PROTOCOL ARCHITECTURE, TCP/IP, AND INTERNET-BASED APPLICATIONS

KEY POINTS

• A protocol architecture is the layered structure of hardware and software that supports the exchange of data between systems and supports distributed applications, such as electronic mail and file transfer.

• At each layer of a protocol architecture, one or more common protocols are implemented in communicating systems. Each protocol provides a set of rules for the exchange of data between systems.

• The most widely used protocol architecture is the TCP/IP protocol suite, which consists of the following layers: physical, network access, internet, transport, and application.

• Another important protocol architecture is the seven-layer OSI model.

THE NEED FOR A PROTOCOL ARCHITECTURE

When computers, terminals, and/or other data processing devices exchange data, the procedures involved can be quite complex.

Consider, for example, the transfer of a file between two computers. There must be a data path between the two computers, either directly or via a communication network. But more is needed. Typical tasks to be performed are as follow:

1. The source system must either activate the direct data communication path or inform the communication network of the identity of the desired destination system.

2. The source system must ascertain that the destination system is prepared to receive data.

3. The file transfer application on the source system must ascertain that the file management program on the destination system is prepared to accept and store the file for this particular user.

4. If the file formats used on the two systems are different, one or the other system must perform a format translation function.

It is clear that there must be a high degree of cooperation between the two computer systems. Instead of implementing the logic for this as a single module, the task is broken up into subtasks, each of which is implemented separately.

In a protocol architecture, the modules are arranged in a vertical stack.
Each layer in the stack performs a related subset of the functions required to communicate with another system. It relies on the next lower layer to perform more primitive functions and to conceal the details of those functions. It provides services to the next higher layer.

Ideally, layers should be defined so that changes in one layer do not require changes in other layers. Of course, it takes two to communicate, so the same set of layered functions must exist in two systems.

Communication is achieved by having the corresponding, or peer, layers in two systems communicate.

The peer layers communicate by means of formatted blocks of data that obey a set of rules or conventions known as a protocol.

The key features of a protocol are as follows:

- **Syntax:** Concerns the format of the data blocks
- **Semantics:** Includes control information for coordination and error handling
- **Timing:** Includes speed matching and sequencing

THE OSI MODEL

The Open Systems Interconnection (OSI) reference model was developed by the International Organization for Standardization (ISO) as a model for a computer protocol architecture and as a framework for developing protocol standards.

The OSI model consists of seven layers:

- Application
- Presentation
- Session
- Transport
- Network
- Data link
- Physical

The intent of the OSI model is that protocols be developed to perform the functions of each layer.

Physical Layer :
The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium.

Representation of bits. The physical layer data consists of a stream of bits (sequence of Os or 1s) with no interpretation. To be transmitted, bits must be encoded into signals--electrical or optical. The physical layer defines the type of encoding (how Os and Is are changed to signals).

o Data rate. The transmission rate-the number of bits sent each second-is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.

o Synchronization of bits. The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.

o Line configuration. The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.

o Physical topology. The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topology (this is a combination of two or more topologies).

o Transmission mode. The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In simplex mode, only one device can send; the other can only receive. The simplex mode is a one-way communication. In the half-duplex mode, two devices can send and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can send and receive at the same time

Data Link Layer :

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer). Figure 2.6 shows the relationship of the data link layer to the network and physical layers.

The data link layer is responsible for moving frames from one hop (node) to the next.

Other responsibilities of the data link layer include the following:

[I Framing. The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

o Physical addressing. If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.

D Flow control. If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

o Error control. The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

D Access control. When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Network Layer:

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.

Other responsibilities of the network layer include the following:

o Logical addressing. The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver.

o Routing. When independent networks or links are connected to create internetworks (network of networks) or a large network, the connecting devices (called router or switches) route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism.

Transport Layer

The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does.

The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level.

Other responsibilities of the transport layer include the following:

o Service-point addressing. Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a service-point address (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.

o Segmentation and reassembly. A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.

o Connection control. The transport layer can be either connectionless or connectionoriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connectionoriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.

o Flow control. Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.

o Error control. Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-toprocess rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

Session Layer

The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The session layer is the network dialog controller. It establishes, maintains, and synchronizes the interaction among communicating systems. The session layer is responsible for dialog control and synchronization. Specific responsibilities of the session layer include the following:

o Dialog control. The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either halfduplex (one way at a time) or full-duplex (two ways at a time) mode.

o Synchronization. The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. Figure 2.12 illustrates the relationship of the session layer to the transport and presentation layers.

Presentation Layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. Figure 2.13 shows the relationship between the presentation layer and the application and session layers.

Specific responsibilities of the presentation layer include the following:

o Translation. The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.

o Encryption. To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.

o Compression. Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

Application Layer

The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.

Figure 2.14 shows the relationship of the application layer to the user and the presentation layer. Of the many application services available, the figure shows only three: XAOO (message-handling services), X.500 (directory services), and file transfer, access, and management (FTAM). The user in this example employs XAOO to send an e-mail message.

Specific services provided by the application layer include the following:

o Network virtual terminal. A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.

o File transfer, access, and management. This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.

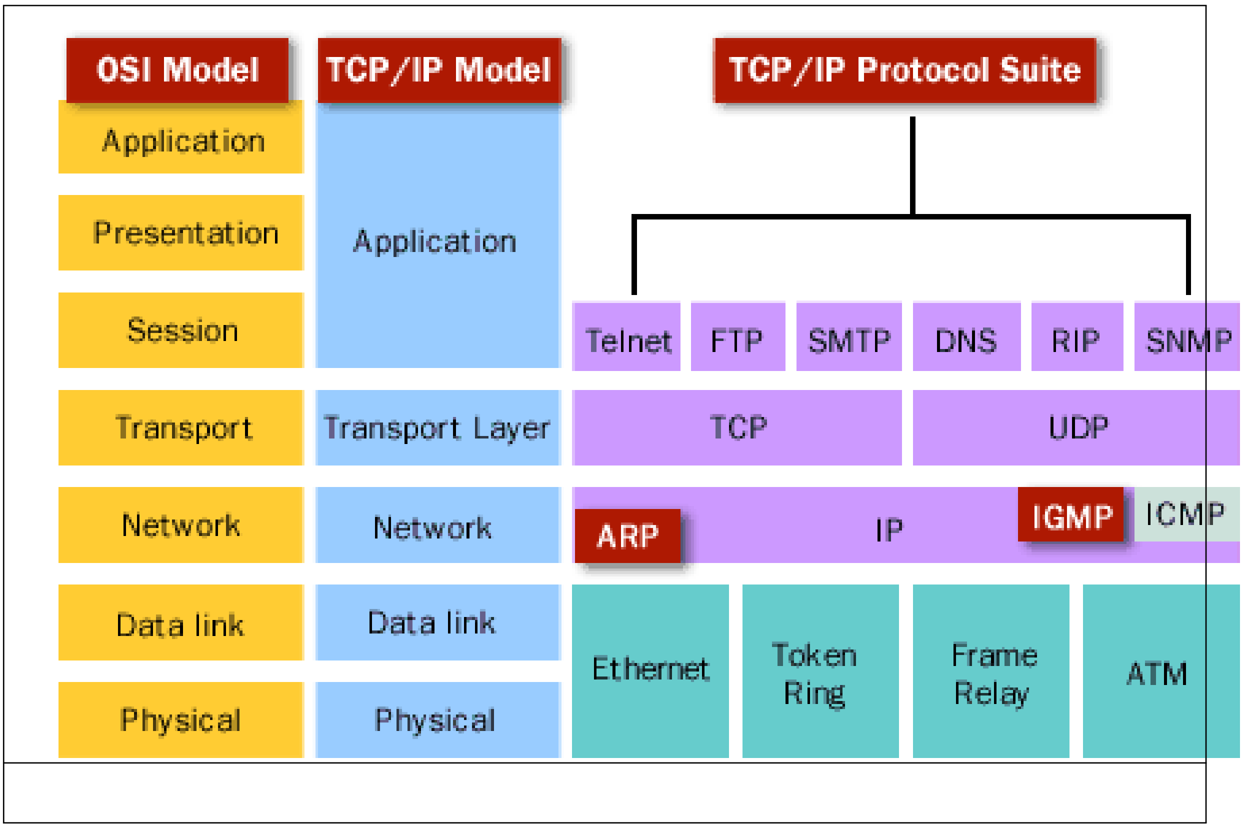
o Mail services. This application provides the basis for e-mail forwarding and storage.

o Directory services. This application provides distributed database sources and access for global information about various objects and services.

OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example		Central Device/ Protocols	
Application (7) Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP	
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT	
Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names	
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP	
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP	
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card —— Switch —— NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch** **Bridge** **WAP** PPP/SLIP	Land Based Layers
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**	

	LAYER #	LAYERS	DATA UNIT	PROTOCOLS
HOST LAYERS	7	**Application** Network Process to Application	Data	HTTP, WebSocket, etc..
	6	**Presentation** Data Representation and Encryption	Data	ACSE, FTAM
	5	**Session** Interhost Communication	Data	L2TP, SMPP
	4	**Transport** End-to-End Connections and Reliability	Segments	TCP / UDP
MEDIA LAYERS	3	**Network** Path Determination and Logical Addressing (IP)	Packets	IP
	2	**Data Link** Physical Addressing (MAC and LLC)	Frames	Ethernet, Wi-Fi
	1	**Physical** Media, Signal, and Binary Transmission	Bits	10 Base T, 802.11



THE TCP/IP PROTOCOL ARCHITECTURE

The TCP/IP protocol architecture is a result of protocol research and development conducted on the experimental packet-switched network, ARPANET, funded by the Defense Advanced Research Projects Agency (DARPA), and is generally referred to as the TCP/IP protocol suite.

This protocol suite consists of a large collection of protocols that have been issued as Internet standards by the Internet Activities Board (IAB).

The TCP/IP Layers:

In general terms, communications can be said to involve three agents: applications, computers, and networks.

Examples of applications include file transfer and electronic mail. The applications that we are concerned with here are distributed applications that involve the exchange of data between two computer systems.

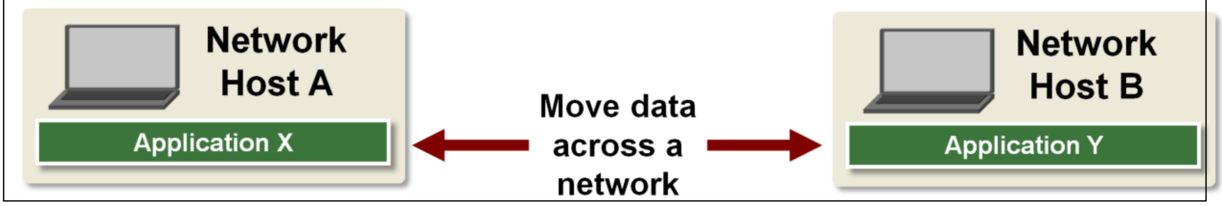
These applications, and others, execute on computers that can often support multiple simultaneous applications.

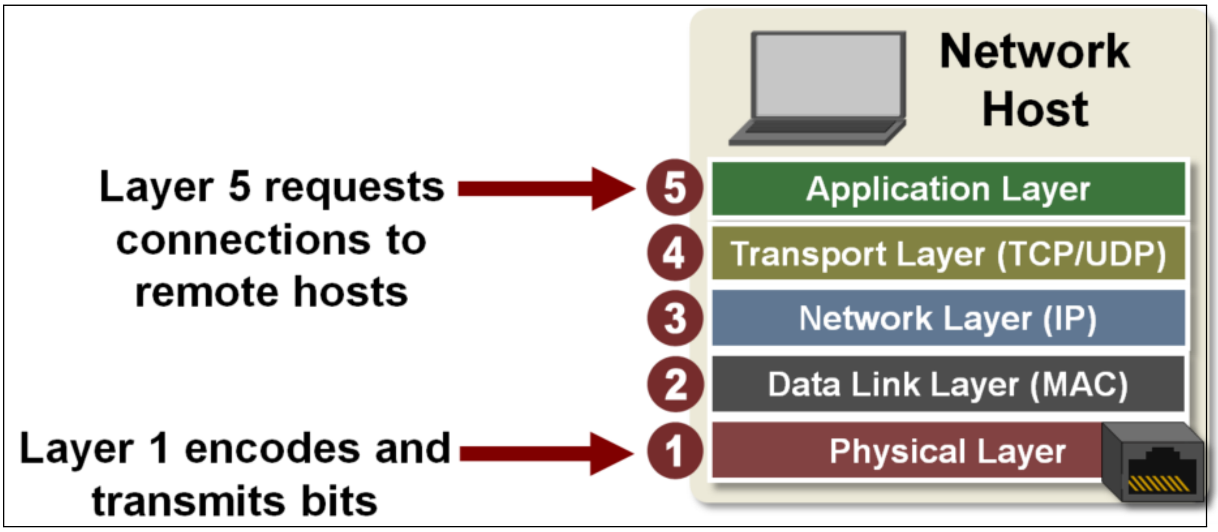
Computers are connected to networks, and the data to be exchanged are transferred by the network from one computer to another. Thus, the transfer of data from one application to another involves first getting the data to the computer in which the application resides and then getting the data to the intended application within the computer.

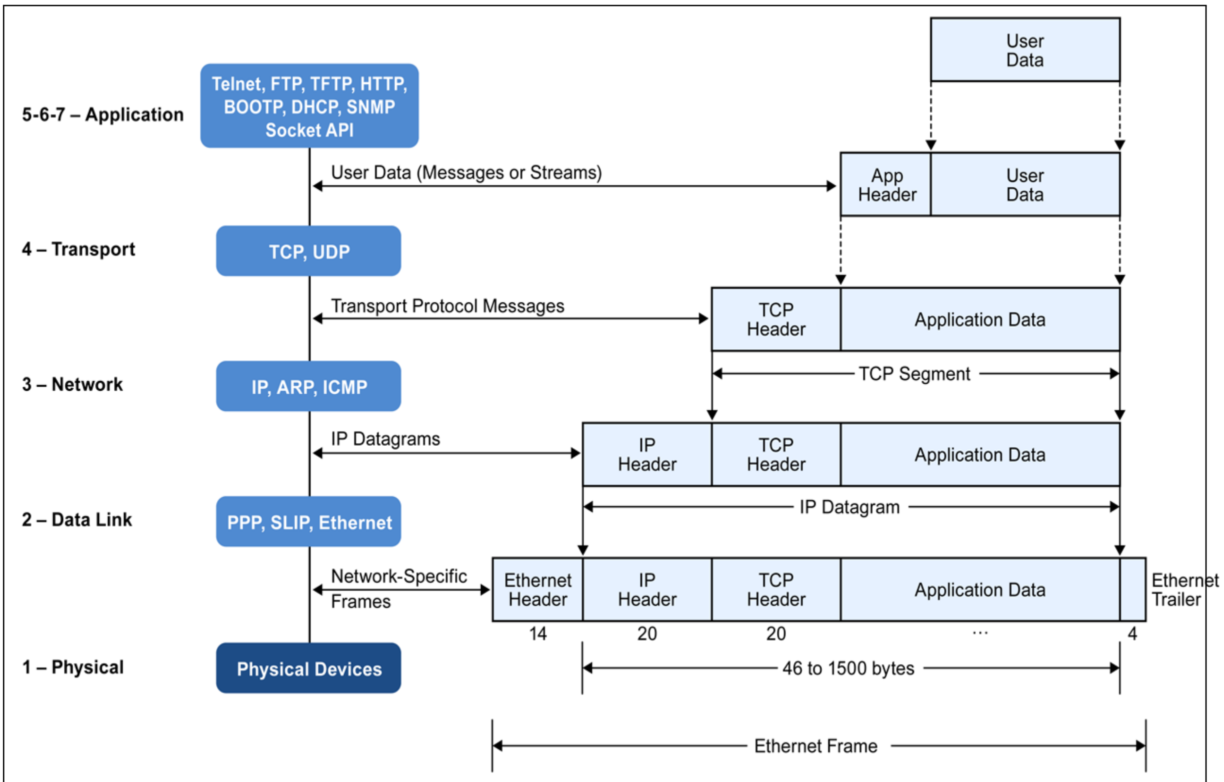
With these concepts in mind, we can organize the communication task into five relatively independent layers.

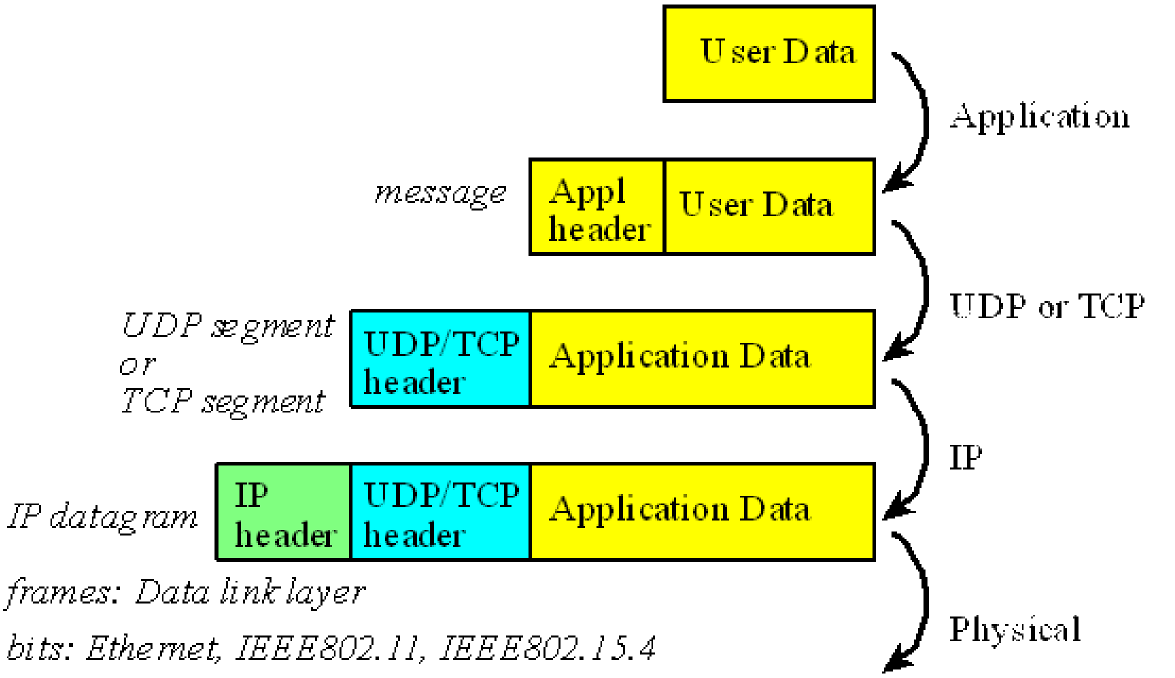
- Physical layer
- Network access layer
- Internet layer
- Host-to-host, or transport layer
- Application layer

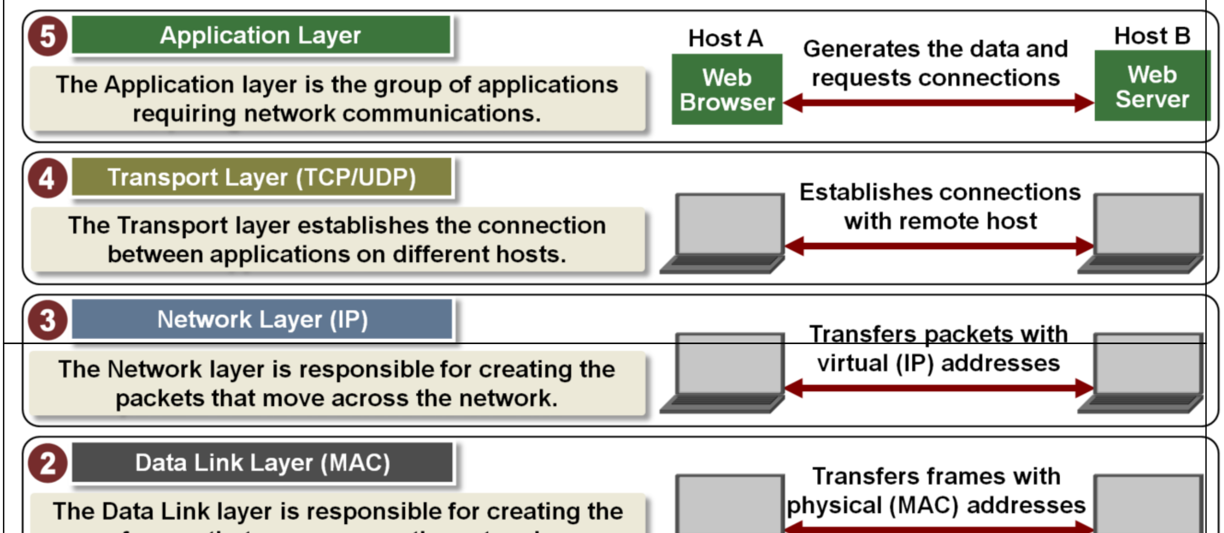
BASIC NEEDS FOR TCP/IP COMMUNICATION:

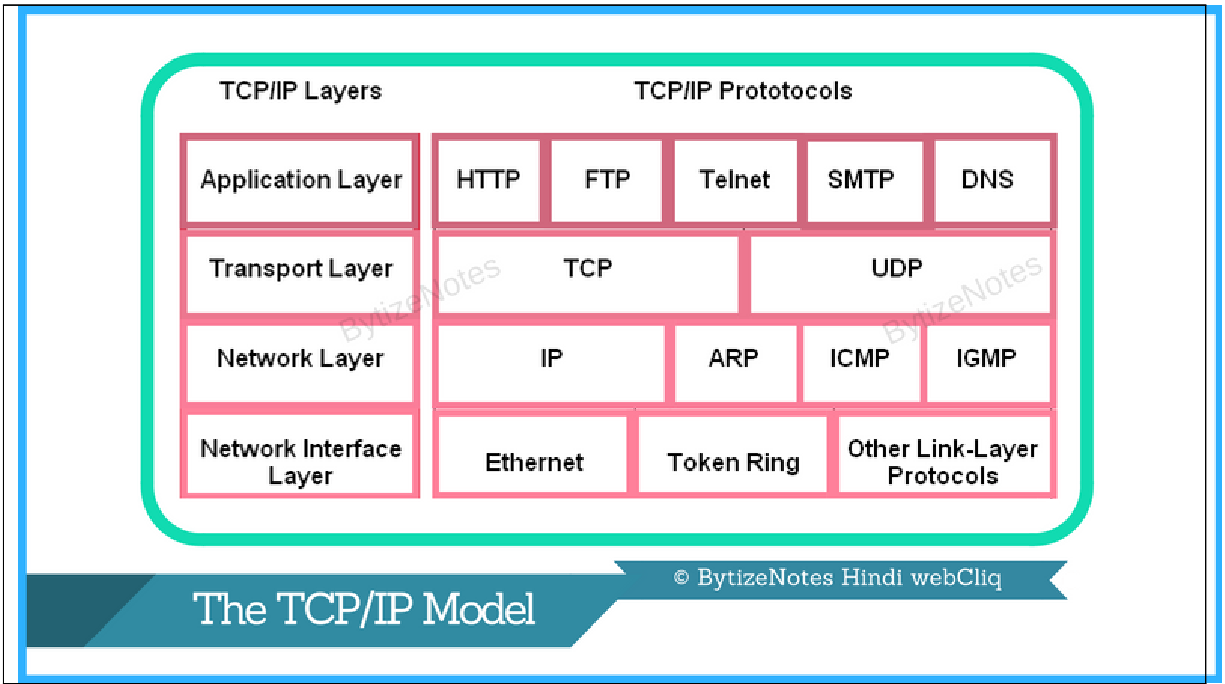


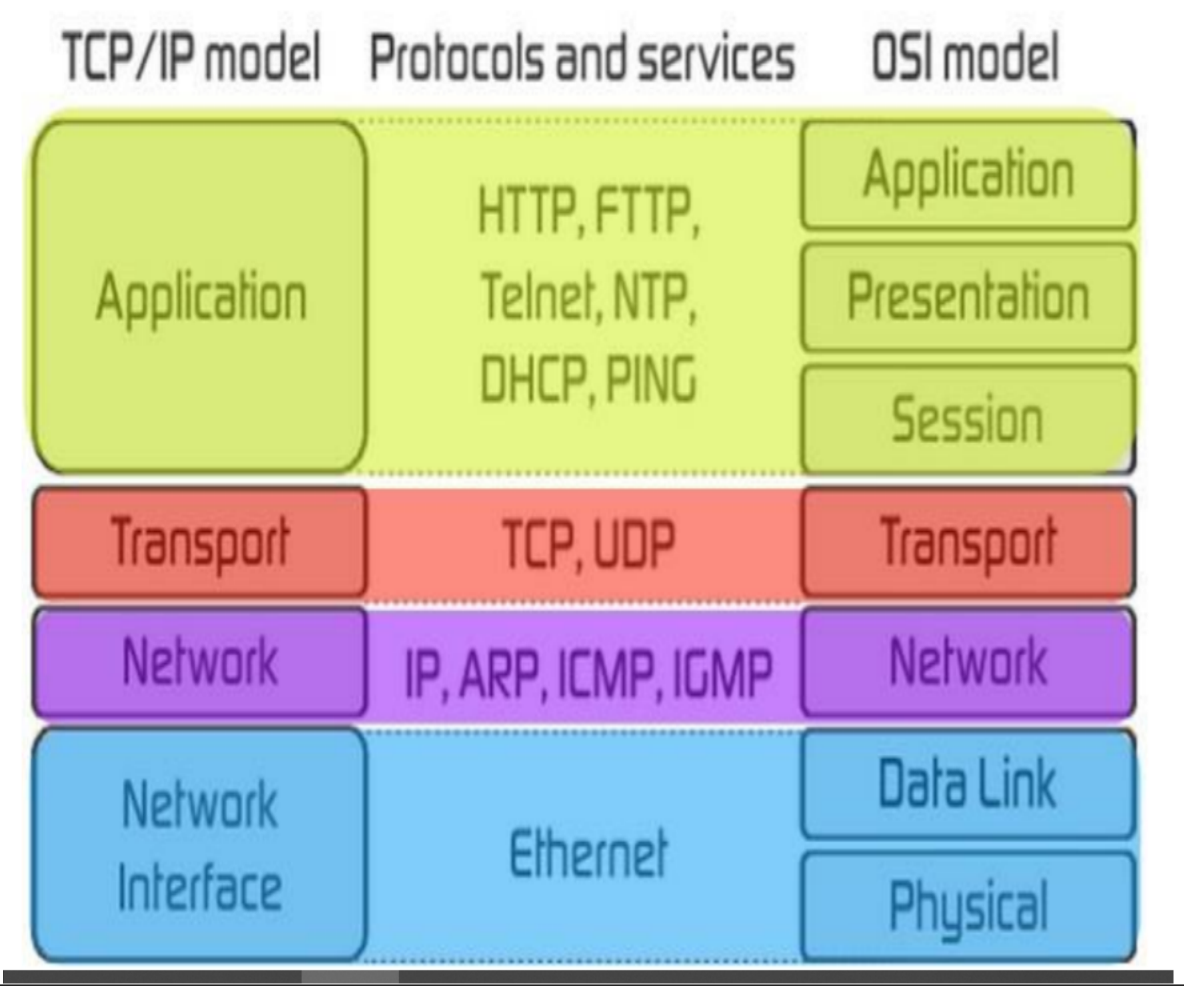


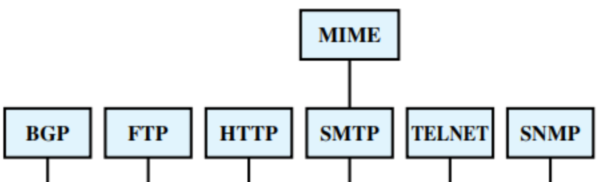

5-6-7 – Application
Telnet, FTP, TFTP, HTTP, BOOTP, DHCP, SNMP Socket API
4 – Transport
TCP, UDP
3 – Network
IP, ARP, ICMP
2 – Data Link
PPP, SLIP, Ethernet
1 – Physical
Physical Devices
User Data (Messages or Streams)
Transport Protocol Messages
IP Datagrams
Network-Specific Frames
User Data
App Header
User Data
TCP Header
Application Data
TCP Segment
IP Header
TCP Header
Application Data
IP Datagram
Ethernet Header
IP Header
TCP Header
Application Data
Ethernet Trailer
14
20
20
...
4
46 to 1500 bytes
Ethernet Frame


User Data
Application
message
Appl header
User Data
UDP or TCP
UDP segment or TCP segment
UDP/TCP header
Application Data
IP
IP datagram
IP header
UDP/TCP header
Application Data
frames: Data link layer
bits: Ethernet, IEEE802.11, IEEE802.15.4
Physical


5
Application Layer
The Application layer is the group of applications requiring network communications.
Host A
Web Browser
Generates the data and requests connections
Host B
Web Server
4
Transport Layer (TCP/UDP)
The Transport layer establishes the connection between applications on different hosts.
Establishes connections with remote host
3
Network Layer (IP)
The Network layer is responsible for creating the packets that move across the network.
Transfers packets with virtual (IP) addresses
2
Data Link Layer (MAC)
The Data Link layer is responsible for creating the
Transfers frames with physical (MAC) addresses


TCP/IP Layers
TCP/IP Prototocols
Application Layer
HTTP
FTP
Telnet
SMTP
DNS
Transport Layer
TCP
UDP
Network Layer
IP
ARP
ICMP
IGMP
Network Interface Layer
Ethernet
Token Ring
Other Link-Layer Protocols
© BytizeNotes Hindi webCliq
The TCP/IP Model


TCP/IP model
Protocols and services
OSI model
Application
HTTP, FTTP, Telnet, NTP, DHCP, PING
Application
Presentation
Session
Transport
TCP, UDP
Transport
Network
IP, ARP, ICMP, IGMP
Network
Network Interface
Ethernet
Data Link
Physical


MIME
BGP
FTP
HTTP
SMTP
TELNET
SNMP