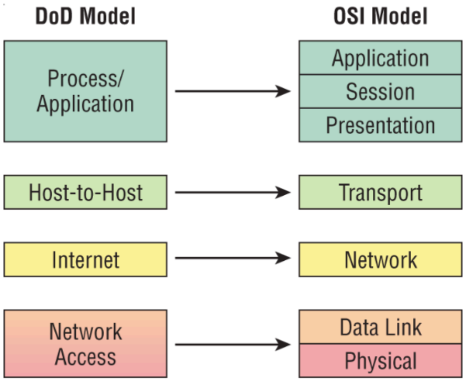
The OSI and DARPA TCP/IP Networking Models

DEC/IBM (SNA) before 1980s, into 90s when TCP/IP prevailed and overtook the OSI model
In old TCP/IP, "Link" comprised Physical and Data-Link layers, Network was called "Internet"
"Link" was also called "Network Interface" or "Network Access" layer

OSI Networking Model (DARPA's TCP/IP dominates, but OSI won the terminology war)

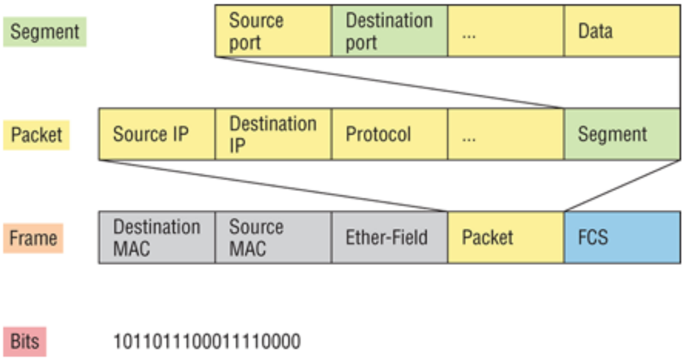


The DoD's Application layer is split into 3 layers in the OSI. Transport is often still called Host-to-Host in the DoD model. Network is often still called the Internet layer in the DoD model. Finally, the Data Link and Physical layers are still often unified in the DoD model, and called the Network Access layer, but also known as the Network Interface, or Link layer.

The OSI has remained- as the network stack is always referred to by it's 7 layers. Additionally, "layer 2" refers to data link, and "layer 3" for transport. The DoD model has since made these changes, yet retains layers 5, 6, and 7 as one Application layer.

***Same-Layer Interactions** are between two computers at the same layer (like HTTP to HTTP)*
***Adjacent-Layer** - A layer passes info to neighboring layer (up or down) on the same computer*

An example of outgoing data, HTTP sends to TCP to put into a segment, it then sends to IP to pack up in a packet, and data-link frames it to go out the wire as bits. Likewise, when frame is pulled out of the incoming bits, it's frame is unpacked, the IP packet passed up the stack to be opened it up, revealing a segment to give to the adjacent transport layer.



A layer's data is referred as: a Protocol Data Unit. A layer 2 frame is a L2PDU, a network IP packet is a L3PDU, and a UDP or TCP segment is a L4PDU.

Benefits of layered protocol specifications: Less complex, easier to learn and develop with modular engineering and standard interfaces. The TCP/IP and OSI stack are *reference models* of a *layered architecture.*

Relation to the TCP/IP model

In the Internet Protocol Suite (TCP/IP), OSI's data link layer functionality is contained within its lowest layer, the link layer. The TCP/IP link layer has the operating scope of the link a host is connected to, and only concerns itself with hardware issues to the point of obtaining hardware (MAC) addresses for locating hosts on the link and transmitting data frames onto the link. The link layer functionality was described in RFC 1122 and is defined differently than the Data Link Layer of OSI, and encompasses all methods that affect the local link.

The Application Layer (Layer 7) refers to communications services to applications and is the interface between the network and the application. Examples include. Telnet, HTTP, FTP, Internet browsers, NFS, SMTP gateways, SNMP, X.400 mail, and FTAM.

The Presentation Layer (Layer 6) defining data formats, such as ASCII text, EBCDIC text, binary, BCD, and JPEG. Encryption also is defined as a presentation layer service. Examples include. JPEG, ASCII, EBCDIC, TIFF, GIF, PICT, encryption, MPEG, and MIDI.

The Session Layer (Layer 5) defines how to start, control, and end communication sessions. This includes the control and management of multiple bidirectional messages so that the application can be notified if only some of a series of messages are completed. This allows the presentation layer to have a seamless view of an incoming stream of data. The presentation layer can be presented with data if all flows occur in some cases. Examples include. RPC, SQL, NFS, NetBios names, AppleTalk ASP, and DECnet SCP
The Transport Layer (Layer 4) defines several functions, including the choice of protocols. The most important Layer 4 functions are error recovery and flow control. The transport layer may provide for retransmission, i.e., error recovery, and may use flow control to prevent unnecessary congestion by attempting to send data at a rate that the network can accommodate, or it might not, depending on the choice of protocols. Multiplexing of incoming data for different flows to applications on the same host is also performed. Reordering of the incoming data stream when packets arrive out of order is included. Examples include. TCP, UDP, and SPX.

The Network Layer (Layer 3) defines end-to-end delivery of packets and defines logical addressing to accomplish this. It also defines how routing works and how routes are learned; and how to fragment a packet into smaller packets to accommodate media with smaller maximum transmission unit sizes. Examples include. IP, IPX, AppleTalk DDP, and ICMP. Both IP and IPX define logical addressing, routing, the learning of routing information, and end-to-end delivery rules. The IP and IPX protocols most closely match the OSI network layer (Layer 3) and are called Layer 3 protocols because their functions most closely match OSI's Layer 3.

The Data Link Layer (Layer 2) is concerned with getting data across one particular link or medium. The data link protocols define delivery across an individual link. These protocols are necessarily concerned with the type of media in use. Examples includE. IEEE 802.3/802.2, HDLC, Frame Relay, PPP, FDDI, ATM, and IEEE 802.5/802.2.

The TCP/IP model is not a top-down comprehensive design reference for networks. It was formulated for the purpose of illustrating the logical groups and scopes of functions needed in the design of the suite of internetworking protocols of TCP/IP, as needed for the operation of the Internet. In general, direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the layering in TCP/IP is not a principal design criterion and in general considered to be "harmful" (RFC 3439). In particular, TCP/IP does not dictate a strict hierarchical sequence of encapsulation requirements, as is attributed to OSI protocols.

Nonetheless, they are similar enough to provide a good representation that makes it easier to understand the general interaction of the overlying technologies.

Layer 7 - Application Layer - Telnet, HTTP, FTP, SMTP, POP3, VoIP, SNMP. Provides an interface between the communications software and any applications that need to communicate outside the computer on which the application resides. It also defines processes for user authentication

Layer 6 - Presentation Layer - ASCII, EBCDIC, BCD; de/compression, de/encryption. "Define and negotiate data formats"; "how standard data should be formatted." Some absurdly infer (alarmingly even Cisco) that image formats like JPG and PNG are associated with the presentation layer.

Layer 5 - Session Layer - PPTP, L2TP, NetBIOS, PAP, RPC, SOCKS. From a client to a server, is coordinated and organized via three different modes: simplex, half-duplex, and full-duplex. How to start, control, and end conversations (called sessions). This includes the control and management of multiple bidirectional messages so that the application can be notified if only some of a series of messages are completed. This allows the presentation layer to have a seamless view of an incoming stream of data.

Application Layer - TCP/IP Merges layers 5-7 into one layer.
TCP/IP Transport Layer - aka Host-to-Host - UDP/TCP/ICMP
Network Layer - Internet layer in TCP/IP stack- IP v4 and v6
Data-Link aka Network Access Layer - In TCP/IP includes the Physical layer - PPP, L2TP, HDLC, Ethernet (IEEE 802.3 MAC)

Layer 1 - Physical Layer - These standards deal with the physical characteristics of the transmission medium, including connectors, pins, use of pins, electrical currents, encoding, light modulation, and the rules for how to activate and deactivate the use of the physical medium.

HTTP Overview and Protocol Mechanisms
Send HTTP GET file.html header request, 200 (ok header return code), 404 (not found)
File is returned [http header][data], followed by [data] - no header, just multiple [data] parts

TCP Error Recovery [TCP SEQ=1][HTTP 200][data], [TCP SEQ=2][data], [TCP SEQ=3]
Receiver notices one lost, asks for resend [TCP resend=2 next please]

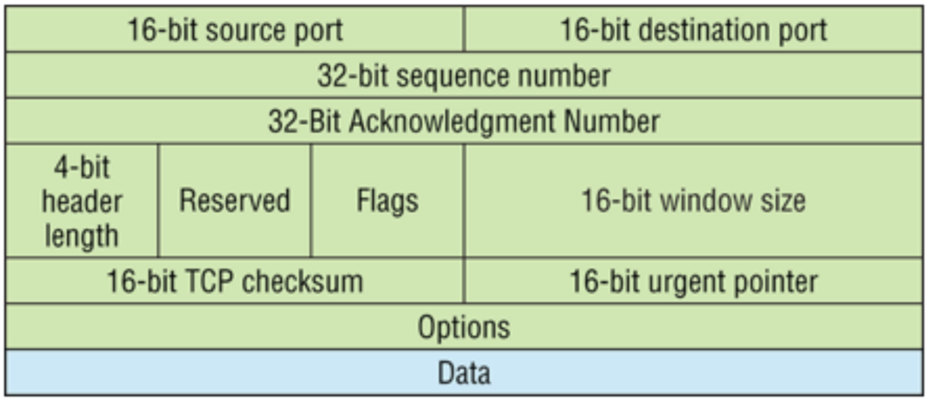
<u>Layer 4 - Transport Layer (aka Host-to-Host)</u> - end-to-end data transport services, logical connection between hosts. TCP implements *reliable networking* - requires that acknowledgments, sequencing, and flow control will all be used. The 3-way handshake virtual circuit setup is often referred to as overhead; removes a lot of programming work, but for real-time video and VoIP, UDP is often better because using it results in less overhead)
Connection establishment and termination: ACK-->SYN-ACK-->SYN and FIN-->FINACK?-->FIN!-->OK

The types of flow control are buffering, windowing, and congestion avoidance.
Sliding Window - TCP allows the receiving device to dictate the amount of data the sender can send before receiving an acknowledgment, the mechanism to grant future windows is typically just a number that grows (slides) upward slowly after each acknowledgment.

"Positive acknowledgment with retransmission" - The sender documents each segment measured in bytes, then sends and waits for acknowledgment before sending the next segment. The transmitting machine starts a timer and will retransmit if it expires before it gets an acknowledgment back from the receiving end.

Multiplexing with port numbers using sockets- Separation of the data from different applications (browser windows and apps); combined with ordered data transfer and data segmentation. *The source port number is arbitrary, destination port is specific to process/application.* The virtual circuit is defined by the source and destination port number plus the source and destination IP address and called a socket. *Understand that the host just makes this up, starting at port number 1024 because 0 through 1023 are reserved for well-known port numbers*. The destination port number defines the upper-layer process or application that the data stream is handed to when the data stream is reliably rebuilt on the receiving host." [example showed a source port for virtual circuit on PC1 is 1028, destination port on PC2 is 23]

The TCP Segment's Format and Header Fields



<table>
<tr><td colspan="3">16-bit source port</td><td>16-bit destination port</td></tr>
<tr><td colspan="4">32-bit sequence number</td></tr>
<tr><td colspan="4">32-Bit Acknowledgment Number</td></tr>
<tr><td>4-bit header length</td><td>Reserved</td><td>Flags</td><td>16-bit window size</td></tr>
<tr><td colspan="3">16-bit TCP checksum</td><td>16-bit urgent pointer</td></tr>
<tr><td colspan="4">Options</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>

The header is 20 bytes long, or up to 24 bytes with options.

<u>Source port:</u> port number of the application on the host sending the data
<u>Destination port:</u> This is the port number of the application requested on the destination host.
<u>Sequence number:</u> Used to put the data back in the correct order (or retransmit missing or damaged data) during sequencing process.
<u>Acknowledgment number:</u> The value is the TCP octet that is expected next.
<u>Header length:</u> The number of 32-bit words in the TCP header, which indicates where the data begins. The TCP header (even one including options) is an integral number of 32 bits in length.
<u>Reserved:</u> Always set to zero.
<u>Code bits/flags:</u> Control bits for functions to manage a session. (SYN, ACK, FIN, URG, PSH, etc)
<u>Window:</u> The window size the sender is willing to accept, in octets.

Checksum: This checksum results in combining data such as addresses involved, segment size and types with a formula that can be checked on the receiving end (it is recalculated in a misleadingly- labeled 'pseudoheader' to do it) - it isn't a CRC which checks all data, but it is sufficient.
Urgent: A valid field only if the Urgent pointer in the code bits is set. If so, this value indicates the offset from the current sequence number, in octets, where the segment of non-urgent data begins.
Options: May be 0, meaning that no options have to be present, or a multiple of 32 bits. However, if any options are used that do not cause the option field to total a multiple of 32 bits, padding of 0s must be used to make sure the data begins on a 32-bit boundary (aka "words").
Data: Handed down to the TCP PDU by the upper-layer headers

```
TCP - Transport Control Protocol              Code:              %011000
 Source Port:         5973                             Ack is valid, Push Request
 Destination Port:    23                      Window:            61320
 Sequence Number:     1456389907               Checksum:         0x61a6
 Ack Number:          1242056456               Urgent Pointer:   0
 Offset:              5                        No TCP Options
 Reserved:            %000000                  TCP Data Area: vL.5.+.5.+.5.+.5  76 4c
```

User Datagram Protocol (***UDP***)

There are times that it's wise for developers to opt for UDP rather than TCP, one of them being when reliability is already taken care of at the Process/Application layer. Such was the case with old versions of NFS, (it later employed TCP). If the segments arrive out of order, which is commonplace in IP networks, they'll simply be passed up to the next layer in whatever order they were received. A UDP header is only 8 bytes (better than TCP's 20!) - has 4 fields, each of which are 2 bytes. Source and checksum are optional in IPv4 (only source is optional in IPv6)

```
UDP - User Datagram Protocol
 Source Port:         1085
 Destination Port:    5136
 Length:              41
 Checksum:            0x7a3c
 UDP Data Area:
 ..Z......00 01 5a 96 00 01 00 00 00 00 00 11 0000 00
```

Port numbers

TCP and UDP must use port numbers to communicate with the upper layers because these are what keep track of different conversations crossing the network simultaneously. Originating-source port numbers are dynamically assigned by the source host and will equal some number starting at 1024. The destination ports are mostly reserved to tell us what application they will be working with. Here are come common examples.

20-21 FTP
22 SSH/SCP
23 Telnet
25 SMTP
49 TACACS
53 DNS (uses both)
67-68 DHCP/BOOTP
69 TFTP
80 HTTP
88 Kerberos

110 POP3
123 NTP
135 Microsoft RPC
137-139 NetBIOS
143 IMAP4
161-162 SNMP
389 LDAP
443 HTTP over SSL
445 Microsoft DS
464 Kerberos

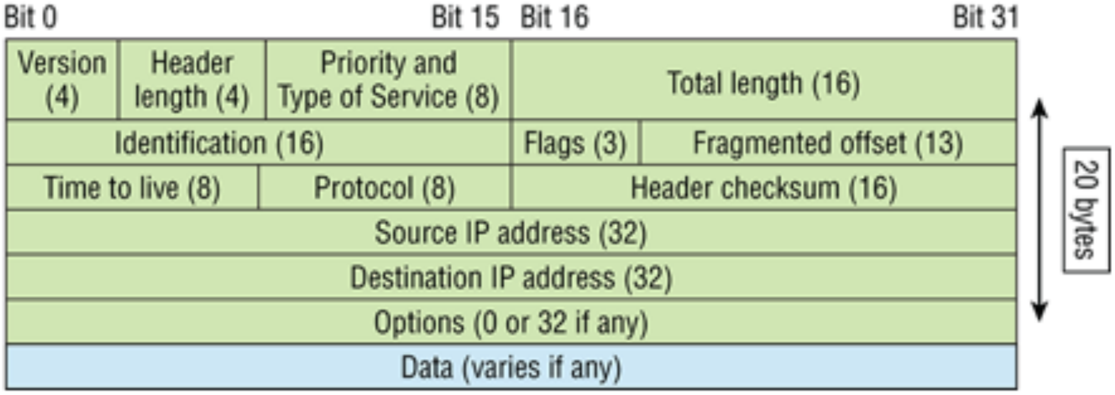
465 SMTP over SSL
500 ISAKMP
512 rexec
513 rlogin
514 syslog
515 LPD/LPR
520 RIP
521 RIPng (IPv6)
554 RTSP
546-547 DHCPv6

587 SMTP
636 LDAP over SSL
646 LDP (MPLS)
860 iSCSI
902 VMware Server
989-990 FTP over SSL
993 IMAP4 over SSL
995 POP3 over SSL
1025 Microsoft RPC

IANA policy assigns a port number for both TCP and UDP even if the service uses only one.
(The above protocols in italics use UDP - some may have been missed)

Layer 3 - Network/ Internet Layer - IPv4, IPv6, IPX, IPSec, ICMP. Has three main features: logical addressing, routing (forwarding), and path determination. Routing defines how devices (typically routers) forward packets to their final destination. Logical addressing defines how each device can have an address that can be used by the routing process. Path determination refers to the work done by routing protocols to learn all possible routes and determine the best way to move data.

The network layer handles two types: data packets and route update packets. *Ultimately, Layer 3 doesn't care about where a particular host is located- only about where networks are located and the best way to reach them*



Version: IP version number.
Header length: HLEN in 32-bit words.
Priority/TOS/Precedence: how to handle the datagram. The first 3 bits are the priority bits, called the differentiated services bits.
Total length: including header and data.
Identification: Unique IP-packet value used to differentiate fragmented packets from different datagrams.
Fragmentation Flags: Specifies whether fragmentation should occur.
Fragment offset: Provides fragmentation and reassembly if the packet is too large to put in a frame. It accommodates different MTUs on the Internet.
Time To Live: TTL set when packet is generated. If it doesn't get to where it's supposed to go before the TTL expires, it's dropped.
Protocol: Port of upper-layer protocol; for example, TCP is port 6 or UDP is port 17. Also supports Network layer protocols, like ARP and ICMP, and can referred to as the Type field in some analyzers.
Header checksum: CRC on header only.
Source IP address: 32-bit IP address of sending station.
Destination IP address: 32-bit IP address of the station this packet is destined for.
Options: Used for network testing, debugging, security, and more.
Data: The upper-layer data.

```
IP Header - Internet Protocol Datagram          Fragmentation Flags: %010 Do Not Fragment
 Version:              4                        Fragment Offset:     0
 Header Length:        5                         Time To Live:       60
 Precedence:               0                     IP Type:            0x06 TCP
 Type of Service:      %000                      Header Checksum:    0xd031
 Unused:               %00                       Source IP Address:  10.7.1.30
 Total Length:         187                       Dest. IP Address:   10.7.1.10
 Identifier:           22486                     No Internet Datagram Options
```

Protocols found in the Protocol field of an IP header
ICMP 1, TCP 6, UDP 17, IPv6 41, GRE 47, EIGRP 88, OSPF 89, L2TP 115
For more: http://www.iana.org/assignments/protocol-numbers

Layer 2 - Data Link Layer - (aka Network Access, Network Interface Layer)

ARP, MAC (IEEE 802.3 Ethernet, DSL, ISDN, FDDI), PPP, L2TP, HDLC Defines the rules that determine when a device can send data over a particular medium. Data link protocols also define the format of a header and trailer that allows devices attached to the medium to successfully send and receive data.

This layer transfers data between adjacent network nodes in a wide area network (WAN) or between nodes on the same local area network (LAN) segment. Frames do not cross the boundaries of a local network (unless encapsulated, these frames are replaced by new frames native to their environment). Internetwork routing and global addressing are higher-layer functions, allowing data-link protocols to focus on local delivery, addressing, and media arbitration (between parties contending for access to a medium, without concern for their ultimate destination). When devices attempt to use a medium simultaneously, frame collisions occur. Data-link protocols specify how devices detect and recover from such collisions, and may provide mechanisms to reduce or prevent them.

The data link layer has two sublayers: logical link control (LLC) and media access control (MAC):

Logical link control sublayer

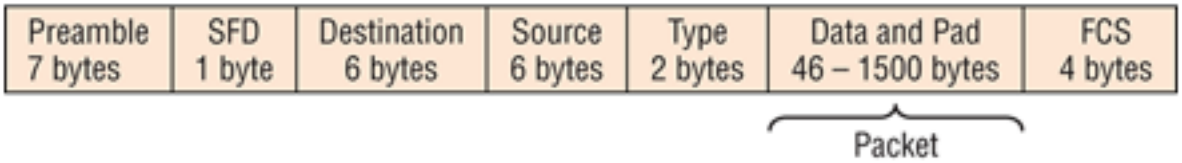
The uppermost sublayer, LLC, multiplexes protocols running atop the data link layer, and optionally provides flow control, acknowledgment, and error notification. The LLC provides addressing and control of the data link. It specifies which mechanisms are to be used for addressing stations over the transmission medium and for controlling the data exchanged between the originator and recipient machines. Data-link-layer error control is not used in LAN protocols such as Ethernet, but in modems and wireless networks.

Media access control sublayer

MAC may refer to the sublayer that determines who is allowed to access the media at any one time (e.g. CSMA/CD). Other times it refers to a frame structure delivered based on MAC addresses inside. The sublayer also determines where one frame of data ends and the next one starts- frame synchronization. There are four means of frame synchronization: time based, character counting, byte stuffing and bit stuffing.

MAC Services: physical addressing, channel-access control (CSMA/CD and CSMA/CA), LAN switching (packet switching), including MAC filtering, Spanning Tree Protocol and Shortest Path Bridging (SPB), data packet queuing or scheduling, Store-and-forward switching or cut-through switching, Quality of Service (QoS) control, Virtual LANs (VLAN)

Layer 2 Protocols - Ethernet Frames at the Data Link Layer



Preamble	SFD	Destination	Source	Type	Data and Pad	FCS
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46 – 1500 bytes	4 bytes

Preamble - 7 bytes - pattern of alternating 1 and 0 bits, allowing devices on the network to easily synchronize their receiver clocks. Since least significant bits are transmitted first, the one breaking the alternating pattern tends to be last.

Start Frame Delimiter (SFD): 1 byte - The SFD octet is designed to break the bit pattern of the preamble and signal the start of the actual frame. Arguably, the SFD could more easily be called part of the preamble.

Destination Address (DA) - 6 bytes - The 48-bit value of the intended recipient. Can also be BCast or MCast.

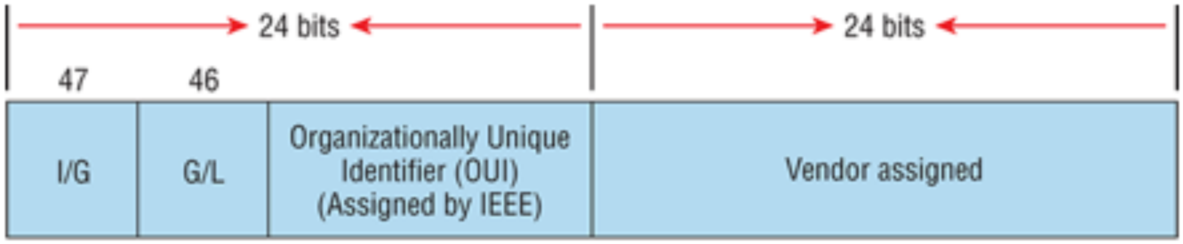
Source Address (SA) - 6 bytes - Sender. BC and MC address formats are illegal within the SA field.

802.1Q tag (optional) - 4 bytes - Ethernet II only

Length or Type - 2 bytes - Old 802.3 uses a Length field, but Ethernet_II frame uses a Type field to ID the layer 3 PDU. 0x86dd for IPv6 data; 0x0800 for IPv4; 0x8100 for VLAN-tagged (802.1Q); 0x0806 for ARP;

0x8147/48 MPLS; 0x88CC LLDP; 0x8163/64 PPPoE; 0x8906 FCoE; 0x8100 Jumbo;
Data - 46-1500 bytes - This is a packet sent down to the Data Link layer from the Network layer. The size can vary from 46 to 1,500 bytes. (44 if 802.1Q tag is present, and greater than 1500 if jumbo). Padding to meet the minimum 46 bytes is added if necessary.
Frame Check Sequence (FCS/CRC) - Algorithm is run when each frame is built based on the data in the frame. If CRCs don't match, the frame is discarded, assuming errors have occurred.

Data Link and Ethernet - Hex to Dec and MAC Addresses



Example: 0000.0c12.3456

48-bit (6-byte) MAC address written in a hexadecimal format (instead of binary for readability).
The organizationally unique identifier (OUI) is assigned by the IEEE to an organization. It’s composed of 24 bits, or 3 bytes, and it in turn assigns a globally administered address

Individual/Group (I/G) bit. When it has a value of 0, we can assume that the address is the MAC address of a device and that it may well appear in the source portion of the MAC header. When it’s a 1, we can assume that the address represents either a broadcast or multicast address in Ethernet.”

Global/local bit, sometimes called the G/L bit or U/L bit, where U means universal. When set to 0, this bit represents a globally administered address, as assigned by the IEEE, but when it’s a 1, it represents a locally governed and administered address. This becomes clearer when discussing IPv6 and MAC addresses.

Manufacturer-assigned code commonly starts with 24 0s for the first card made and continues to 16,777,216 (24 ones in binary). It is often incorporated into the serial number as well.

Hex to Binary conversion: A "nibble" is 4 bits and a byte is 8 bits (or an octet)
If we have a 1 placed in each spot of our nibble, we would then add up 8 + 4 + 2 + 1 to give us a maximum value of 15. Another example for our nibble values would be 1001, meaning that the 8 bit and the 1 bit are turned on, which equals a decimal value of 9. If we have a nibble binary value of 0110, then our decimal value would be 6, because the 4 and 2 bits are turned on. You take these two nibbles as binary and run them together as a byte: you get the real value 10010110, which is 128+16+2+4 = 150

Example: 0x6A- each hex character is one nibble and that two hex characters joined together make a byte. To figure out the binary value, put the hex characters into two nibbles and then join them together into a byte. 6 = 0110; A, which is 10 in hex = 1010; so the complete binary byte would be 01101010, and 64+32+8+2= 106 is the decimal value.

A binary number: 11001100. What's it in hex? Split it: 1100 = 12 and 1100 = 12, so therefore, it’s representation is CC in hex, but the decimal conversion would be 128 + 64 + 8 + 4 = 204, the real value.

10110101 - The hex answer would be 0xB5, since 1011 converts to B and 0101 converts to 5 in hex value. The decimal equivalent is 128 + 32 + 16 + 4 + 1 = 181

Hex	***Bin***	***Dec***
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Layer 2 Protocols - Address Resolution Protocol (ARP)

ARP was defined by RFC 826 in 1982, and is used for mapping a IPv4 address to a physical MAC address. In IPv6 the functionality of ARP is provided by the Neighbor Discovery Protocol (NDP). When IP has a datagram to send, it must inform a Network Access protocol, such as Ethernet or wireless, of the destination's hardware address on the local network. If IP doesn't find the destination host's hardware address in the ARP cache, it uses ARP to find this information.

```
ARP - Address Resolution Protocol
 Hardware                     1 (Ethernet 10Mb)
 Protocol:                    0x0800 (IP)
 Hardware Address Length:     6
 Protocol Address Length:         4
 Operation:                   1 ARP Request
 Sender Hardware Address:     00:A0:24:48:60:A5
 Sender Internet Address:     172.16.10.3
 Target Hardware Address:     00:00:00:00:00:00 (ignored)
 Target Internet Address:     172.16.10.10
```

Inverse ARP is primarily used in Frame Relay (DLCI) and ATM networks, which need the corresponding Layer 3 addresses before those virtual circuits can be used

An ARP probe is an ARP request constructed with an all-zero sender IP address. Before beginning to use an IPv4 address a host must test to see if the address is already in use, by broadcasting ARP probe packets

A gratuitous ARP announcement updates any cached entries in the ARP tables of other hosts that receive the packet. Many operating systems perform gratuitous ARP during startup. It helps in case a network card was recently changed and other hosts need to update ARP caches. Gratuitous ARP is also needed in teaming network cards, and can be used to defend link-local IP addresses in the Zeroconf.

ARP spoofing - ARP has no authentication, so replies can come from systems other than the one with the required address. An ARP proxy is a legitimate system which answers the ARP request on behalf of another system for which it will forward traffic, such as for a dialup internet service. ARP spoofing is where a system impersonates another system's address with the aim of intercepting data bound for that system, resulting in man-in-the-middle and DoS attacks.

ARP mediation is WAN resolution of Layer 2 addresses through a Virtual Private Wire Service (VPWS) when different resolution protocols are used circuits- e.g., Ethernet on one end and Frame Relay on the other. In IPv4, each Provider Edge (PE) device discovers the IP address of the locally attached Customer Edge (CE) device and distributes that IP address to the corresponding remote PE device. Then each PE device responds to local ARP requests using the IP address of the remote CE device and the hardware address of the local PE device. In IPv6, each PE device discovers the IP address of both local and remote CE devices and then intercepts local Neighbor Discovery (ND) and Inverse Neighbor Discovery (IND) packets and forwards them to the remote PE device.

ICMP

Destination unreachable - a router can't send an IP datagram any further
Buffer full/source quench - a router's buffer for incoming datagrams is full.
Hops/time exceeded- last router says it reached limit before arriving at destination
Ping and Traceroute - Packet Internet Groper (Ping) echo-reply messages to check connectivity.
Traceroute is used to reveal the path a packet takes by using ICMP errors, time-outs. and TTL.

The Ping program uses the alphabet in the data portion of the packet as a payload, typically around 100 bytes by default, unless, of course, you are pinging from a Windows device, which thinks the alphabet stops at the letter W (and doesn't include X, Y, or Z) and then starts at A again.

***Layer 1 - Physical Layer* -** These standards deal with the physical characteristics of the transmission medium, including connectors, pins, use of pins, electrical currents, encoding, light modulation, and the rules for how to activate and deactivate the use of the physical medium.

The Variety of Ethernet Physical Layer Standards

10 Mbps	Ethernet	10BASE-T	802.3	Copper, 100 m
100 Mbps	Fast Ethernet	100BASE-T	802.3u	Copper, 100 m
1000 Mbps	Gigabit Ethernet	1000BASE-LX	802.3z	Fiber, 5000 m
1000 Mbps	Gigabit Ethernet	1000BASE-T	802.3ab	Copper, 100 m
10 Gbps	10 Gig Ethernet	10GBASE-T	802.3an	Copper, 100 m

Ethernet at the Physical Layer
EIA/TIA (Electronic Industries Alliance/ Telecommunications Industry Association)
Most common IEEE Ethernet standards, starting with 10 Mbps Ethernet:
10Base-T (IEEE 802.3) 10Mbps- CAT 3 UTP - 100 meters.
100Base-TX (IEEE 802.3u) 100Base-TX aka Fast Ethernet- CAT 5, 5E, or 6 UTP- 100 meters
100Base-FX (IEEE 802.3u) 62.5/125-micron MMF. Point-to-point; up to 412 meters long. ST or SC.
1000Base-CX (IEEE 802.3z) Twinax, balanced coax up to 25 meters - 9-pin connector- High Speed Serial Data Connector (HSSDC). Cisco's Data Center technologies.
1000Base-T (IEEE 802.3ab) Category 5, four-pair UTP wiring up to 100 meters long and up to 1 Gbps.
1000Base-SX (IEEE 802.3z) 1GEthernet MMF- 850nm laser. 220 m 62.5-micron core, 550m 50-micron
1000Base-LX (IEEE 802.3z) SMF 9-micron core, 1300 nm laser, 3km-10km.
1000Base-ZX (Cisco std) Cisco specified for GB Ethernet. Up to 43.5 miles (70 km).
10GBase-T (802.3.an) 10Gbps- CAT 5e, 6, or 7 cables. 100-meter

Meaningless formal definitions:
"Encoding Scheme"- The transmitting node changes the electrical signal over time, while the other node, the receiver, using the same rules, interprets those changes as either 0s or 1s

Cisco physical ports whose port hardware can be changed later
Gigabit Interface Converter (GBIC), arrived the same time as Gigabit Ethernet, so it was given the same "gigabit" name; newer Small Form-factor Pluggable (SFP), provide the same function of giving users the ability to swap hardware and change the type of physical link

Auto-MDIX that notices when the wrong cable is used and automatically changes its logic to make the link work. Don't assume it exists.

Crossover cable: If the endpoints transmit on the same pin pair, crosses them for us
Straight-through cable: If the endpoints transmit on different pin pairs
(Use crossover for "like devices" - not unlike)

Transmits on Pins 1,2: workstation NICs, routers, cabled non-USB WAPs
Transmits on Pins 3,6: switches and hubs

Straight-through cable: If the endpoints transmit on different pin pairs
Crossover cable: If the endpoints transmit on the same pin pair,

UTP Cabling Pinouts:
10BASE-T and 100BASE-T = 2 pairs (4 wires)
Straight-Through Cable Pinout = 2 pairs use 1 and 2 to 1 and 2 on the other end (transmit to receiver), and 3 and 6 to 3 and 6 on the other end (receive from transmit)
Crossover is 1 and 2 to 3 and 6 on the other end

1000BASE-T= 4 pairs (8 wires)
The straight-through cable connects each pin with the same numbered pin on the other side, but it does so for all eight pins—pin 1 to pin 1, pin 2 to pin 2, up through pin 8. It keeps the one pair at pins 1 and 2 and another at pins 3 and 6, just like in the earlier wiring. It adds a pair at pins 4 and 5 and the final pair at pins 7 and 8

The Gigabit Ethernet crossover cable crosses the same two-wire pairs (pins 1,2 and 3,6). It also crosses the two new pairs as well (the pair at pins 4,5 with the pair at pins 7,8).

Rollover cable for switch and router console.   USB converter, serial, rollover, CONSOLE port connector.

Use a rolled Ethernet cable to connect host EIA-TIA 232 interface to router console serial COM port
Probably the easiest cables to make because you just cut the end off on one side of a straight-through cable, turn it over, and put on a new connector

Media Access Control (MAC) addresses, are 6-byte-long (48-bit-long) binary numbers. For convenience, most computers list MAC addresses as 12-digit hexadecimal numbers.

" MAC addresses represent a single NIC or other Ethernet port, so these addresses are often called a unicast Ethernet address."

Fundamentally, the purpose of routers is to segment a network into broadcast domains.
Each port on a router is a separate broadcast domain (and thus also a separate collision domain).
Each port on a switch is a separate collision domain. (bridges did something similar, so some consider switches to be multiple-port bridges with more intelligence)
Reduce collisions in a BC domain by adding more collision domains with switches, but remember that this adds more hosts to the broadcast domain!

Never to allow broadcast domains to grow too large and get out of control. Both collision and broadcast domains can easily be controlled with routers and VLANs

Routers do ARP - switches dont.

Routers, by default, will
- not forward any broadcast or multicast packets.
- use the logical address in a Network layer header to determine the next-hop
- use administrator-created ACLs for types of packets allowed to enter or exit an interface.
- provide connections between VLANs and can provide quality of service (QoS)

Routers can provide layer 2 bridging functions if needed and can simultaneously route through the same interface.

A *routing protocol* is designed to update routing tables between routers, reducing or eliminating the need for manual configuration. They determine the path of a packet through an internetwork by ensuring that all routers have matching routing tables. Examples of routing protocols are RIP, RIPv2, OSPF, EIGRP, IS-IS, BGP.

A *routed protocol* is used to transport user traffic from source node to destination node through the established internetwork. Routed protocols are assigned to an interface and determine the method of packet delivery. Examples of routed protocols are IPv4, IPv6, IPX and AppleTalk.
The output of the ipconfig command provides the basic routed protocol information on a host.

Queuing is generally considered “congestion management”
When a device, such as a switch or a router, receives traffic faster than it can be transmitted, the device attempts to buffer (that is, store) the extra traffic until bandwidth becomes available. This buffering process is called queuing or congestion management.
Queuing is the temporary storing of data in accordance with its priority.

Found in a test question:
For Layer 2, isochronous transmission uses little network overhead when compared to asynchronous or synchronous transmission methods, because isochronous does not need to provide clocking
- at the beginning of a data string (as does synchronous transmission)
- or for every data frame (as does asynchronous transmission)

Data Center Bridging Capability Exchange protocol (DCBX)

The TCP handshake and termination - quick overview:

TCP Handshake:
PC1 --> SYN --> PC2
PC1 <-- SYN, ACK <-- PC2 --
PC1 --> ACK --> PC2

TCP Termination:
PC1 --> ACK, FIN --> PC2
PC1 <-- ACK <-- PC2
PC1 <-- ACK, FIN? <-- PC2
PC1 --> ACK --> PC2

The TCP Handshake in Detail:

The Initial Sequence and Response Numbers (ISN and IRN) are numbers exchanged in TCP segments during computer network communication between a client and a server. These are central in a SYN flood defense known as SYN cookies.

Here is a sample session:
1. The client sends a SYN with an ISN of 1664882716.

2. Server replies with a SYNACK with an IRN of 829007135 and an ACK value of 1664822717. The ACK reports the next the server expects from the client in this sequence (1664822717)

3. Client sends an ACK back 829007136 to increment the server's IRN in the SYNACK, which also reports it expects from the server in this sequence (829007136) It sends this with a sequence number of 1664882717, just like the server expects.

And it keeps going. Here is the whole thing:
1. Client: SYN seq 1664882716
2. Server: SYNACK seq 829007135 1664882717
3. Client: ACK seq 1664882717 829007136
4. Server: ACK seq 829007136 1664882718
5. Client: ACK seq 1664882718 829007137

Then later, server terminates the connection:
Client: ACK seq 1664882733 829008199
Server: FIN-ACK seq 829008199 1664882734
Client: ACK! seq 1664882734 829008200
Client: FIN-ACK? seq 1664882734 829008200 (yes, two different responses)
Server: ACK seq 1664882735

Switching basic decisionmaking:
IEEE 802.1D/w Spanning Tree puts each port in forward or blocking state. Ports in a blocking/ discarding state won't process any frames except STP messages - the switch physically receives the frame on blocked port, but ignores it.

That being said, for all posts in a forwarding state:
If destination address is:
- *same as source? Ignore (filter).*
- *known? Forward to correct port*
- *unknown? LEARN source MAC to port#. Flood out all ports except entry*

Address learning:
- When flooded out, source MAC is added to table, but the destination will NOT be learned.

For each received frame, examine the source MAC, note the interface.
- If not in table, add- *set inactivity timer to 0.*
- If it is in table, *reset the inactivity timer for the entry to 0.*

On new frame, if MAC-to-port is different than previously recorded, it will be updated.
- MAC table instability (info repeatedly updated erroneously from loops) is prevented by STP

Modes of forwarding:
Store-and-forward - receive all bits in the frame and check the FCS before forwarding.
Fragment-free - receiving the first 64 bytes first to weed out collision-damaged frames.
Cut-through - checks dest MAC, forwards frame ASAP -reduces latency but no FCS check (later versions check for QoS and/or ACLs)

If asked, the three switch Layer 2 functions: address learning, forward/filter, loop avoidance (STP)

Forward/ Filter? MAC table says for Port 1, not port 2, forward to port 1, filter from port2.
(Stupidly, it is looked at by some people as two decisions:)
1. Forwarding decision: to send it to the right port (associated with that MAC address)
2. Filtering decision: to NOT send it out the other ports.

IP Routing basic decisionmaking:

<u>Four-step process of how routers route (forward) packets</u>

Layer 2 first- check FCS. If errors occurred, discard the frame.

Trash the old layer 2 header and trailer, leaving the IP packet.

Compare to the routing table, and find the outgoing interface of the router (next-hop IP).

Encapsulate into new layer 2 header/trailer for the outgoing interface and forward the frame. (how does it know the destination MAC? The ARP table!)

Goals of IP routing protocols

- dynamically learn and fill the routing table with a route to each subnet on the network.
- choose and place the best route in the routing table.
- notice when table's routes are no longer valid, remove them. Get new one from neighbor.
- work quickly - for fast convergence time and routing updates
- prevent routing loops.

The TCP Header

Remember a "word" is 4 bytes:
"The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of 20 bytes and maximum of 60 bytes, allowing for up to 40 bytes of options in the header"

Source Port, Destination Port
Sequence Number
Acknowledgement Number
Offset, Reserved, Flag Bits, Window
Checksum, Urgent

<u>Switch Delegation</u>

The following list summarizes the terms that describe the roles in a hierarchy switches:
Access layer: For end-user devices. Doesn't normally forward frames between two other access switches.
Distribution: Aggregation point for access switches, forwarding, not connecting directly to end-user devices.
Core: Aggregates distribution switches in very large LANs, providing very high forwarding rates.

<u>Some of the benefits of switching:</u>

- Switch ports for a single device microsegment the LAN, providing dedicated bandwidth to that device.
- Switches allow multiple simultaneous conversations between devices on different ports.
- Switch ports for a single device support full-duplex, doubling the amount of bandwidth available to the device.
- Switches support rate adaptation, which means that devices that use different Ethernet speeds can communicate through the switch (hubs can't).

Broadcast and Collision Domains:

- Collision domain: *a set of NICs* for which a frame sent by one NIC could result in a collision with a frame sent by any of the other NICs.
- Broadcast domain: *a set of NICs* for which a broadcast frame sent by one NIC is received by all other NICs.

Collisions happen on "same segments" and segment there doesn't mean a wire (hubs)

Each port on a switch (or bridge) is a separate collision domain.
Each port on a router is a separate broadcast domain (and thus also a separate collision domain).

- *But what about VLANs? Doesn't multiple VLANs on a switch mean multiple broadcast domains?*

All ports on a switch by default are part of one broadcast domain (VLAN1 - the native VLAN).
Fundamentally, it's the purpose of <u>routers</u> to segment a network into multiple broadcast domains.

With multiple configured VLANs, a switch groups interfaces into specific broadcast domains with each interface in a specific VLAN. So a switch can relay multiple broadcast domains by putting some interfaces into different VLANs, BUT these multiple broadcast domains require a router to properly interact among them.

Adding collision domains means increased bandwidth
- one sender at a time per collision domain (devices share the bandwidth):
- Hub: only one PC can send at a time, for a theoretical maximum capacity of 100 Mbps for the entire LAN.
- Replace with a switch, and you get 100 Mbps per link, for a total of 1000 Mbps (1Gbps) and the ability to use full-duplex on each link, effectively doubling the capacity to 2000 Mbps (2Gbps)

Broadcasts leading to DoS
Finally, remember- when a **host** receives a broadcast, the host *must process* the received frame. This means that the NIC must interrupt the computer's CPU, and the CPU must spend time thinking about the received broadcast frame.

Duplex
Half-duplex is usually with hubs, less than or equal to 10Mbps, needs CSMA/CD for collisions.
On collision, a jamming sig sent (backoff) and devices set timers to try again.

CPE - Customer Premise Equipment
Leased line/circuit; serial is just bits sequentially; T1 is 1.544 Mbps; customer premise equipment (CPE)
Channel service unit/data service unit (CSU/DSU), Ethernet over MPLS- physical connections between router and external CSU/DSU are DTE serial cables. May also need also a 2nd corresponding serial DCE cable. Clock is set by DCE

HDLC instead of ethernet for point to point. doesn't need addresses, uses them since telcos once offered multidrop circuits. In HDLC frames, Flag is like EN Preamble, also uses a "control" field instead of a "type" field.
Ethernet frames are broken down and re-encapsulated as HDLC frames. "Ethernet emulation" and Ethernet over MPLS (EoMPLS) just means line to SP uses ethernet frames, which are de-encapsed and re-encapsed (as any other frame) with new source/destination MAC addresses.

Remember that DSL and cable are not considered "leased lines"
DSLAM simply modulates from a voice switch, while cable separates video from data links.

Beginning to end:

-New PC is plugged into the network.
It's NIC sends out a broadcast ARP with it's MAC address
Switch adds it to the table and initiates the MAC's inactivity timer
-(security is checked also) switchport port-security violation {protect|restrict|shutdown}

Simple Network - Routing and Switching Overview
HostA (172.16.10.2) pings HostB (172.16.20.5) through one Router

1- ICMP's ping process creates it's echo request payload, which is simply the alphabet in the data field

2- ICMP hands payload segment to IP to form a packet. At a minimum, this packet contains an IP source address, an IP destination address, and a Protocol field with 0x01

3- IP checks to see if the destination IP address is a device on the local LAN or on a remote network. Since the destination device is on a remote network, the packet needs to be sent to the default gateway, so it finds it in the routing table.
(frames can't just be dropped in another subnet- only it's local network. It needs a gateway router to get to a remote network like 172.16.20.0. Remember that hardware addresses are always local, and never pass through a router's interface)

4- It finds the default GW is 172.16.10.1, so the ARP cache is checked for the MAC if it has already been resolved. If the ARP cache doesn't have it, an ARP broadcast is sent out on local network for 172.16.10.1, router responds, and the cache updated with the info and an activity timer for it.

It turns out that the ARP table had it, and says it's the router's interface with the MAC address on Ethernet0, so it's activity timer is reset to 0, and the packet is *then* ready to be handed to a different part of the Data Link layer for framing.

5- With destination gateway and source MAC addresses, ether-type (0x0800 for IP) and the packet payload, the LAN driver is used to provide media access via the type of LAN (Ethernet), a frame is then generated, adding the FCS field (the result of the CRC).

6- Frame is given to the wire bit-by-bit.

THE FRAME HAS FINALLY ENTERED THE NETWORK

ARP is known to operate at Later 2, which is supposedly not aware of IP addresses, a contradiction encouraging people to refer to ARP as working at "Layer 2.5". The only other way to put it is that ARP is a layer 2 service call invoked from Layer 3. To make matters even more confusing, ARP and ethernet framing fall under the definition of the media access control sublayer of the data-link layer, rather than the upper logical link control sublayer! Just remember that if you are asked where ARP is on a test, it is Layer 2.

7- All devices on the collision domain get the bits, rebuild the frame, and run a CRC to compare the FCS field for integrity. If it fails the integrity check it's discarded.
(note: in this scenario we are talking about a direct connection to a router, it be a switch and it would be just HostA and the interface in the collision domain. The above would be more relevant if it was a hub with more devices in a collision domain)

8- If it passes, the destination MAC is checked to see if it matches it's own address. Eth0 on the router gets it, checks the ether-type field and gives the enclosed packet to IP (the rest is discarded).

9- IP on the router checks the packet header for errors (it doesn't run a complete CRC like the Data Link layer does on the frame). Since the packet's destination IP doesn't match any of the addresses configured on the router's interfaces, the router will look in its routing table.

If there is no entry for the network 172.16.20.0 the packet will be discarded immediately and an ICMP message will be sent back to the originating device with a destination network unreachable message.

In this case it was found- the packet is routed to the exit interface (Eth1) for that subnet. Since the routing table shows "directly connected," no routing protocols are needed.

10- As the IP layer hands the packet to Layer 2, the MAC of HostB's IP address is either found in the ARP cache, or an ARP request is sent out for it (just as before in steps 3-6). We can safely say that the IP packet has been switched to a buffer before it's framing. It's framed, and sent out, just like in steps 5 and 6.

THE FRAME HAS LEFT THE ROUTER

11- HostB receives the frame and immediately runs a CRC. If the result matches the information in the FCS field, the hardware destination address will be then checked next. If the host finds a match, the Ether-Type field is finally checked to determine the Network layer protocol that the packet belongs with (IPv4), and it is passed up the stack.

12- IP gets the packet, runs a CRC on the header, checks the destination address and sees it's own address, so it checks Protocol field to find out which Layer 4 protocol gets the payload segment.

DESTINATION REACHED!

13- The payload is handed to ICMP, which understands that this is an echo request. ICMP responds to this by immediately discarding the packet and generating a new payload as an echo reply.

14- A packet is then created including the source and destination addresses, Protocol field, and payload. The destination device is now HostA.

15- Steps 3-6 are repeated on HostB: determining it goes to the gateway's IP address, getting the gateway's MAC, framing it, and sending it to wire for the router.

16- The router's Eth1 interface receives the bits and builds a frame. The CRC is run, and the FCS field is checked to make sure the answers match. It's handed to IP and steps 7-10 are repeated with destination swapped to HostA

17- To route the packet, the device knows to get to network 172.16.10.0 it should exit Eth0, and the MAC for 172.16.10.2 is already cached from the originating trip to HostB, so it builds a frame and sends it out on the wire.

If a packet is lost on the return trip to the originating host, you will usually see a request timed-out message when the response payload isn't received by the ICMP ping process (the original known payload got to the destination). If the error occurs in transit to the destination device, like a route is not in the routing table, you will see a destination unreachable message returned.

(LEAVES THE ROUTER)

18- The destination host receives the frame, runs a CRC, checks the destination MAC, then looks at the Ether-Type field, saying to hand it to IP at the Network layer, it checks the Protocol field which says ICMP, and ICMP determines the packet to be an ICMP echo reply.

19- ICMP acknowledges that it has received the reply by sending an exclamation point (!) to the user interface. ICMP then attempts to send four more echo requests to the destination host."

ASIDE NEEDED FOR ANOTHER PART: **On ARP: Layer 2-3 Confusion**

ARP (IPv4) and NDP (IPv6) operate at the media access control (MAC) layer- the lower sublayer of the data link layer responsible for physical addressing.

Considering that in the "real world" the layers are for our figurative, simplified understanding- we have to remember that Layer 2 knows nothing about IP addressing- so ARP in this usage is like a invoked layer 2 service call,

□ **Logical Link Control (LLC):**
functions required for the establishment and control of logical links between local devices
provides services to the network layer and hides the rest of the data link layer
IEEE 802.2 LLC protocol.
□ **Media Access Control (MAC):** This refers to the procedures used by devices to control access to the network medium. Since many networks use a shared medium (such as a single network cable, or a series of cables that are electrically connected into a single virtual medium) it is necessary to have rules for managing the medium to avoid conflicts. For example. **Ethernet uses the CSMA/CD method of media access control,** while Token Ring uses token passing.

Another source says about the Ethernet reference model: 802 is MAC-client(upper layer), and 802.3 is MAC (lower layer), and if you get in really ugly detail you can look at things that go between those and the physical layer like the reconciliation layer. I consistently see LLC, MAC Control, and MAC in 3-sublayer models for Ethernet layer 2.

On Autonegotiation
Default autonegotiation actions:
 - Speed: Sense the speed by wire signal. If that fails, use IEEE default (slowest supported, often 10Mbps).
 - Duplex: If speed = 10 or 100Mbps, use half-duplex; otherwise, use full-duplex.
If the speed is not known, use 10 Mbps, half-duplex. When one device has no autonegotiation, and the other does, use speed's defaults.
When displaying interface settings, duplex and speeds set by autonegotiation are prefixed with an "a-"

in show interfaces status: duplexes a- means autonegotiated. just half or full (not a-full) means set manually)
needs to be gigabit to default to full duplex

<u>Data Capacity - The beercan analogy:</u>
DS0 - 56/64Kbps Modem - 1 voice/data phone line - One can
DS1/T1 - 1.544 Mbps [1.536 Mbps + framing/control bits] - a case of beer (24 cans)
DS3/T3 - 44.736 Mbps - 28 DS1's (672 DS0) - a pallet of beer with 28 cases
-- End of copper capacity - begin SONET --
OC1 = 1 DS3 over fibre - a pallet of 28 cases shrink-wrapped together
OC3 - 155.52 Mbps = 3 DS3's = 84 DS1's = 2016 DS0's - a truck that can hold 3 pallets
OC12 - 622.08 Mbps = 12/DS3 = 336/DS1 = 8064/DS0 - a railroad train car - 12 pallets
OC48 - 2488.32 Mbps = 4 OC12's = 48 DS3's = 1344/DS1's = 4 train cars x 12 pallets
OC192 - 9953.28 Mbps = 16/OC12 = 192/DS3 = 5376/DS1 = 192,024/DS0 - 16 train cars
OC768 - 39813.12 Mbps = 64/OC12 = 768/DS3 = 21,504/DS1 - a train with 64 railroad cars

 - OC48 popular choice for many regional networks, several deployed in parallel when bandwidth needs haven't reached the level requiring investments in OC192 or 10 gigabit Ethernet
 - OC192 and OC768 were deployed more with bandwidth needs, lower hardware prices, perhaps with limited fibre availability. Often skipped to reduce expense and complexity for 10 gigabit Ethernet fibre lines.

<u>Bytes to bits in bandwidth conversions</u>
1 Kbps = 1000/8=125 bytes
1 Mbps = 1,000,000/8=125,000 bytes
Bytes are made up of eight bits, so one kilobyte equals eight kilobits
1KB per sec = 8 Kbps
1MB per sec = 8 Mbps

"A megabit per second (abbreviated as Mbps, Mbit/s, or mbps) is a unit of data transfer rates equal to 1,000,000 bits per second (this equals 1,000 kilobits per second). Because there are 8 bits in a byte, a transfer speed of 8 megabits per second (8 Mbps) is equivalent to 1,000,000 bytes per second (approximately 976 KiB/s)"