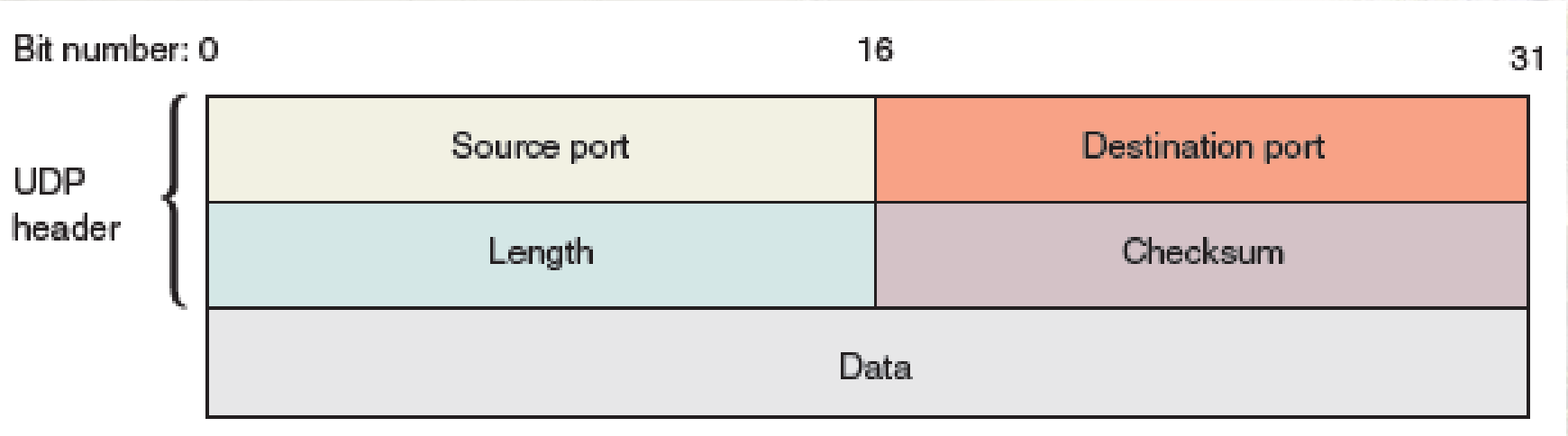
Table 4-1 Fields in a TCP segment *(continued)*

Field		Length	Function
			• *SYN*—If set to *1*, the sender is requesting a synchronization of the sequence numbers between the two nodes. This code indicates that no payload is included in the segment, and the acknowledgment number should be increased by 1 in response. • *FIN*—If set to *1*, the segment is the last in a sequence and the connection should be closed.
	Sliding-window size (or window)	16 bits	Indicates how many bytes the sender can issue to a receiver before acknowledgment is received. This field performs flow control, preventing the receiver's buffer from being deluged with bytes.
	Checksum	16 bits	Allows the receiving node to determine whether the TCP segment became corrupted during transmission.
	Urgent pointer	16 bits	Indicates a location in the data field where urgent data resides.
	Options	0–32 bits	Specifies special options, such as the maximum segment size a network can handle.
	Padding	Variable	Contains filler bits to ensure that the size of the TCP header is a multiple of 32 bits.
Data		Variable	Contains data sent by the source host. The data field is not part of the TCP header—it is encapsulated by the TCP header. The size of the data field depends on how much data needs to be transmitted, the constraints on the TCP segment size imposed by the network type, and the limitation that the segment must fit within an IP packet at the next layer.

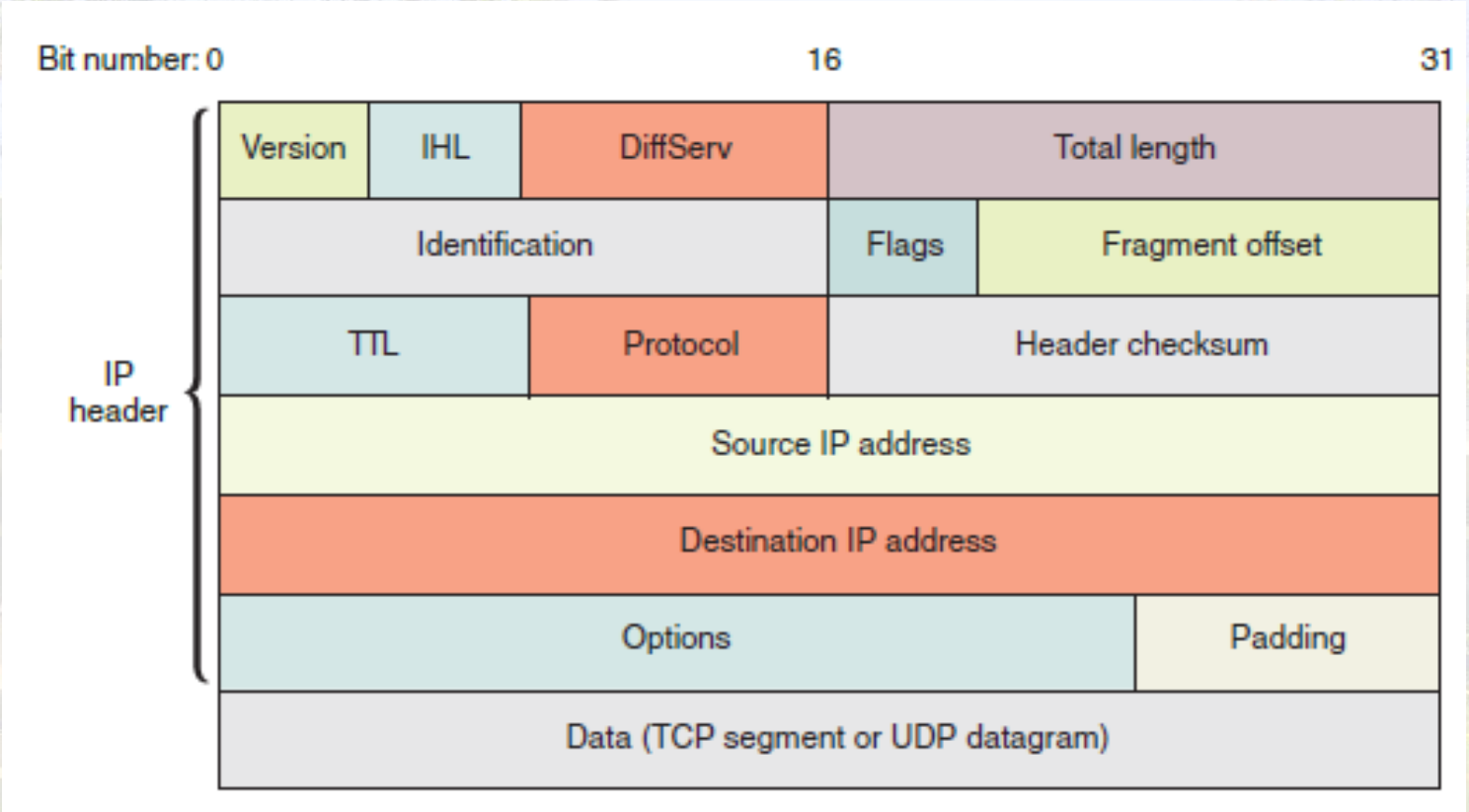
TCP Three-Way Handshake



UDP (User Datagram Protocol)



IP (Internet Protocol) – IPv4



7	APPLICATION
6	PRESENTATION
5	SESSION
4	TRANSPORT
3	NETWORK
2	DATA LINK
1	PHYSICAL

Table 4-3 Fields in an IPv4 packet

Field		Length	Function
Header	Version	4 bits	Identifies the version number of the protocol—for example, IPv4 or IPv6. The receiving workstation looks at this field first to determine whether it can read the incoming data. If it cannot, it will reject the packet.
	IHL (Internet header length)	4 bits	Indicates the length of the IP header in bytes. The header can be a minimum of 20 bytes to a maximum of 60 bytes in 4-byte increments. It's also called the Data offset field because it indicates the offset from the beginning of the packet until the start of the data carried by the packet.
	DiffServ (Differentiated services)	8 bits	Informs routers the level of precedence they should apply when processing the incoming packet. Differentiated services allows up to 64 values and a wide range of priority-handling options.
	Total length	16 bits	Identifies the total length of the IP packet, including the header and data, in bytes. An IP packet, including its header and data, cannot exceed 65,535 bytes.
	Identification	16 bits	Identifies the message to which a packet belongs and enables the receiving host to reassemble fragmented messages. This field and the following two fields, Flags and Fragment offset, assist in reassembly of fragmented packets. IP packets that are larger than what the network allows are fragmented into smaller packets for transmission.
	Flags	3 bits	Indicates whether a message is fragmented and, if it is fragmented, whether this packet is the last fragment. The first bit is reserved for future use. When the second bit is set, it prevents the packet from being fragmented. A value of 1 in the third bit indicates more fragments are on the way.
	Fragment offset	13 bits	Identifies where the packet fragment belongs in the series of incoming fragments.
	TTL (Time to Live)	8 bits	Indicates the maximum duration that the packet can remain on the network before it is discarded. Although this field was originally meant to represent units of time, on modern networks it represents the number of times a packet can still be forwarded by a router, or the maximum number of router **hops** it has remaining. The TTL for packets varies and can be configured; it is usually set at 32 or 64. Each time a packet passes through a router, its TTL is reduced by 1. When a router receives a packet with a TTL equal to 0, it discards that packet and sends a *TTL expired* message via ICMP back to the source host.
	Protocol	8 bits	Identifies the type of protocol that will receive the packet (for example, TCP, UDP, or ICMP).

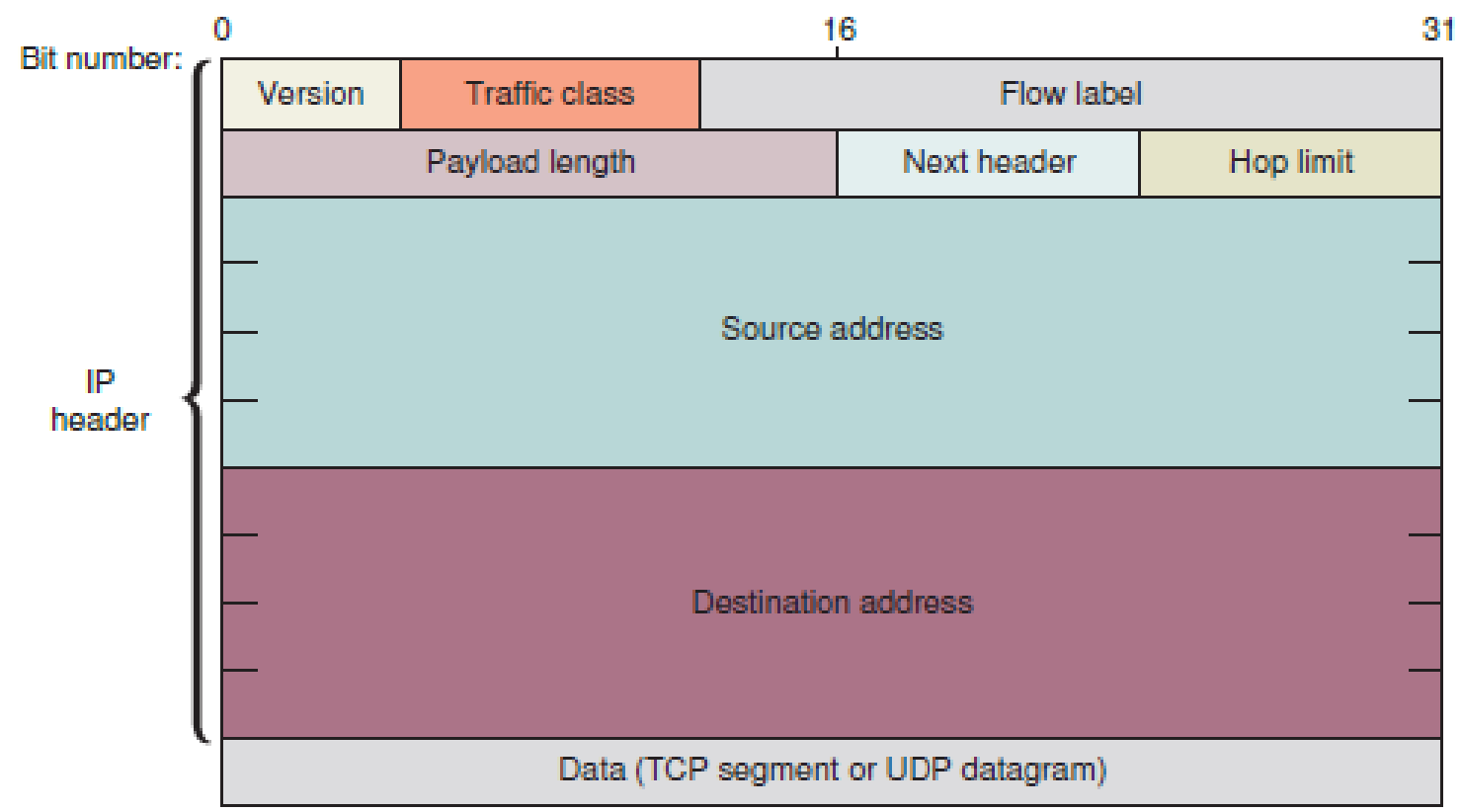
IP (Internet Protocol) – IPv4

IP (Internet Protocol) – IPv4

Table 4-3 Fields in an IPv4 packet *(continued)*

Field		Length	Function
	Header checksum	16 bits	Allows the receiving host to calculate whether the IP header has been corrupted during transmission. If the checksum accompanying the message does not match the calculated checksum when the packet is received, the packet is presumed to be corrupt and is discarded.
	Source IP address	32 bits	Indicates the IP address of the source host.
	Destination IP address	32 bits	Indicates the IP address of the destination host.
	Options	Variable	May contain optional routing and timing information.
	Padding	Variable	Contains filler bits to ensure that the header is a multiple of 32 bits.
Data		Variable	Includes the data originally sent by the source host, plus any headers from higher layers. The data field is not part of the IP header—it is encapsulated by the IP header.

IP (Internet Protocol) – IPv6

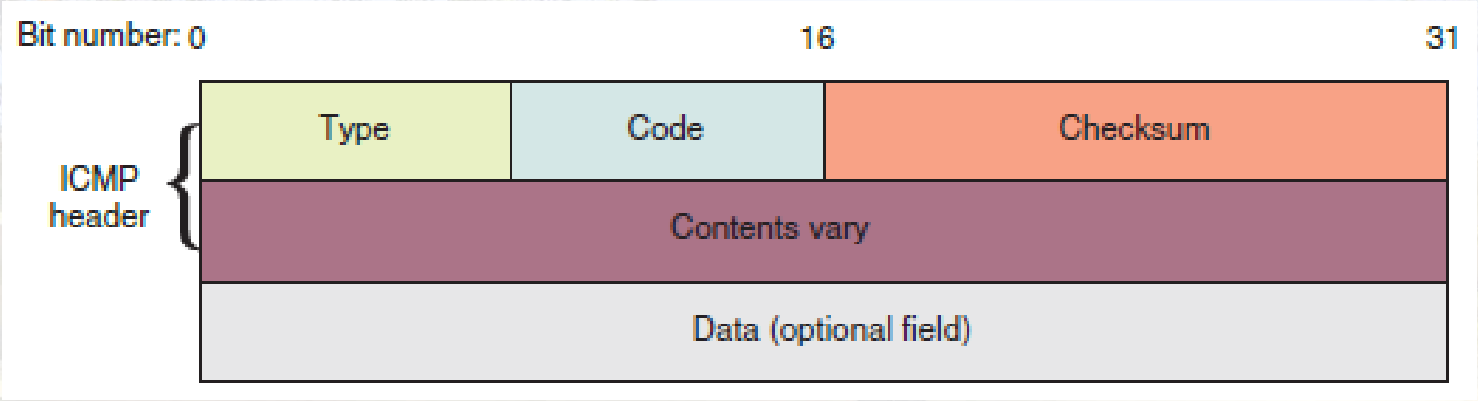


IP (Internet Protocol) – IPv6

Table 4-5 Fields in an IPv6 packet

Field		Length	Function
Header	Version	4 bits	Indicates which IP version the packet uses.
	Traffic class	8 bits	Identifies the packet's priority. It is similar to, but not the same as, the DiffServ field in IPv4 packets.
	Flow label	20 bits	Indicates which flow, or sequence of packets from one source to one or multiple destinations, the packet belongs to. Routers interpret flow information to ensure that packets belonging to the same transmission arrive together. Flow information may also help with traffic prioritization.
	Payload length	16 bits	Indicates the size of the payload, or data, carried by the packet. Unlike the Total length field in IPv4 packets, the Payload length in IPv6 packets does not refer to the size of the whole packet.
	Next header	8 bits	Identifies the type of header that immediately follows the IP packet header, usually TCP or UDP.
	Hop limit	8 bits	Indicates the number of times the packet can be forwarded by routers on the network, similar to the TTL field in IPv4 packets. When the hop limit reaches 0, the packet is discarded.
	Source address	128 bits	Indicates the full IP address of the source host.
	Destination address	128 bits	Indicates the full IP address of the destination host.
Data		Variable	Includes the data originally sent by the source host, plus any headers from higher layers. The data field is not part of the IPv6 header—it is encapsulated by the IPv6 header.

ICMP (Internet Control Message Protocol)



7	APPLICATION
6	PRESENTATION
5	SESSION
4	TRANSPORT
3	NETWORK
2	DATA LINK
1	PHYSICAL

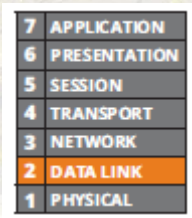
ICMP (Internet Control Message Protocol)

Table 4-7 An ICMP packet

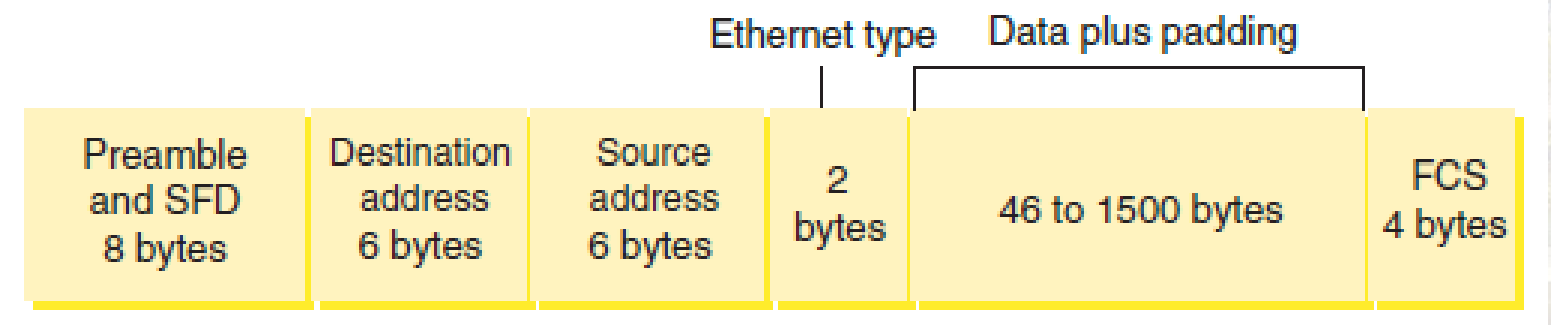
Field		Length	Function
Header	Type	8 bits	Indicates the type of ICMP message, such as Destination Unreachable.
	Code	8 bits	Indicates the subtype of the message, such as Destination host unknown.
	Checksum	16 bits	Allows the receiving node to determine whether the ICMP packet became corrupted during transmission.
	Rest of header	32 bits	Varies depending on message type and subtype.
Data		Variable	Usually contains the IP header and first 8 bytes of the data portion of the IP packet that triggered the ICMP message.

ARP (Address Resolution Protocol) on IPv4 Networks

IP Address	Hardware Address	Type
123.45.67.80	60:23:A6:F1:C4:D2	Static
123.45.67.89	20:00:3D:21:E0:11	Dynamic
123.45.67.73	A0:BB:77:C2:25:FA	Dynamic



Ethernet



7	APPLICATION
6	PRESENTATION
5	SESSION
4	TRANSPORT
3	NETWORK
2	DATA LINK
1	PHYSICAL

Ethernet

Table 4-8 Fields of an Ethernet II frame

Field name		Length	Description
Preamble		7 bytes	Synchronizes the recipient's receiver clock.*
SFD (start frame delimiter)		1 byte	Indicates the frame is about to begin.*
Header	Destination address	6 bytes	Provides the recipient's MAC address.
	Source address	6 bytes	Provides the sender's MAC address.
	Type field	2 bytes	Specifies the upper-layer protocol carried in the frame. For example, an IP packet has 0x0800 in the Type field.
Data		46 bytes to 1500 bytes	If the data is not at least 46 bytes, padding is added to meet the minimum.
Trailer	FCS (frame check sequence)	4 bytes	Ensures that the data at the destination exactly matches the data issued from the source using the CRC (cyclic redundancy check) algorithm.

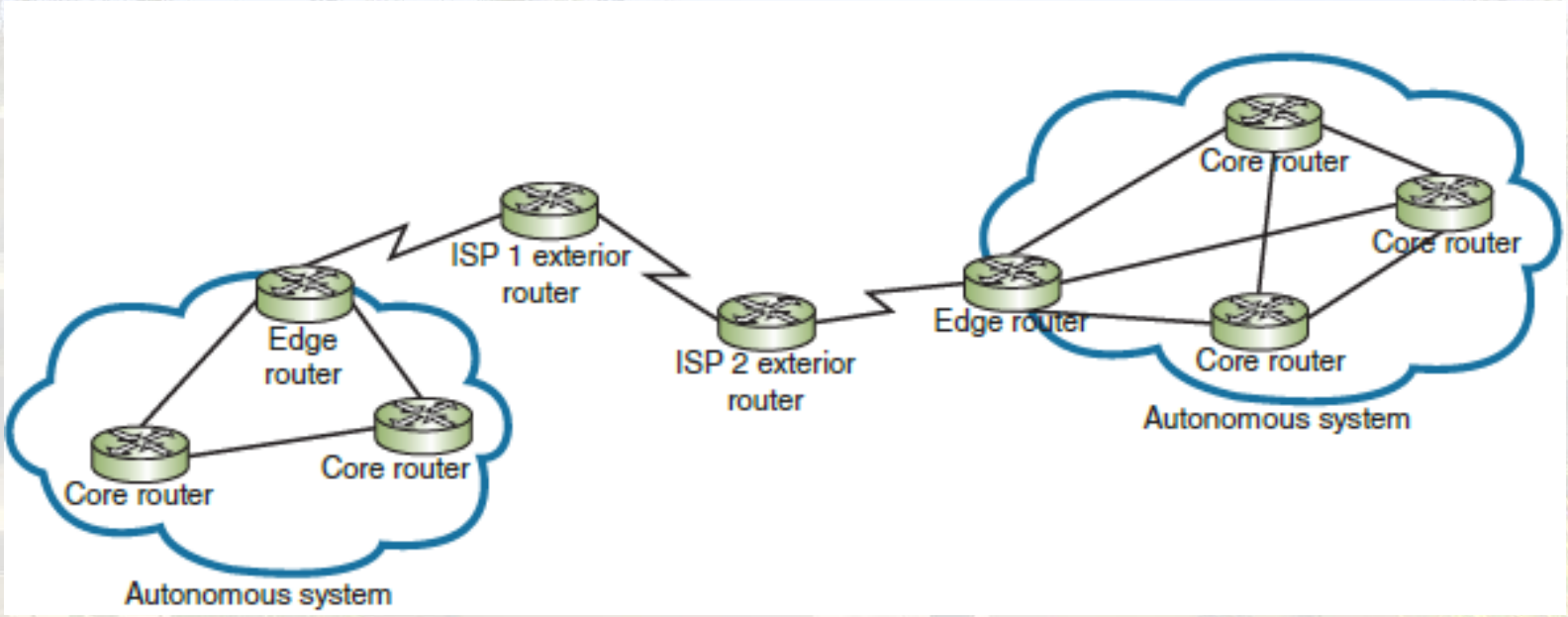
Routers and How They Work

- Explain the concepts and characteristics of routing and switching.
- Given a scenario, determine the appropriate placement of networking devices on a network and install/configure them.
- Explain the purposes and use cases for advanced networking devices.
- Given a scenario, use the appropriate tool.

Routers and How They Work

7	APPLICATION
6	PRESENTATION
5	SESSION
4	TRANSPORT
3	NETWORK
2	DATA LINK
1	PHYSICAL

Routers and How They Work



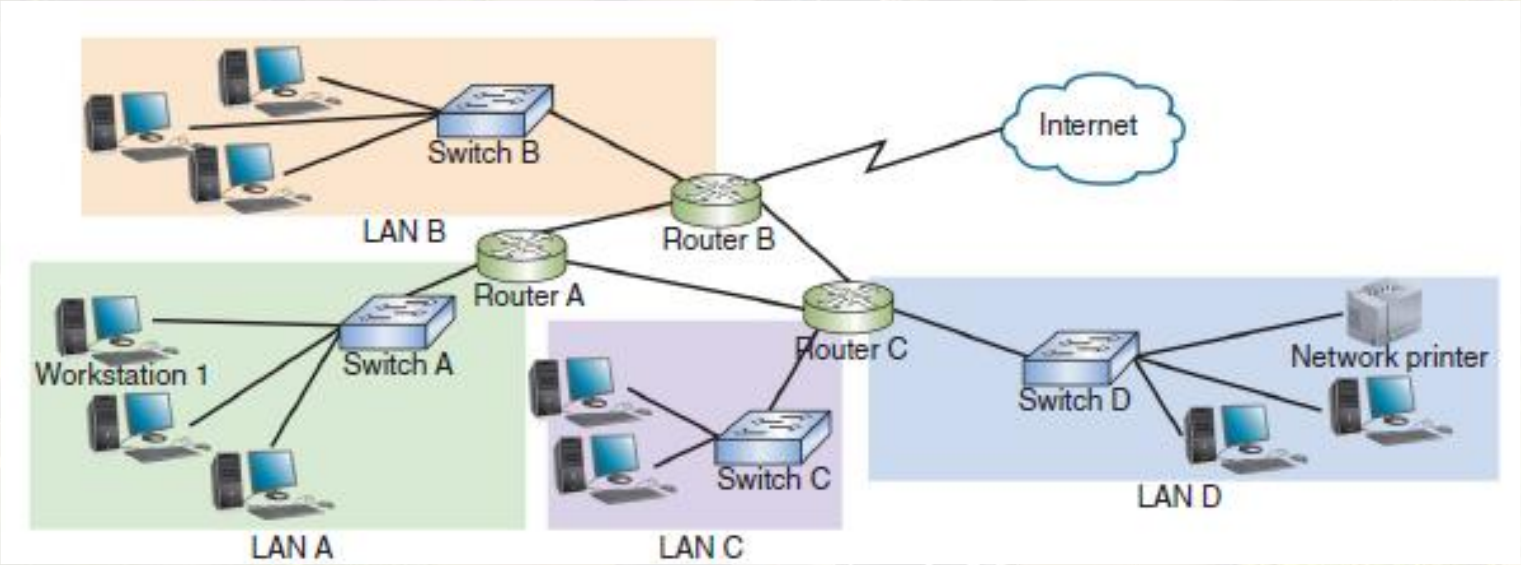
Multilayer Switches

- A Layer 3 switch is a switch that is capable of interpreting Layer 3 data and works much like a router
- Layer 4 switches also exist and are capable of interpreting Layer 4 data.
- The features of Layer 3 and Layer 4 switches vary widely depending on the manufacturer and price and can cost significantly more than Layer 2 switches.

Routing Tables

- A routing table is a database that holds information about where hosts are located and the most efficient way to reach them.

7	APPLICATION
6	PRESENTATION
5	SESSION
4	TRANSPORT
3	NETWORK
2	DATA LINK
1	PHYSICAL



Routing Path Types

- static routing—A network administrator configures a routing table to direct messages along specific paths between networks.
- dynamic routing—A router automatically calculates the best path between two networks and accumulates this information in its routing table.

The route Command

- The route command allows you to view a host's routing table. Here are some variations for different operating systems:
  - Linux or UNIX—Enter route at the shell prompt.
  - Windows—Enter route print at the command prompt.
  - Cisco's IOS—Enter show ip route at the CLI in enable mode.

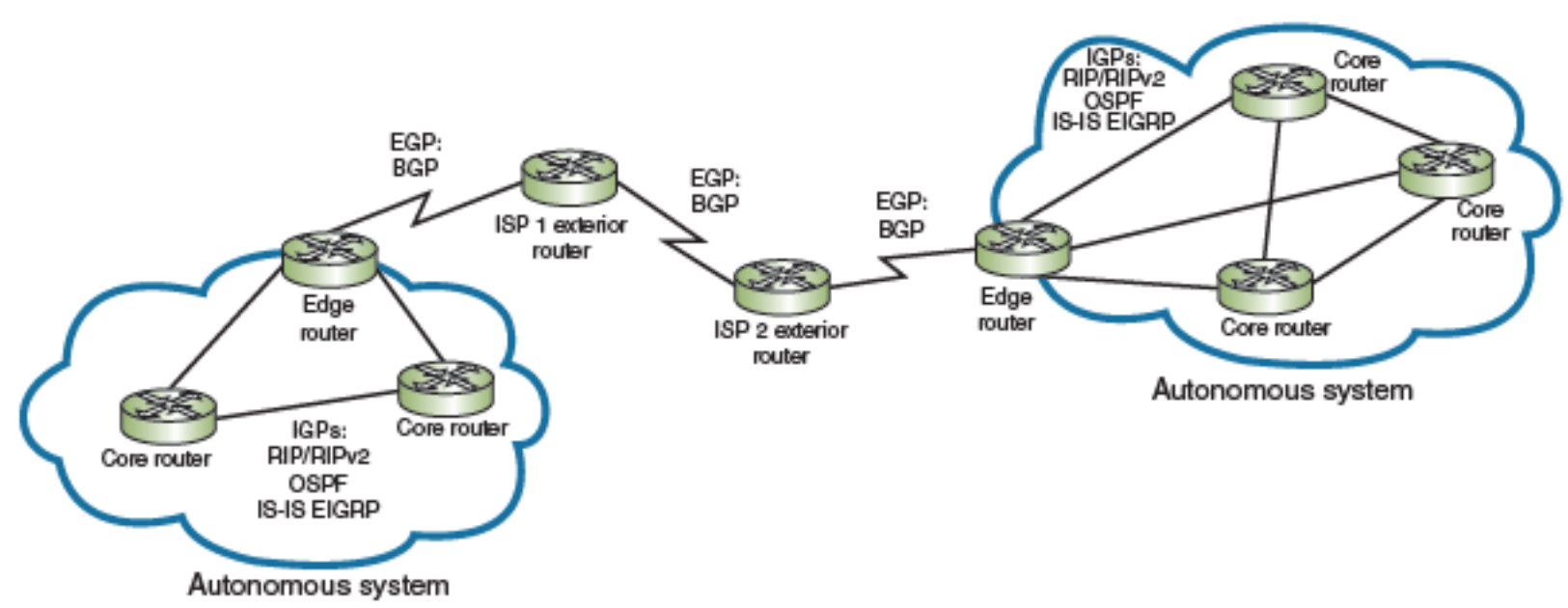
Routing Metrics

- Hop count, which is the number of network segments crossed
- Theoretical bandwidth and actual throughput on a potential path
- Delay, or latency, on a potential path, which results in slower performance
- Load, which is the traffic or processing burden sustained by a router in the path
- MTU, which is the largest IP packet size in bytes allowed by routers in the path without fragmentation (excludes the frame size on the local network)

Routing Metrics

- Routing cost, which is a value assigned to a particular route as judged by the network administrator; the more desirable the path, the lower its cost
- Reliability of a potential path, based on historical performance
- A network's topology

Interior and Exterior Gateway Protocols



Troubleshooting Route Issues

- Explain the concepts and characteristics of routing and switching.
- Explain authentication and access controls.
- Summarize common networking attacks.
- Given a scenario, use the appropriate tool.
- Given a scenario, troubleshoot common network service issues.

Troubleshooting Tools

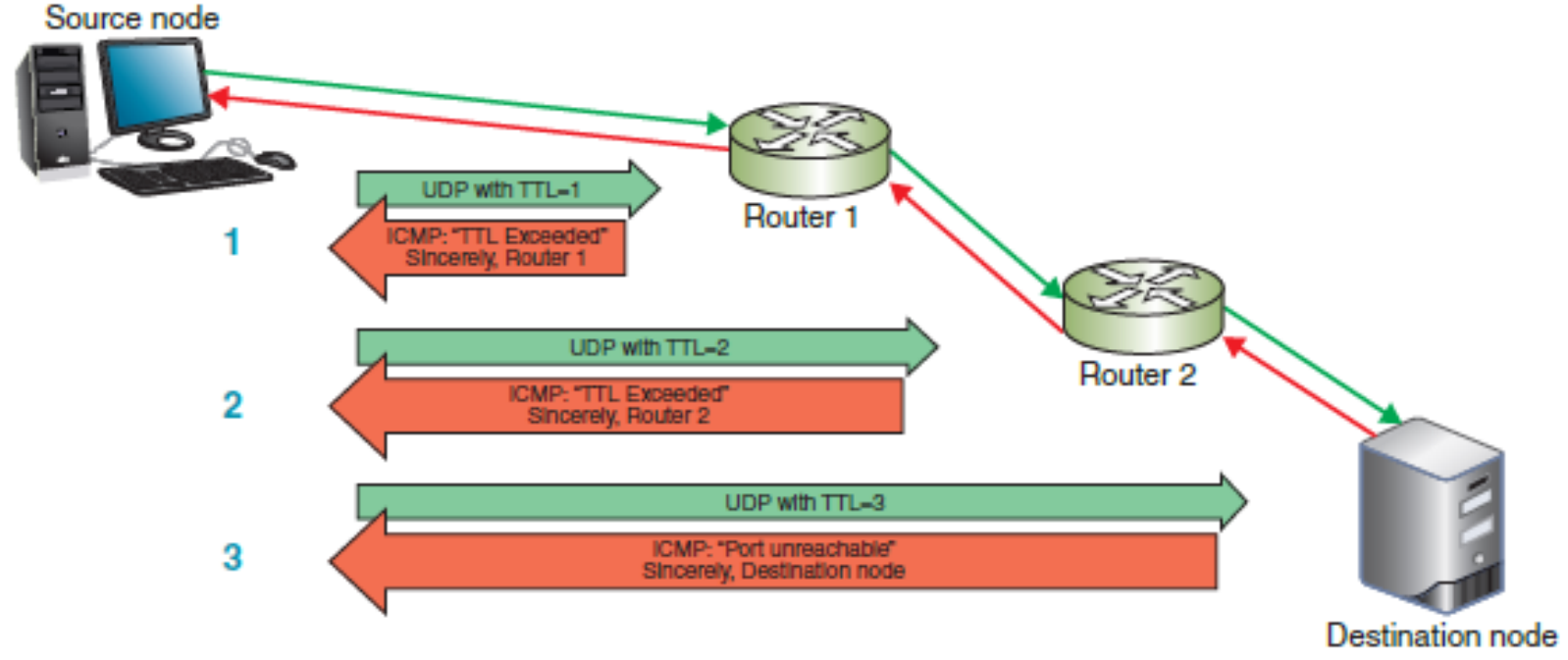
- netstat

Table 4-11 `netstat` command options

`netstat` command	Description
`netstat`	Lists all active TCP/IP connections on the local machine, including the Transport layer protocol used (usually just TCP), messages sent and received, IP address, and state of those connections.
`netstat -n`	Lists current connections, including IP addresses and ports.
`netstat -f`	Lists current connections, including IP addresses, ports, and FQDNs.
`netstat -a`	Lists all current TCP connections and all listening TCP and UDP ports.
`netstat -e`	Displays statistics about messages sent over a network interface, including errors and discards.
`netstat -s`	Displays statistics about each message transmitted by a host, separated according to protocol type (TCP, UDP, IP, or ICMP).
`netstat -r`	Displays routing table information.
`netstat -o`	Lists the PID (process identifier) for each process using a connection and information about the connection.
`netstat -b`	Lists the name of each process using a connection and information about the connection. Requires an elevated Command Prompt.

Troubleshooting Tools

- tracert or traceroute



Troubleshooting Tools

- tracert or traceroute

Table 4-12 `traceroute` and `tracert` command options

Command	Description

`traceroute -n google.com`
or
`tracert -d google.com`
	Instructs the command to not resolve IP addresses to host names.

`traceroute -m 12 google.com`
or
`tracert -h 12 google.com`
	Specifies the maximum number of hops when attempting to reach a host; this parameter must be followed by a specific number. Without this parameter, the command defaults to 30.

`traceroute -w 2 google.com`
or
`tracert -w 2000 google.com`
	Identifies a timeout period for responses; this parameter must be followed by a variable to indicate the number of seconds (in Linux) or milliseconds (in Windows) that the utility should wait for a response. The default time is usually between 3 and 5 seconds for Linux and 4000 milliseconds (4 seconds) for Windows.

`traceroute -f 3 google.com`	Sets the first TTL value and must be followed by a variable to indicate the number of hops for the first probe. The default value is 1, which begins the trace at the first router on the route. Beginning at later hops in the route can more quickly narrow down the location of a network problem. `tracert` does not have a corresponding parameter for this function.

Troubleshooting Tools

- pathping

Table 4-13 `pathping` command options

`pathping` command	Description
`pathping -n google.com`	Instructs the command to not resolve IP addresses to host names.
`pathping -h 12 google.com`	Specifies the maximum number of hops the messages should take when attempting to reach a host (the default is 30); this parameter must be followed by a specific number of hops.
`pathping -p 2000 google.com`	Identifies the wait time between pings; this parameter must be followed by a variable to indicate the number of milliseconds to wait. The default time is 4000 milliseconds (4 seconds).
`pathping -q 4 google.com`	Limits the number of queries per hop; must be followed by a variable to indicate the number of queries allowed. By default, `pathping` sends 100 pings per hop, which tends to take a long time to run.

Troubleshooting Tools

- tcpdump

Table 4-14 `tcpdump` command options

`tcpdump` command	Description
`tcpdump not port 22` or `tcpdump not port 23`	Filters out SSH or Telnet packets, which is helpful when running `tcpdump` on a remotely accessed network device.
`tcpdump -n`	Instructs the command to not resolve IP addresses to host names.
`tcpdump -c 50`	Limits the number of captured packets to 50.
`tcpdump -i any`	Listens to all network interfaces on a device.
`tcpdump -D`	Lists all interfaces available for capture.
`tcpdump port http`	Filters out all traffic except HTTP.
`tcpdump -w capture.cap`	Saves the file output to a file named capture.cap.
`tcpdump -r capture.cap`	Reads the file capture.cap and outputs the data in the terminal window. This file can also be read by applications like Wireshark.

Solving Common Routing Problems

Table 4-15 Command-line utilities

Command	Common uses
`arp`	Provides a way of obtaining information from and manipulating a device's ARP table.
`dig`	Queries DNS servers with more advanced options than `nslookup`.
`ipconfig` or `ifconfig`	Provides information about TCP/IP network connections and the ability to manage some of those settings.
`netstat`	Displays TCP/IP statistics and details about TCP/IP components and connections on a host.
`nmap`	Detects, identifies, and monitors devices on a network.
`nslookup`	Queries DNS servers and provides the ability to manage the settings for accessing those servers.
`pathping` (`mtr` on Linux/UNIX/ macOS)	Sends multiple pings to each hop along a route, then compiles the information into a single report.
`ping`	Verifies connectivity between two nodes on a network.
`route`	Displays a host's routing table.
`tcpdump`	Captures traffic that crosses a computer's network interface.
`traceroute` or `tracert`	Traces the path from one networked node to another, identifying all intermediate routers between the two nodes.

Duplicate MAC Addresses

Most of the time, though, duplicate MAC addresses only cause intermittent connectivity issues for the computers involved in the duplication. Here's how the situation develops:

- Step 1—Each computer regularly broadcasts its IP address and the duplicated MAC address so devices on the network can update their ARP tables.
- Step 2—Those other devices, in response, update their records to point toward one computer, and then the other computer, and then back to the first one, and so on, depending upon the latest transmission they received.
- Step 3—Sometimes devices will send communications to the correct computer, and sometimes their records will be wrong.

Hardware Failure

1. Use tracert or traceroute (depending on your OS) to track down malfunctioning routers and other devices on larger networks. Because ICMP messages are considered low priority, be sure to run the command multiple times and compare the results before drawing any conclusions.
2. Keep in mind that routers are designed to route traffic to other destinations. You might get more accurate tracert or traceroute feedback on a questionable router if you target a node on the other side of that router rather than aiming for the router itself.
3. As you hone in on the troublesome device, use ping to test for network connectivity.

Discovering Neighbor Devices

- Routers learn about all the devices on their networks through a process called
- neighbor discovery. This process can go awry when changes are made to the network, or when a problem is developing but is only producing sporadic symptoms.
- On IPv4 networks, neighbor discovery is managed by ARP with help from ICMP. The arp command can be used on IPv4 devices to diagnose and repair problems with ARP tables. If you notice inconsistent connectivity issues related to certain addresses, you might need to flush the ARP table on any device experiencing the problem. This forces the device to repopulate its ARP table in order to correct any errors.
- IPv6 devices use NDP (Neighbor Discovery Protocol) in ICMPv6 messages to automatically detect neighboring devices, and to automatically adjust when neighboring nodes fail or are removed from the network. NDP eliminates the need for ARP and some ICMP functions in IPv6 networks, and is much more resistant to hacking attempts than ARP.