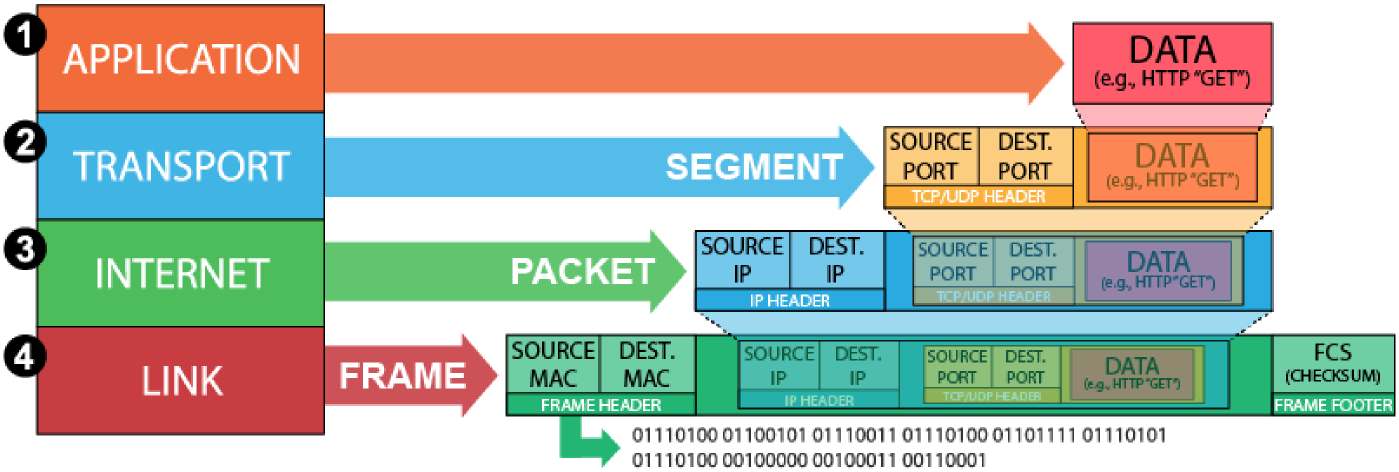
Data Encapsulation - Basics

Encapsulation is the process of breaking a message into packets; adding control and other information; and then transmitting the message through the transmission medium. The following diagram uses the TCP/IP model to describe the four-step data encapsulation process:



1. The Application layer prepares the data to be sent through the network.
2. The Transport layer breaks the data into pieces called segments, adding sequencing and control information.
3. The Internet layer converts the segments into packets, adding logical network and device addresses.
4. The Link layer converts the packets into frames, adding physical device addressing information and a frame check sequence (FCS) footer for error detection. It also converts the frames into bits (0s and 1s) for transmission across the transmission media.

On the destination host, the process operates in reverse, with bits from the network medium being received by the Link layer and being processed up the model to the destination application.

The encapsulation process works in the same manner using the OSI model. As data travels through the OSI model layers, it is broken into segments at the Transport layer. Logical addresses are added at the Network layer, making each segment a packet. The Data Link layer creates frames from each packet using the physical device address (MAC address). Frames are converted to bits at the Physical layer.

Data Encapsulation: TCP/IP and OSI Protocol Suites

Taken from my technical writeup on Data Encapsulation:
“Data encapsulation occurs as data from a sender is sent to a receiver. At each logical step of the data encapsulation process, the data becomes larger as new information is added to it. Both the TCP/IP and OSI models use data encapsulation processes to facilitate and regulate how data transmission should occur over a network.”

TCP/IP Protocol Suite

The TCP/IP model applies a four-step data encapsulation process that directly corresponds to the four layers of the TCP/IP protocol suite.

TCP/IP layers and their functions:

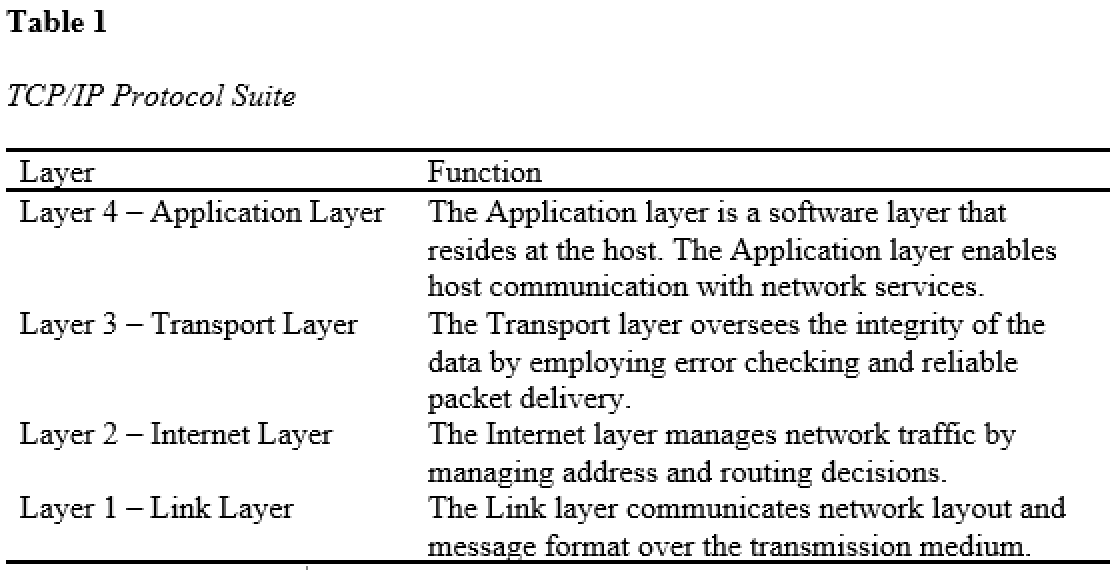


Table 1

TCP/IP Protocol Suite

Layer	Function
Layer 4 – Application Layer	The Application layer is a software layer that resides at the host. The Application layer enables host communication with network services.
Layer 3 – Transport Layer	The Transport layer oversees the integrity of the data by employing error checking and reliable packet delivery.
Layer 2 – Internet Layer	The Internet layer manages network traffic by managing address and routing decisions.
Layer 1 – Link Layer	The Link layer communicates network layout and message format over the transmission medium.

TCP/IP data encapsulation process:

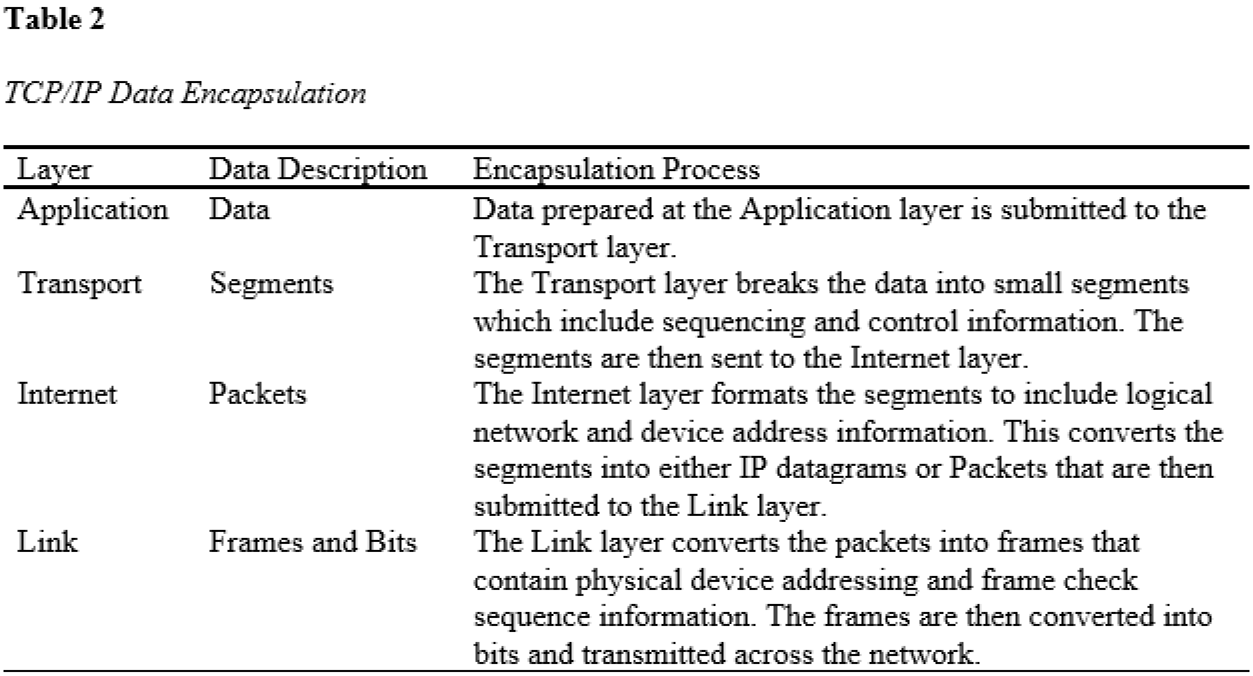


Table 2

TCP/IP Data Encapsulation

Layer	Data Description	Encapsulation Process
Application	Data	Data prepared at the Application layer is submitted to the Transport layer.
Transport	Segments	The Transport layer breaks the data into small segments which include sequencing and control information. The segments are then sent to the Internet layer.
Internet	Packets	The Internet layer formats the segments to include logical network and device address information. This converts the segments into either IP datagrams or Packets that are then submitted to the Link layer.
Link	Frames and Bits	The Link layer converts the packets into frames that contain physical device addressing and frame check sequence information. The frames are then converted into bits and transmitted across the network.

OSI Model

The OSI protocol stack is a reference model that classifies how data is theoretically prepared and transported across a network. It is the most widely used method to discuss and understand network communication. The seven layers within the OSI protocol stack and their functions are presented in Table 3.

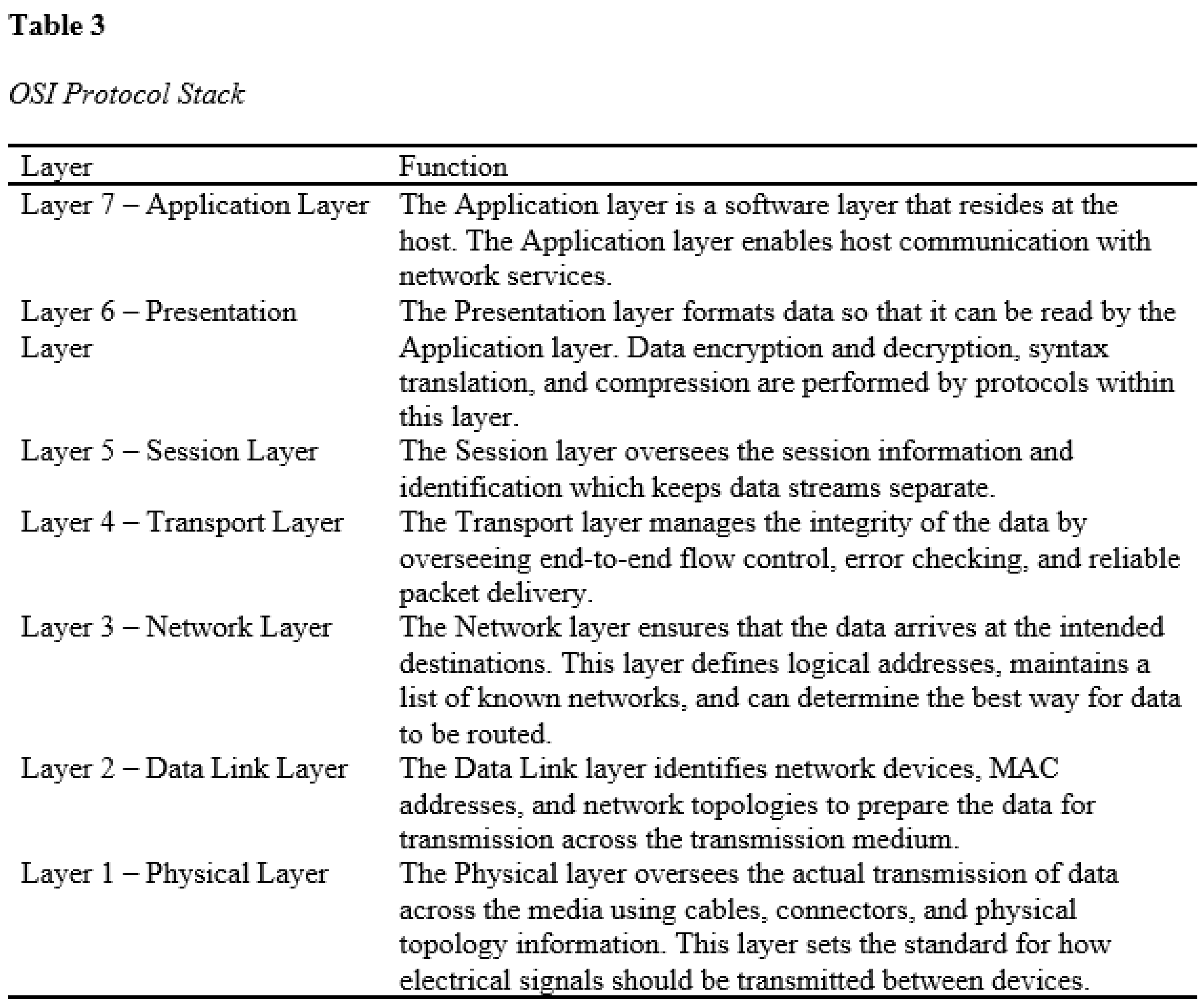


Table 3

OSI Protocol Stack

Layer	Function
Layer 7 – Application Layer	The Application layer is a software layer that resides at the host. The Application layer enables host communication with network services.
Layer 6 – Presentation Layer	The Presentation layer formats data so that it can be read by the Application layer. Data encryption and decryption, syntax translation, and compression are performed by protocols within this layer.
Layer 5 – Session Layer	The Session layer oversees the session information and identification which keeps data streams separate.
Layer 4 – Transport Layer	The Transport layer manages the integrity of the data by overseeing end-to-end flow control, error checking, and reliable packet delivery.
Layer 3 – Network Layer	The Network layer ensures that the data arrives at the intended destinations. This layer defines logical addresses, maintains a list of known networks, and can determine the best way for data to be routed.
Layer 2 – Data Link Layer	The Data Link layer identifies network devices, MAC addresses, and network topologies to prepare the data for transmission across the transmission medium.
Layer 1 – Physical Layer	The Physical layer oversees the actual transmission of data across the media using cables, connectors, and physical topology information. This layer sets the standard for how electrical signals should be transmitted between devices.

*The layer assignment is typically selected based upon when that protocol is first initiated.

An overview of the OSI model data encapsulation process is presented in Table 4.

Table 4

OSI Data Encapsulation

Layer	PDU	Encapsulation Process
Application	Data	Data prepared at the Application layer is submitted to the Presentation layer.
Presentation	Data	The data is encrypted at the Presentation layer and sent to the Session layer.
Session	Data	The data receives a session ID at the Session layer and is sent to the Transport layer.
Transport	Segment PDU	The Transport layer breaks the data into small segments which include sequencing and control information. Here the data becomes Segment PDU and is sent to the Network layer.
Network	Packet PDU	The Network layer formats the Segment PDU into Packet PDU that includes logical address and data routing information. The Packet PDU is then sent to the Data Link layer.
Data Link	Frame PDU	The Data Link layer converts the Segment PDU into Frame PDU which contains MAC address and network topology information, then sends the Frame PDU to the Physical layer.
Physical	Bits PDU	The Physical layer converts the Frame PDU into Bits PDU and transfers this data across the transmission media.

Address Resolution Protocol (ARP)

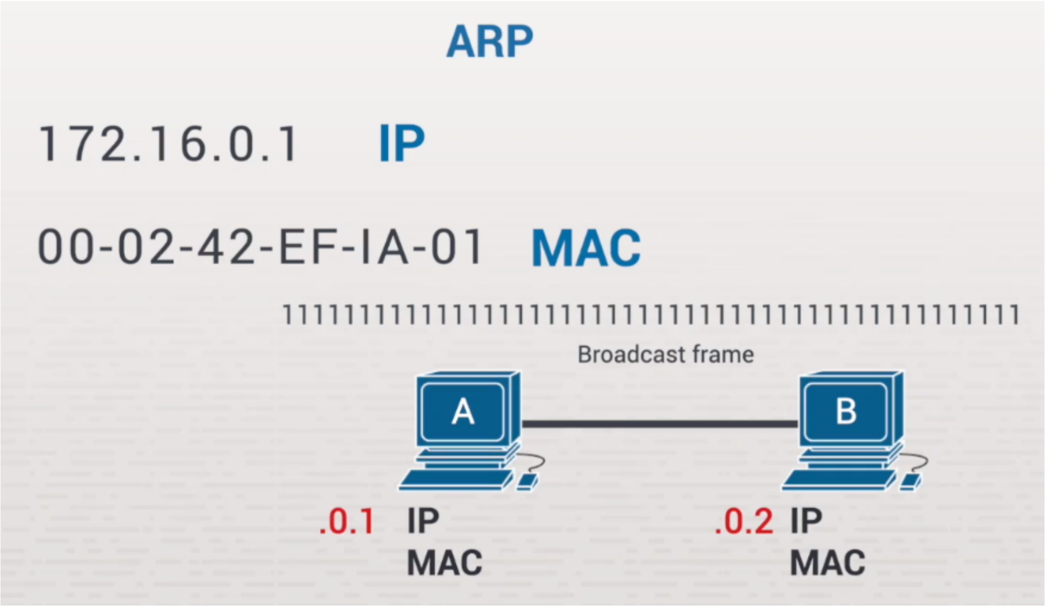
- **IP Addresses** are a Layer 3 Address (network layer) and are 4 bytes of information
- **MAC Addresses** are 6 bytes of information expressed in hexadecimal notation (both letters and numbers)
- **The IP and MAC addresses** identify the sender from both a card perspective and network (or routing) perspective.

ARP

ARP's purpose is to enable station A to dynamically discover the MAC address of Station B. If Station A has never talked to Station B, it won't know the MAC address. To discover the MAC Address, ARP sends out a broadcast frame.

Broadcast Frame: A broadcast frame is a piece of data intended for all recipients. It's a special MAC address built into the frame. It's made up of all binary 1s. Each byte corresponds to 8 bits. 6 times 8 is 48, so there are 48 consecutive bits. That's what's going into the destination MAC address field of the frame. When the frame goes out, it'll be received by Station B and then by anyone else that may be on that network. B will pick it up send back its MAC address.

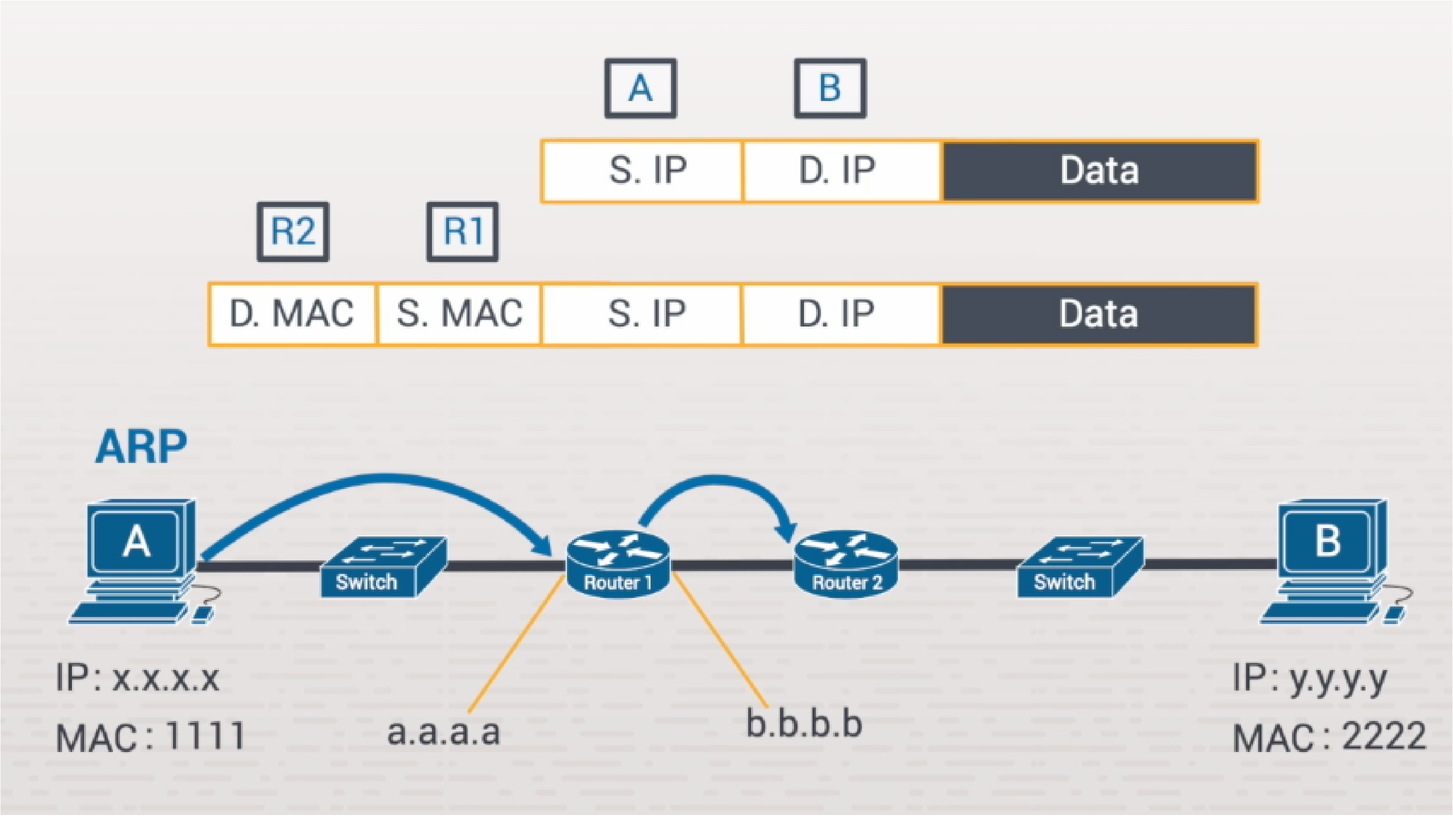
ARP Table: The MAC address is stored on the requesters system in an ARC table. These entries only stay for a certain amount of time (minutes to hours)



Packets and Frames

TCP/IP Model

Headers are changed each time they hop with new destination and source MAC addresses. Packets are reframed for each hop.



Network Communication Process Facts

During IP-based communications, the sending host and the destination host follow a process that uses the OSI model to ensure that data is transferred. The following processes occur when two network hosts use IP-based communications:

1. The data to be transferred is encapsulated on the sending host by moving from the top layer of the TCP/IP or OSI model to the bottom.
2. The data is transmitted on the network medium.
3. If necessary, the data is transferred to various routers, which forward the data to the appropriate network.
4. The data is delivered to the destination host.
5. The data received is de-encapsulated on the destination host by moving from the bottom layer of the TCP/IP or OSI model to the top.

This process is detailed in the following table:

Process Step	Description
Source host encapsulation	The data to be transferred is encapsulated on the sending host from the top layer of the TCP/IP or OSI model to the bottom. The following events occur: 1. The Application layer prepares the data to be sent through the network by encoding it using the appropriate Application layer protocol. 2. The Transport layer receives the stream of data from the Application layer and breaks it into smaller chunks called segments. A Transport layer header, which identifies the source port as well as the destination port, is applied to each segment. Sequencing and control information is also added to the header. 3. The Internet layer converts the segments into packets by adding an Internet layer header, which specifies source and destination IP addresses for each packet. IP addresses are 32-bit (4-byte) logical addresses that can be assigned, unassigned, and reassigned as needed. 4. The Link layer converts the packets into frames by adding a Link layer header, which specifies source and destination MAC addresses for each frame. A MAC address is a 48-bit (6-byte) address that is physically assigned in the firmware of all network interfaces to uniquely identify each interface on the network. MAC addresses are displayed using hexadecimal notation. 5. Each frame is converted into bits and transmitted across the network media.

Network transmission	If necessary, the data is transferred to various routers, which forward the data to the appropriate network. The source and destination network addresses are used to determine whether the hosts reside on the same network or on different networks. If they reside on the same network, the data can be sent directly to the destination host. The Address Resolution Protocol (ARP) is used to determine the MAC address of the host using the destination IP address: 1. The sending host checks its ARP cache to see if it already has an IP-to-MAC address mapping for the destination host. If so, it transmits the frames to the destination host's MAC address. If not, it must use the remaining steps to determine the appropriate MAC address. 2. The sending host sends out an ARP broadcast frame addressed to all MAC addresses on the subnet to ask for the hardware address of the host with the destination IP address. 3. The host with the destination IP address responds to the ARP broadcast with a unicast transmission containing its MAC address. All other hosts ignore the broadcast. 4. The sending host caches the destination host's MAC address in its ARP cache. 5. The source MAC address of the frames is set to the MAC address of the sending system and the destination MAC address is set to the MAC address of the receiving system. 6. The sending host transmits the frames to the destination host's MAC address. If the sending and receiving hosts reside on different networks, the packets must be forwarded from router to router until they reach the appropriate destination network and host. The source IP address of each packet in the transmission is the IP address of the sending system and the destination IP address is the IP address of the receiving system. However, the frames can't be sent directly to the receiving system because it is not on the same network and ARP can only be used on the local subnet. The following occurs in this situation: 1. If it's not already cached, the source system uses ARP to determine the MAC address of the first hop router interface that is connected to the same network segment as the source host (usually the default gateway router). 2. The source MAC address of the frames is set to the MAC address of the sending system, but the destination MAC address is set to the MAC address of the router interface identified with ARP. 3. The frames are transmitted to the first router. 4. The router removes the frame header information and examines the packets in the transmission for the source and

	destination IP addresses. If the destination host is on a network that is directly connected to the router, the router uses ARP to discover its MAC address (if it's not already cached), re-encapsulates the packets in new frames with the destination host's MAC address, and transmits the frames directly to the destination host. If the destination host is not on a directly-connected network, the remaining steps occur. 5. The router uses its routing table to determine the next router the packets should be sent to. 6. The router re-encapsulates the packets in the transmission in new frames. 7. The source MAC address of the frames is set to the MAC address of the local router interface and the destination MAC address is set to the MAC address of the next hop router interface. 8. The router transmits the frames to the MAC address of the next hop router interface. The routing process repeats until the packets arrive at a router that is directly connected to same network as the destination host. 1. The router receives the frames and removes the frame headers. 2. The router examines the packets. 3. It recognizes that the destination host resides on a network that is directly connected to the router. 4. If necessary, the router uses ARP to determine the MAC address of the destination system. 5. The router re-encapsulates the packets in new frames. 6. The source MAC address of the frames is set to the MAC address of the router interface. 7. The destination MAC address is set to the MAC address of the destination host. 8. The frames are transmitted to the destination host.
Destination host de-encapsulation	The data received is de-encapsulated on the destination host by moving from the bottom layer of the TCP/IP or OSI model to the top: 1. The Link layer converts bits received on the network medium into frames and passes them to the Internet layer. 2. The Internet layer extracts the packets from the frames and passes them to the Transport layer. 3. The Transport layer receives packets and uses sequencing and error control information to request retransmission of any missing or damaged packets. 4. The Transport layer uses sequencing information to convert the packets into segments and passes them to the Application layer.

	5. The Application layer uses the appropriate Application layer protocol to convert the segments back into the original data stream from the application on the source host.

NOTE

The OSI model uses the term protocol data unit (PDU) instead of the terms frame, packet and segment.

Presentation and session are layers 5 and 6 of the OSI model respectively and do not correspond to the use of frame, packet, and segment in the TCP/IP model.

IEEE Ethernet standard refers to the standard that defines Ethernet.