Chapter 5

TCP/IP SUITE

Objectives:-

- TCP/ IP Model Concept.
- Defining/functioning of different Layers of TCP / IP suite.

5.1 Introduction –Addressing mechanism in the Internet

- An IP address is an address used in order to uniquely identify a device on an IP network.
- The address is made up of 32 binary bits, which can be divisible into a network portion and host portion with the help of a subnet mask.
- The 32 binary bits are broken into four octets (1 octet = 8 bits). Each octet is converted to decimal and separated by a period (dot).
- For this reason, an IP address is said to be expressed in dotted decimal format (for example, 172.16.81.100).
- The value in each octet ranges from 0 to 255 decimal, or 00000000 − 11111111 binary.
- The Internet Assigned Numbers Authority (IANA) assigns network identifiers to avoid duplications.

5.2 IP Addressing – IP Address classes, classless IP addressing, Subnetting, supernetting, Masking.

- IPv4 addresses are unique.
- They are unique in the sense that each address defines one, and only one, connection to the Internet.
- Two devices on the Internet can never have the same address at the same time.

Address Space:

- IPv4 uses 32-bit addresses, which means that the address space is $\mathbf{2^{32}}$ or 4,294,967,296 (more than 4 billion).
- This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet.

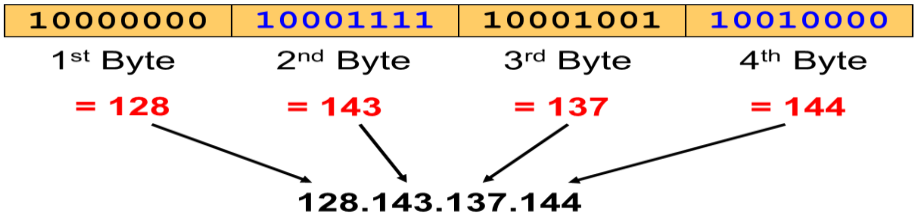
Notations

There are two prevalent notations to show an IPv4 address: Binary notation and Dotted decimal notation.

- ***Binary Notation:***
  - In binary notation, the IPv4 address is displayed as 32 bits.
  - Each octet is often referred to as a byte.

- So it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address.
- The following is an example of an IPv4 address in binary notation: **01110101 10010101 00011101 00000010**

- ***Dotted-Decimal Notation:***
  - To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes.
  - Each byte is identified by a decimal number in the range [0..255].
  - The following is the dotted decimal notation of the above address: **117.149.29.2**
  - **Example:**



Example: Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111

Solution:
We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

a. 129.11.11.239
b. 193.131.27.255

IP Address classes

- IPv4 addressing, at its inception, used the concept of classes.
- This architecture is called classful addressing.
- In classful addressing, the address space is divided into five classes: A, B, C, D, and E.
- Each class occupies some part of the address space.
- We can find the class of an address when given the address in binary notation or dotted-decimal notation.
- If the address is given in binary notation, the **first few bits** can immediately tell us the class of the address.

	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

Fig: Finding the class in binary notation.

- If the address is given in decimal-dotted notation, the **first byte defines** the class.

	First byte	Second byte	Third byte	Fourth byte
Class A	0 to 127			
Class B	128 to 191			
Class C	192 to 223			
Class D	224 to 239			
Class E	240 to 255			

Fig: Finding the class in decimal notation

Example*:* Find the class of each address.

a. 00000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 14.23.120.8
d. 252.5.15.111

Solution:

a. The first bit is O. This is a **class A** address.
b. The first 2 bits are 1; the third bit is O. This is a class C address.
c. The first byte is 14 (between 0 and 127); the class is A.
d. The first byte is 252 (between 240 and 255); the class is E.

Classes and Blocks

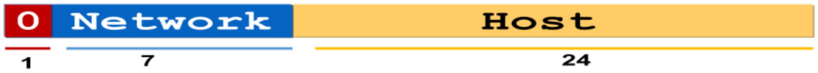
One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size.

Class	*Number ofBlocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Fig: Number of blocks and block size in classful IPv4 addressing.

Class A:

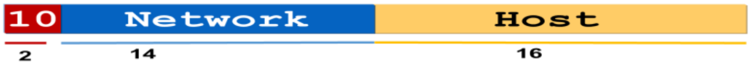
- The high-order (First) bit in a class-A address is always set **to zero**.
- The **next seven bits** complete the network ID.
- The remaining 24 bits represent the host ID.
- This allows for **128 networks** and **16,777,214 hosts per network**.
- In this **7 bits are used for network field** and **24 bits for host field**.
- Class A IP address range includes **1.0.0.0 to 127.255.255.255**



Note: Millions of class A addresses are wasted.

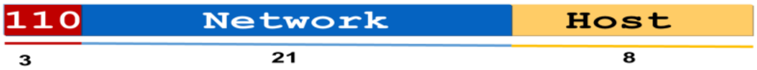
Class B:

- *Class B* addresses are assigned to medium-sized to large-sized networks.
- The two high-order bits in a class B address are always set to **binary 1 0**.
- The next 14 bits complete the network ID.
- The remaining 16 bits represent the host ID.
- This allows for **16,384 networks** and **65,534 hosts** per network.
- Class B IP address range includes **128.0.0.0 to 191.255.255.255**



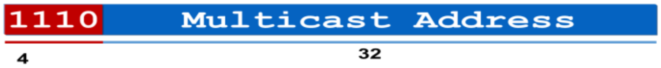
Class C:

- *Class C* addresses are used for small organizations with a small number of attached hosts or routers.
- The three high-order bits in a class C address are always set to **binary 1 1 0.**
- The next 21 bits complete the network ID.
- The remaining 8 bits (last octet) represent the host ID.
- This allows for **2097152 networks** and **256 hosts** per network.
- Class C IP address range includes **192.0.0.0 to 223.255.255.255.**



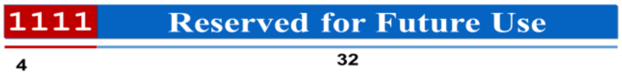
Class D:

- *Class D* addresses are reserved for IP multicast addresses.
- The four high-order bits in a class D address are always set to **binary 1 1 1 0.**
- The remaining bits recognize hosts.
- Class D IP address range includes **224.0.0.0 to 239.255.255.255**



Class E:

- *Class E* is an experimental address that is reserved for future use.
- The high-order bits in a class E address are set to **binary 1111**.
- Class E IP address range includes **240.0.0.0 to 255.255.255.255**



Netid and Hostid

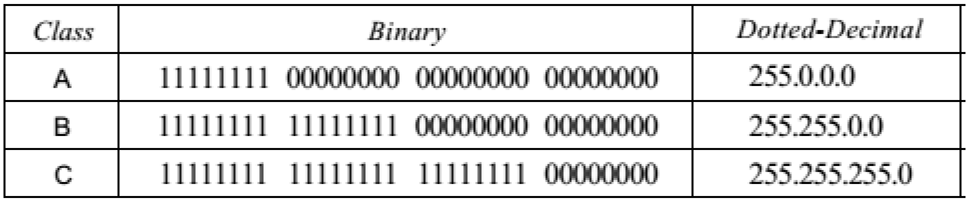
- In classful addressing, an IP address in class A, B, or C is divided into netid and hostid.
- These parts are of varying lengths, depending on the class of the address.

network prefix
host number

- Note that the concept does not apply to classes D and E.
- In **class A, one byte** defines the netid and three bytes define the hostid.
- In **class B, two bytes** define the netid and two bytes define the hostid.
- In **class C, three bytes** define the netid and one byte defines the hostid.

Mask

- Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask/natural masks), a 32-bit number made of contiguous 1's followed by contiguous 0's.
- The masks for classes A, B, and C are shown in Table.
- The concept does not apply to classes D and E.



Class	*Binary*	*Dotted-Decimal*
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

- The mask can help us to find the netid and the hostid.
- For example, the mask for a class-A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.

Subnetting

- If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets) or, in rare cases, share part of the addresses with neighbors.
- Subnetting increases the number of 1's in the mask.
- To create multiple logical networks that exist within a single Class A, B, or C network.
- If you do not subnet, you are only able to use one network from your Class A, B, or C network, which is unrealistic.

- The subnet mask follows two rules:
  - If a **binary bit is set to a 1** (**or *on***) in a subnet mask, the corresponding bit in the address **identifies the network.**
  - If a binary bit is **set to a 0** (or *off*) in a subnet mask, the corresponding bit in the address **identifies the host.**

Finding The Subnet Address: We use binary notation for both the address and the mask and then apply the AND operation to find the subnet address.

Example: What is the subnetwork address if the destination address is 200.45.34.56 and the subnet mask is 255.255.240.0?

Solution :

Step 1: Convert given IP and Subnet mask to Binary

Step 2: Perform AND Operation on these two.

11001000 00101101 00100010 00111000 Binary 200.45.34.56

11111111 11111111 11110000 00000000 Subnet Mask 255.255.255.0

11001000 00101101 0010**0000 00000000**

The subnetwork address is **200.45.32.0**.

Step 3: Convert the result of AND operation to Dotted Decimal format which is Subnet mask.

Example 2: (VIMP)

A company is granted the site address **201.70.64.0** (class C). The company needs **six subnets.** Design the subnets.

Solution:

- The number of 1s in the default mask is 24 (class C).
- The company needs six subnets.
- This number 6 is not a power of 2.
- The next number that is **a power of 2 is 8** (**2^3**).
- We need 3 more 1's in the subnet mask.
- The total number of 1's in the subnet mask is 27 (24 + 3).
- The total number of 0's is 5 (32 - 27).
- The mask is

11111111 11111111 11111111 11100000

or

255.255.255.224

- The number of subnets is 8.
- The number of addresses in each subnet is **2^5** (5 is the number of 0s) or 32.

- **Subnet 1:**
  The bit combination is **001**.
  Taking last octet in binary:**0 0 1** 0 0 0 0 0 = 32 (10)
  Hence the subnet address is, 201.70.64. **32**
- **Subnet 2:**
  The bit combination is **01 0**.
  Taking last octet in binary:**0 0 1** 0 0 0 0 0 = 64(10)
  Hence the subnet address is, 201.70.64. **64**
- **Subnet 3:**
  The bit combination is **011**.
  Taking last octet in binary:**0 1 1** 0 0 0 0 0 = 96(10)
  Hence the subnet address is, 201.70.64. **96**
- **Subnet 4:**
  The bit combination is **100**.
  Taking last octet in binary :**1 0 0** 0 0 0 0 0 = 128(10)
  Hence the subnet address is, 201.70.64. **128**
- **Subnet 5:**
  The bit combination is **101**.
  Taking last octet in binary :**1 0 1** 0 0 0 0 0 = 160(10)
  Hence the subnet address is, 201.70.64. **160**
- **Subnet 6:**
  The bit combination is **110**.
  Taking last octet in binary :**1 1 0** 0 0 0 0 0 = 192 (10)
  Hence the subnet address is, 201.70.64. **192**

Example 3:

A company is granted the site address **181.56.0.0** (class B). The company needs 1000 subnets. Design the subnets.

Solution:

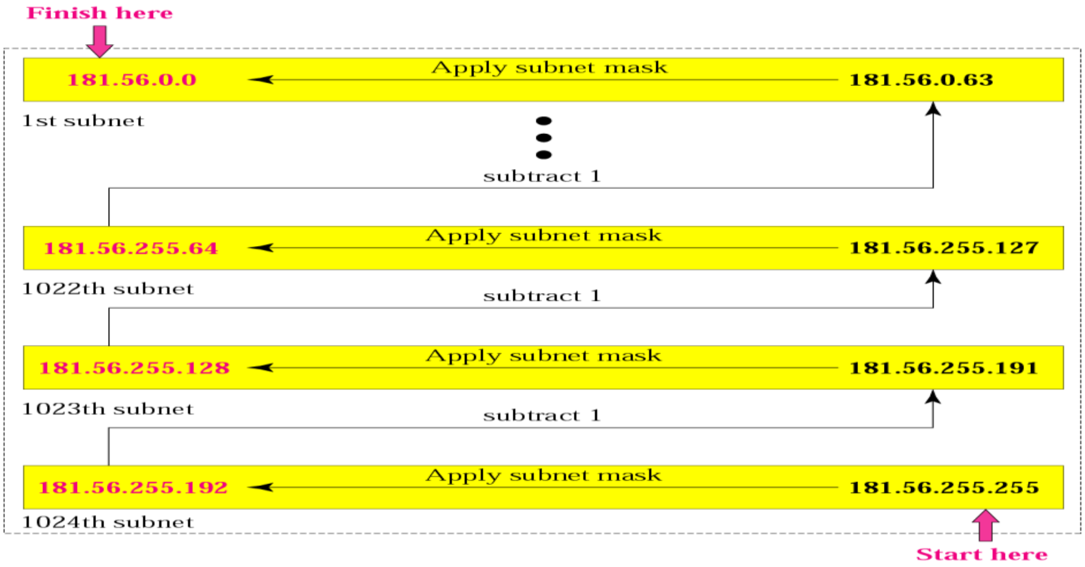
- The number of 1s in the default mask is 16 (class B).
- The company needs 1000 subnets.
- This number is not a power of 2.
- The next number that is a power of 2 is 1024 (2^{10}).
- We need 10 more 1's in the subnet mask.
- The total number of 1's in the subnet mask is 26 (16 + 10).
- The total number of 0's is 6 (32 - 26).
- **The mask is**

11111111 11111111 11111111 11000000

or

255.255.255.192

- The number of subnets is 1024.
- The number of addresses in each subnet is 2^6 (6 is the number of 0s) or 64.



Supernetting

- The most of the class A and class B addresses were exhausted; however, there was still a huge demand for midsize blocks.
- The size of a class C block with a maximum number of 256 addresses did not satisfy the needs of most organizations.
- One solution was supernetting.
- In supernetting, an organization can combine several class C blocks to create a larger range of addresses.
- In other words, several networks are combined to create a supernetwork or a supernet.
- An organization can apply for a set of class C blocks instead of just one.
- For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks.
- The organization can then use these addresses to create one supernetwork.
- Supernetting decreases the number of 1's in the mask.
- For example,
  - if an organization is given four class C addresses, the mask changes from 24 to 22.

Example:

- We need to make a supernetwork out of 16 class C blocks. What is the supernet mask?

Solution:

- We need 16 blocks.
- For 16 blocks we need to change four 1s to 0s in the default mask. So the mask is

11111111 11111111 1111**0000** 00000000

Or

255.255.240.0

Address Depletion

- The flaws in classful addressing scheme combined with the fast growth of the Internet lead to the near depletion of the available addresses.
- Yet the number of devices on the Internet is much less than the 2^{32} address space.
- We have run out of class A and B addresses, and a class C block is too small for most midsize organizations.
- One solution that has alleviated the problem is the idea of classless addressing.

Classless Addressing

- To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented.
- In this scheme, there are no classes, but the addresses are still granted in blocks.

Address Blocks

- In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses.
- The size of the block (the number of addresses) varies based on the nature and size of the entity.
- For example, a household may be given only two addresses; a large organization may be given thousands of addresses.
- An ISP, as the Internet service provider, may be given thousands of addresses based on the number of customers it may serve.

Restriction: To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:

1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (1, 2, 4, 8, ...).
3. The first address must be evenly divisible by the number of addresses.

Example:

A company needs 600 addresses. Which of the following set of class C blocks can be used to form a supernet for this company?

1. 198.47.32.0 198.47.33.0 198.47.34.0
2. 198.47.32.0 198.47.42.0 198.47.52.0 198.47.62.0
3. 198.47.31.0 198.47.32.0 198.47.33.0 198.47.52.0
4. 198.47.32.0 198.47.33.0 198.47.34.0 198.47.35.0

Solution:

1: No, there are only three blocks.
2: No, the blocks are not contiguous.
3: No, 31 in the first block is not divisible by 4.
4: **Yes, all three requirements are fulfilled.**

5.3 Layered Structure of the TCP / IP Model – Host-to-Network, Internet, Transport, Application Layer.

- The TCP/IP protocol suite was developed prior to the OSI model.
- Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model.
- The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application.

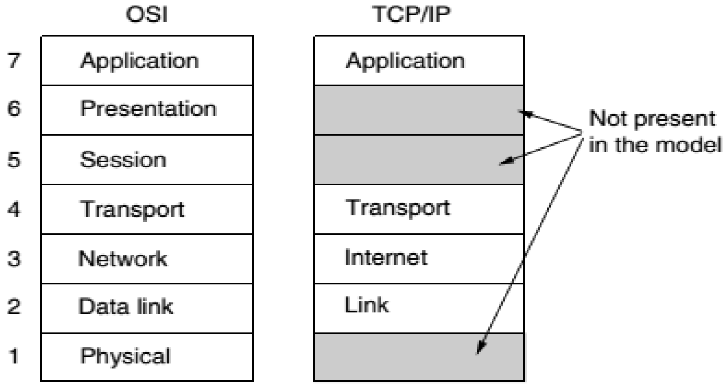


Fig: TCP/ IP Model

- When TCP/IP is compared to OSI,
- The **host-to-network layer** is equivalent to the **combination of the physical and data link layers**.
- The **internet layer** is equivalent to the **network layer**, and the **application layer doing the job of the session, presentation, and application** layers.

Network Access Layer

- Also called as Host-to-Network Layer.
- Performs all functions of physical Layer and Data Link Layer.
- Exchange of data between end system and network.
- Address of host and destination
- Prioritization of transmission.
- This deals with hardware level, connections as in other network model.
- TCP/IP Protocol Suite includes Host-to-Network Layer protocols such as-
- Serial Line Internet Protocol (SLIP) And Point to Point Protocol (PPP).

TCP/IP Internet Layer

- An Internet is an interconnection of two or more networks.
- Internet layer handles tasks similar to network access layer, but between networks rather than between nodes on a network.
- Uses IP for addressing and routing across networks.
- Implemented in workstations *and* routers.

- This layer is concerned with the format of datagrams as defined in the internet protocol(IP).
- The protocols in this layer include Address Resolution Protocol (ARP),
- Reverse Address Resolution Protocol (RARP) and
- Internet Control Message Protocol (ICMP).

TCP/IP Transport Layer

- This layer is concerned with the transmission of the data.
- The two main protocols that operate at this layer are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).
- TCP is reliable transmission protocol and it guarantees that the proper data transfer will take place.
- UDP is not designed to be reliable or guarantee data delivery.
- Also called host-to-host layer.
-

Functions of Transport Layer

1. **Service point addressing:** - Delivery is from specific process on computer to specific process on another computer. For this transport layer uses port addresses.
2. **Segmentation and reassemble:** -Each segment of a message contains a sequence number which is used to reassemble the message correctly.
3. **Connection control:**-Logical connection is created between source and destination for the duration of complete message transfer.
4. **Flow control:-**Flow control is performed end to end**.**
5. **Error control**:-Error control is performed process to process. It ensures that entire message arrives at receivers transport layer without error (damage or loss or duplication). Error correction is done by retransmission.

TCP/IP Application Layer

- The application layer is concerned with providing network services to applications.
- There are many application network processes and protocols that work at this layer, including
- Hyper Text Transfer Protocol (HTTP), Simple Mail Transport Protocol (SMTP) and File Transfer Protocol (FTP).

5.4 TCP / IP Protocol Suite:
Host-to-Network-SLIP and PPP,
Internet Layer-ARP, RARP and IP:Introduction, IPv4, IPv6 (Header Format), Difference between IPv4 & IPv6.
Transport Layer- TCP and UDP (Frame Format, port addresses),
Application Layer- FTP, SMTP, DNS.

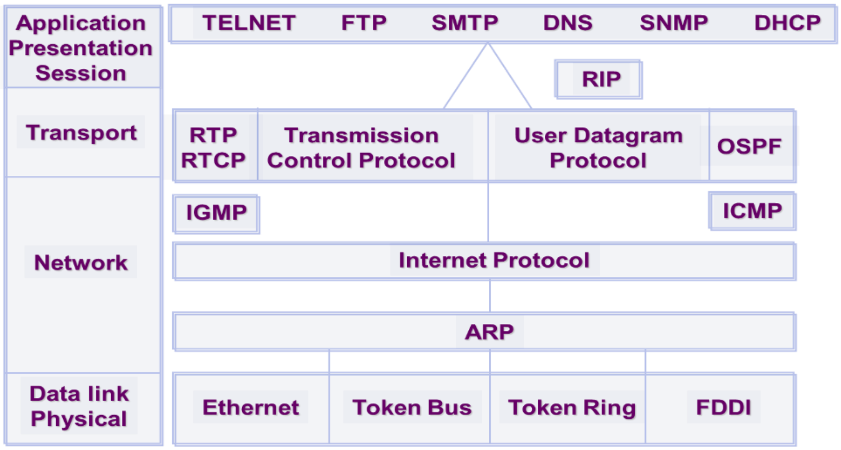


Fig: TCP/ IP Protocol Suite

- **Host-to-Network Layer Protocol**
- Host to network Layer Defines two protocols
    - SLIP
    - PPP
- SLIP and PPP Protocols allow a user to dial into an ISP over Telephone Line.

SLIP (Serial Line Internet Protocol)

- It designed to work over serial ports and modem connections.
- Defines a sequence of bytes that frame IP packets on a serial line.
- Commonly used for point-to-point serial connections running TCP/IP.
- It is designed to transmit signals over a serial connection and has very low overhead.
- Data transmission with SLIP is **very simple**.
- This protocol sends a frame composed **only of data to be sent** followed by an end of transmission character (the **END** character, the ASCII code **192**).
- A SLIP frame looks like:

Data to be Transmitted	END

Problems with SLIP

- SLIP does not perform error detection and correction.
- SLIP does not provide any authentication.
- SLIP is not approved internet standard.
- SLIP supports static IP address assignment.

PPP (Point to Point Protocol)

- It is a much more developed protocol than SLIP(which is why it is replacing it).
- It transfers additional data, better suited to data transmission over the Internet.
- PPP is More Complex than SLIP.
- PPP Protocol Supports a set of Authentication Protocol.

Line Control Protocol (LCP):

- Responsible for establishing, maintaining, and terminating connection.

Password Authentication Protocol(PAP):

- The second is of authentication.
- Password Authentication is used.

Network Control Protocol (NCP):

- After authentication is done, PPP sends NCP packet.
- This packet tells ISP server what kind of traffic is to be passed over PPP link.

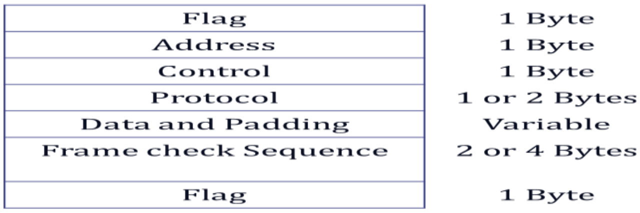
IP Control Protocol (IPCP):

- Finally the IP packets are exchanged.
- IPCP Establishes and terminates the Network layer connection.

PPP Solves the problems of SLIP

- PPP is point to point protocol.
- PPP perform error detection.
- PPP provides authentication and security.
- PPP is approved internet standard.
- PPP supports IP and other protocols.
- PPP supports Dynamic IP address assignment

PPP Frame Format



Flag	1 Byte
Address	1 Byte
Control	1 Byte
Protocol	1 or 2 Bytes
Data and Padding	Variable
Frame check Sequence	2 or 4 Bytes
Flag	1 Byte

SLIP Vs PPP

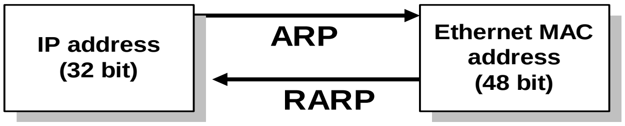
SLIP	**PPP**
Serial Line Internet Protocol does not establish or maintain connection between the client and ISP server.	In PPP, LCP (Line Control Protocol) is responsible for establishing, maintaining and termination connection between two end points.
Communication starts once the connection between two modems are established.	Communication begins only after authentication and the types of traffic is sent by the client.
Type of traffic cannot be selected in SLIP.	Type of traffic can be selected by NCP(Network Control Protocol)
No protocol for termination.	IPCP(IP Control Protocol) terminates a network layer connection between the user and ISP.
No addressing mechanism provided.	Additional services for addressing mechanism is provided
Doesn't allow error control	Allows error control
No provision for data compression	Provides Data compression.

2. Internet Layer Protocols

- The Four Network Layer protocols are:
  - ARP
  - RARP
  - IP
  - ICMP

1. ARP-Address Resolution Protocol

- ARP takes the IP address of a host as input & gives its **corresponding physical address** as the output.
  - The Internet is based on IP addresses
  - Data link protocols (Ethernet, FDDI, ATM) may have different (MAC) addresses
- The ARP and RARP protocols perform the **translation between IP addresses and MAC** layer addresses.

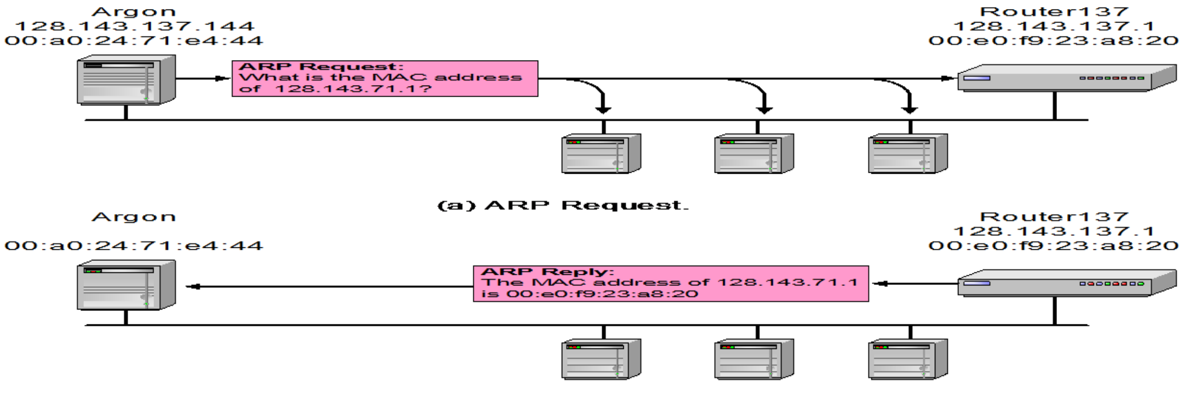


- ARP sends the IP broadcast message to all the computer on the network.
- The computer whose IP address matches the broadcast IP address sends a reply and along with, its physical address to the broadcasting computer.
- All other computers ignore the broadcast message.

Address Translation with ARP

Example:

ARP request: Argon broadcasts an ARP request to all stations on the network: **"What is the hardware address of Router137?"**



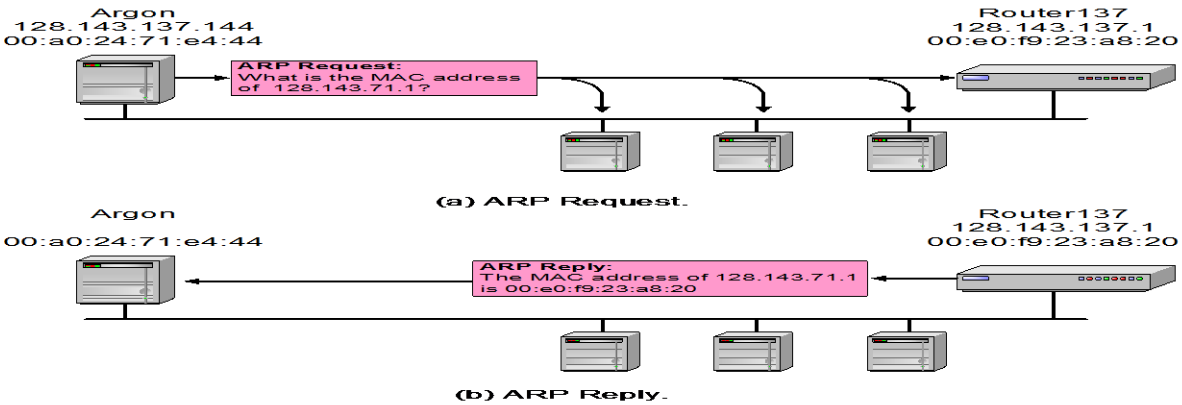
ARP Request:
What is the MAC address
of 128.143.71.1?

(a) ARP Request.

ARP Reply:
The MAC address of 128.143.71.1
is 00:e0:f9:23:a8:20

(b) ARP Reply.

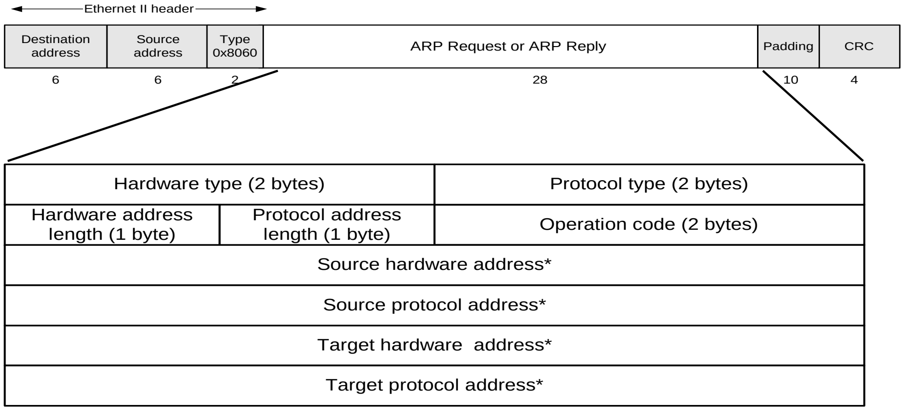
ARP Reply: **Router137** responds with an ARP Reply which contains the hardware address



(a) ARP Request.

(b) ARP Reply.

ARP Packet Format



* Note: The length of the address fields is determined by the corresponding address length fields

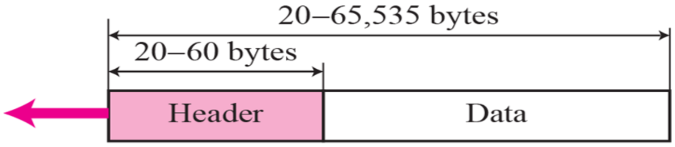
2. RARP (Reverse Address Resolution Protocol)

- If we have to obtain the IP address corresponding to the given Ethernet address.
- RARP works in very similar way of ARP, but in exactly opposite direction.
- The RARP server looks at this request.
- Then it looks up the Ethernet address in its configuration files and sends back the corresponding IP address.

ARP	RARP
ARP converts an Internet Protocol address to its physical network address (MAC).	RARP converts Ethernet MAC address to IP address.
ARP broadcast an IP address in an effort to discover its equivalent hardware address.	RARP broadcast the System's hardware address.
Local host maintain the ARP Table.	A RARP server maintain the RARP table.
RFC 826 describes ARP	RFC 903 describes RARP

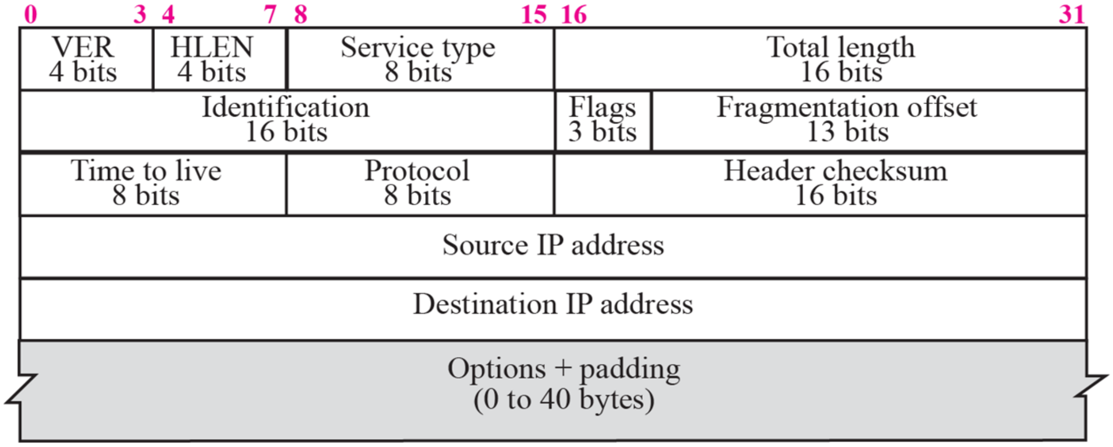
3. Internet Protocol (IP)

- **IP** is internet Protocol.
- It is **unreliable protocol** because it does not provide any **error control and flow control**.
- Packets in IP are called “**Datagram**”.
- Datagram is variable length packet with two parts –header and data



a. IP datagram

IP Header Format



b. Header format

Fields Of IP Header:

- **Version (4 bits)**: current version is 4.
- **Header length (4 bits)**: length of IP header, in multiples of 4 bytes
- **DS/ECN field (1 byte):** This field was previously called as Type-of-Service (TOS) field.
    - Differentiated Service (DS) (6 bits): Used to specify service level (currently not supported in the Internet)
    - Explicit Congestion Notification (ECN) (2 bits):New feedback mechanism used by TCP.
- **Identification (16 bits):** Unique identification of a datagram from a host. Incremented whenever a datagram is transmitted.
- **Flags (3 bits):**
    - First bit always set to 0
    - DF bit (Do not fragment)
    - MF bit (More fragments)

- **Time To Live (TTL) (1 byte):** Specifies longest paths before datagram is dropped.

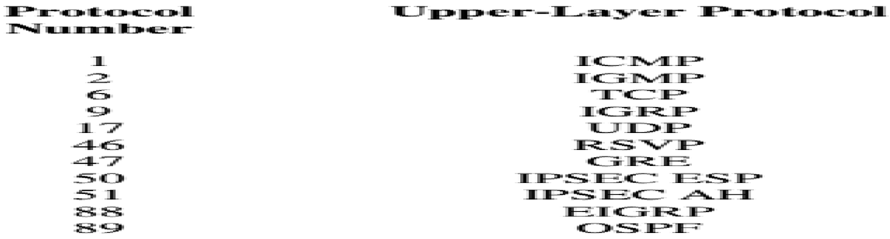
  Role of TTL field: Ensure that packet is eventually dropped when a **routing loop** occurs.

  Used as follows:

  - Sender sets the value (e.g., 64)
  - Each router decrements the value by 1
  - When the value reaches 0, the datagram is dropped

- **Protocol (1 byte):**

  - Specifies the higher-layer protocol.



Protocol Number	Upper-Layer Protocol
1	ICMP
2	IGMP
6	TCP
9	IGRP
17	UDP
46	RSVP
47	GRE
50	IPSEC ESP
51	IPSEC AH
88	EIGRP
89	OSPF

- **Header checksum (2 bytes):**

  - A simple 16-bit long checksum which is computed for the header of the datagram.
  - The receiving host will discard the packet if it fails the checksum calculation.

- **Options:** Security restrictions

  - **Record Route:** each router that processes the packet adds its IP address to the header.
  - **Timestamp:** each router that processes the packet adds its IP address and time to the header.
  - **Source Routing:** specifies a list of routers that must be traversed.

- **Padding:** Padding bytes are added to ensure that header ends on a 4-byte boundary.

- **Source and Destination Address**: 32 bit IP address.

Functions of the IP

1. **Addressing:**

   - In order to perform the job of delivering datagrams, IP must know where to deliver them to. For this reason, IP includes a mechanism for host addressing.

2. **Data Encapsulation and Formatting/ Packaging:**

   - IP accepts data from the transport layer protocols UDP and TCP.
   - It then encapsulates this data into an IP datagram using a special format prior to transmission.

3. **Fragmentation and Reassembly:**

    - IP *fragment* IP datagrams **into pieces**.
    - The receiving device uses the **reassembly function** to recreate the whole IP datagram again.

4. **Routing / Indirect Delivery:**

    - When an IP datagram must be sent to a destination on the **same local network**, this is done using ***direct delivery***.
    - However, if the final destination is on a distant network not directly attached to the source datagram must be **delivered *indirectly***.

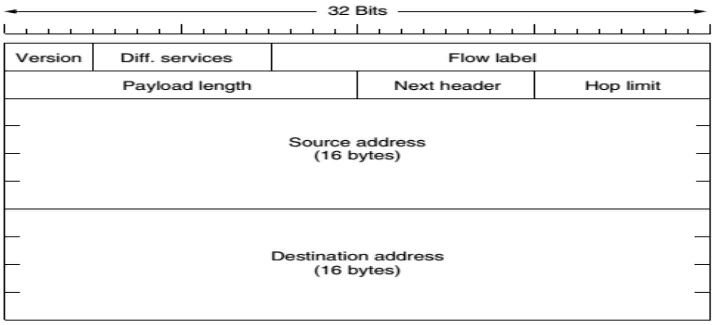
IPV6

- IP version 6 (IPv6) is an advanced version of IPv4.
- It takes all good features of IPv4 and adds new ones.
- **Larger address space**: IPv6 uses 128 bit(16 Bytes) Address.
- **Better header format**: This simplifies and speeds up the routing process.
- **New options.** IPv6 has new options to allow for additional functionalities.
- **Allowance for extension:** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.

IPv6 major goals:

1. Support **billions of hosts**.
2. Reduce the size of the routing tables.
3. Simplify the protocol.
4. Provide **better security** (authentication and privacy).
5. More attention to the type of service
6. Aid multicasting by allowing scopes to be specified.
7. Make it possible for a host to roam without changing its address.
8. Allow the protocol to evolve in the future.
9. Permit the old and new protocols to coexist for years.

IPv6 Header Format



Header Fields:

- **Version (4-bit): De**fines the version number of the IP. For IPv6, the value is 6.
- **Priority(4-bit):** Defines the priority of the packet with respect to traffic congestion.
- **Flow label (3-byte /24-bit):** It is designed to provide special handling flow of data.

- **Payload length(2-byte):** Defines the length of the IP datagram excluding the base header.
- **Hop limit (8-bit): S**erves the same purpose as the TTL field in IPv4.
- **Next header (**8-bit):
  - Defines the header that follows the base header in the datagram.
  - The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP.
  - Note that this field in version 4 is called the *protocol.*
- **Source address.:**
  - The source address field is a 16-byte (128-bit)
  - Internet address that identifies the **original source** of the datagram.
- **Destination address.**
  - The destination address field is a 16-byte (128-bit)
  - Internet address that usually identifies the **final destination** of the datagram.
  - However, if source routing is used, this field contains the address of the next router.

IPv4	**IPv6**
1. Source and destination addresses are 32 bits (4 bytes) in length.	1. Source and destination addresses are 128 bits(16 bytes)in length.
2. Uses broadcast addresses to send traffic to all nodes on a subnet.	2. There are no IPv6 broadcast addresses. Instead, multicast scoped addresses are used.
3. Fragmentation is supported at Originating hosts and intermediate routers.	3. Fragmentation is not supported at routers. It is only supported at the
4. IP header include a checksum.	4. IP header does not include a checksum.
5.IP header includes options.	5. All optional data is moved toIPv6 extension headers.
6.IPsec support is optional	6.IPsec support is required in a full IPv6 implementation.
7. No identification of payload for QoS Handling by routers is present within the IPv4 header.	7. Payload identification for QoS handling By routers is included in theIPv6 header using the Flow Label field.
8. Address must be configured either manually or through DHCP.	8. Addresses can be automatically assigned using stateless address auto configuration, assigned using DHCPv6, or manually configured.
9. *IP* address represented in decimal number system	*9. IP* address is represented in hexadecimal number system
10. *"."* used as seperator	10. *': '* used as separator .
11. Uses host address (A) resource records in the domain name system to map host names to IPv4 addresses.	11. Uses host address (AAAA) resource records in the domain name system to map host names to IPv6 addresses.

4. ICMP

- It is **internet control message protocol**.
- It **reports error** and sends **control messages**.
- **Error reporting** messages include – **destination unreachable**, **source quench** , **time exceed**, **parameter problem**, **redirection** etc.
- **Query message** includes –**echo request** and **reply**, **time stamp request and reply**, router solicitation and advertisement, etc.

Transport Layer Protocols

- Transport Layer Works on top of Internet Layer.
- It is concerned with transport of packets from the source to destination.
- In TCP/IP the transport layer is represented by two Protocols:
  - **TCP**
  - **UDP**

1. Transmission Control Protocol (TCP)

- TCP is transmission control protocol.
- It Provides:
  - Connection oriented service
  - Reliable service
  - Stream delivery service
  - Sending and receiving buffers
  - Bytes and segments
  - Full duplex service

- TCP is a **connection oriented protocol**.
  - *Connection oriented* means that a **virtual connection is established** before any user data is transferred.
  - If the connection cannot be established, the user program is **notified**.
  - If the connection is ever **interrupted**, the user program finds out there is a problem.

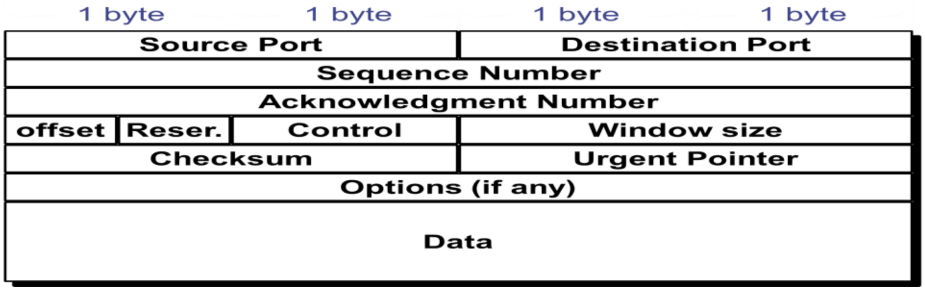
- **TCP is Reliable-**
  - *Reliable* means that every transmission of data is **acknowledged** by the receiver.
  - Reliable does not mean that things don't go wrong, it means that we find out when things go wrong.
  - If the sender does not receive acknowledgement within a specified amount of time, the sender retransmits the data.

- **Stream delivery service:**
  - TCP is a stream oriented protocol.
  - It allows the sending and receiving process to obtain as a stream of bytes.
  - TCP creates a working environment in such a way that the sending and receiving processes seem to be connected by an **imaginary "tube"** This is called as stream delivery service.
- **TCP : Flow Control**
  - **Sending and receiving buffers:**
  - The sending and receiving process may not **produce and receive data at the same speed**.
  - Hence TCP needs buffers for storage.
  - There are two types of buffers used in each direction:

  1) Sending buffer
  2) Receiving buffer
- **Full duplex service:**
  - TCP offers full duplex service where the data can flow in both the direction simultaneously.
  - The TCP segments are sent both the directions.

- **Process to process communication:**
  - The TCP uses port numbers as transport layer addresses.
  - Also called as Port to Port communication.

TCP Header

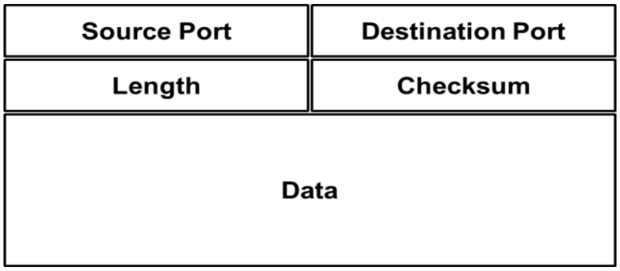


<table>
<tr><th colspan="2">1 byte</th><th colspan="2">1 byte</th><th colspan="2">1 byte</th><th colspan="2">1 byte</th></tr>
<tr><td colspan="4">Source Port</td><td colspan="4">Destination Port</td></tr>
<tr><td colspan="8">Sequence Number</td></tr>
<tr><td colspan="8">Acknowledgment Number</td></tr>
<tr><td>offset</td><td>Reser.</td><td colspan="2">Control</td><td colspan="4">Window size</td></tr>
<tr><td colspan="4">Checksum</td><td colspan="4">Urgent Pointer</td></tr>
<tr><td colspan="8">Options (if any)</td></tr>
<tr><td colspan="8">Data</td></tr>
</table>

2. UDP

- UDP is user datagram protocol.
- It is connectionless protocol because data is sent without establishing a connection between sender and receiver before sending the data.
- UDP is unreliable because data is delivered without acknowledgement.
- UDP does not perform Auto retransmission.
- UDP does not use flow control .
- UDP has high transmission speed.

UDP Datagram Format



Source Port	Destination Port
Length	Checksum
Data	

UDP Vs TCP

	UDP	TCP
Complexity	UDP is less complex	TCP is more complex
Connection	UDP is connection less protocol	TCP is connection oriented protocol
Reliability	It provides unreliable delivery of messages	It provides reliable delivery of messages
Function	By using this protocol one program can send a load of packets to another and that would be the end of the relationship.	As a message makes its way across the internet from one computer to another. This is connection based.
layer they exist	Transport layer	Transport layer
Flow controlling	UDP has no flow control	TCP has flow control
Overhead	Overhead is very low	Overhead is low
Which is powerful	UDP is less powerful	TCP is more powerful.

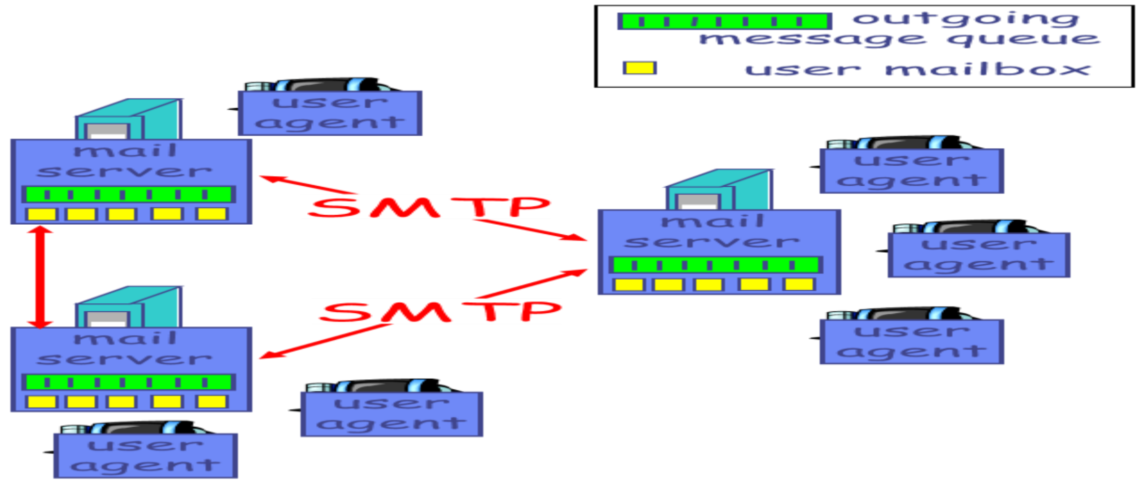
Application Layer Protocols

- SMTP
- FTP
- DNS
- Telnet

SMTP

- SMTP is simple mail transfer protocol.
- It is connection oriented text based protocol.
- Sender communicates with receiver using a command and supplying data over reliable TCP connection.
- SMTP is standard application layer protocol for delivery of email over TCP/IP network.
- SMTP establish a TCP connection between sender and port number 25 of receiver.

- Electronic Mail



- Three major components:
  - user agents
  - mail servers
  - simple mail transfer protocol: SMTP

<u>User Agent</u>

- Also called as “mail reader”
- composing, editing, reading mail messages e.g., Eudora, Outlook, Mozilla Thunderbird
- outgoing, incoming messages stored on server.

Mail Servers

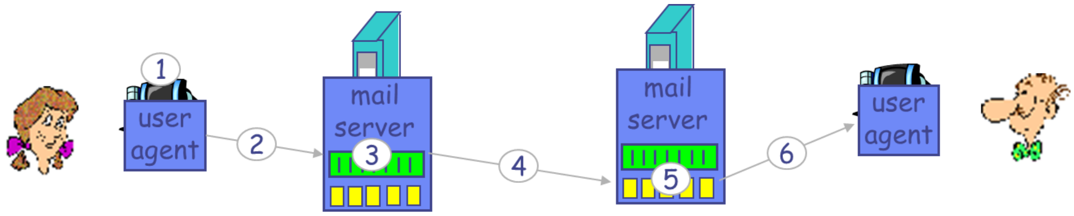
- mailbox contains incoming messages for user
- message queue of outgoing   (to be sent) mail messages

SMTP

- protocol between mail servers to send email messages
- client: sending mail server
- “server”: receiving mail server

Scenario: Alice sends message to Bob

1) Alice uses UA to compose message and “to” **bob@yahoo.com**
2) Alice’s UA sends message to her mail server; message placed in message queue
3) Client side of SMTP opens TCP connection with Bob’s mail server
4) SMTP client sends Alice’s message over the TCP connection
5) Bob’s mail server places the message in Bob’s mailbox
6) Bob invokes his user agent to read message

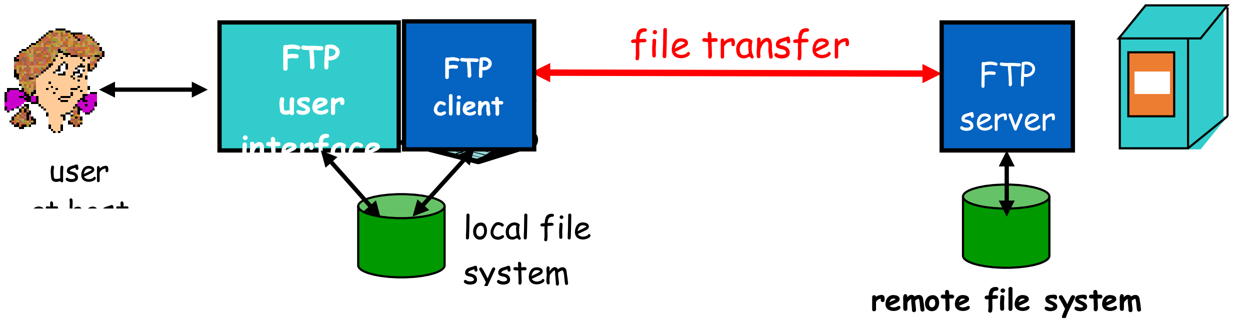


Sample SMTP interaction

```
S: 220 exite.com
C: HELO yahoo.com
S: 250  Hello exite.com, pleased to meet you

C: MAIL FROM: <alice@yahoo.com>
S: 250 .. Sender ok

C: RCPT TO: <bob@exite.com>
S: 250 .. Recipient ok

C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 exite.com closing connection
```

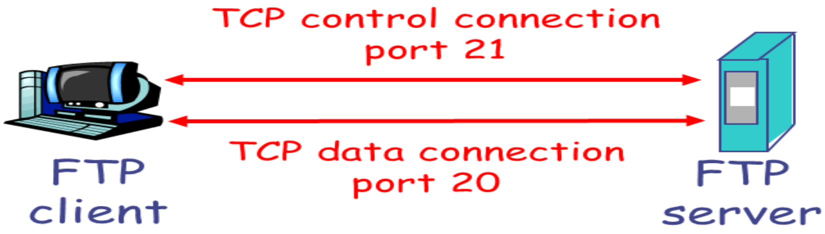
FTP

- FTP is used for copying a file from one host to the other.
- Some of the problem in transferring files :
  - Two systems may use different file name conventions.
  - Two systems may represent text data in different types.
  - The directory structure of the two systems may be different.
- FTP provides a simple solution to all these problems.
- FTP established two connections between the client and server.
- One is for data transfer and the other is for the control information.



FTP: separate control, data connections

- FTP client contacts FTP server at port 21
- client authorized over control connection.
- client browses remote directory by sending commands over control connection.
- when server receives file transfer command, server opens *2nd* TCP connection (for file) to client after transferring one file, server closes data connection.
- server opens another TCP data connection to transfer another file.
- FTP server maintains “state”: current directory, earlier authentication.



- **Control connection:**
  - Control connection remains alive during the entire process.
  - The IP uses **minimize delay** type services because this is an **interactive connection** between a user and server.
- **Data Connection:**
  - Data connection uses the **port 20** at the site.
  - This connection is opened when data to be transferred is ready and it is closed when transfer of data is over.
  - The service types used by IP is maximize throughput.

TELNET

- TELNET is abbreviation for Terminal Network.
- It is standard TCP/IP protocol for virtual terminal services proposed by ISO.
- TELNET enables establishment of connection to a remote system in such a way that a local terminal appears to be terminal at remote system.
- TELNET is general purpose client server application program.

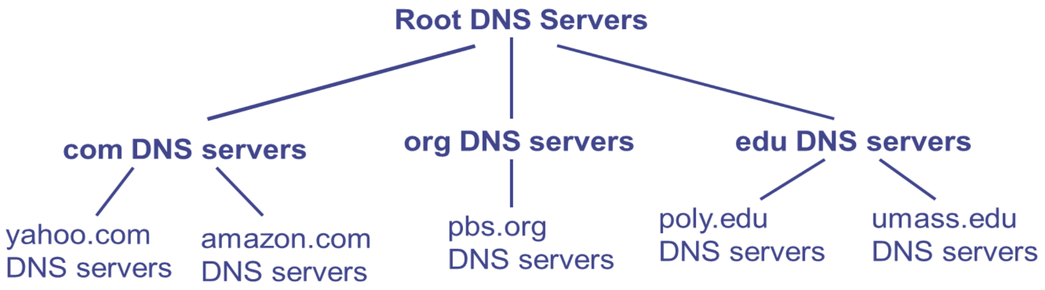
Remote login

- When user wants to access the application or utility located at the remote machine ,he or she performs remote login.
- Here the telnet client and server program come into use.

- The user sends the keystrokes to local operating system. local operating system accept is, but do not interpret them.
- The characters are send to TELNET client.
- TELNET client transform the character to a universal character set called Network Virtual Terminal Character and deliver them to the local TCP/IP stack.

DNS-Domain Name System

- Domain name is human readable name assigned to computer on the internet.
- Domain refers to group of computers called by common name.
- DNS is TCP/IP Application that maps Human Readable computer names to IP Addresses.
- DNS translates internet domain and host names to IP Addresses.



Example: Client wants IP for www.amazon.com:

- client queries a root server to find com DNS server
- client queries com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

Summery

- **Connectionless protocols:**
  - **IP**
  - **ICMP**
  - **UDP**
- **Connection oriented protocol:**
  - **TCP**
  - **SLIP**
  - **PPP**
  - **SMTP**

5.5 Comparison between OSI and TCP / IP Network Model.

OSI reference model	TCP/IP network model
1)It has 7 layers	**1)It has 4 layers**
2)Transport layer guarantees delivery of packets	**2)Transport layer does not guarantees delivery of packets**
3)Horizontal approach	**3)Vertical approach**
4)Separate presentation layer	**4)No session layer, characteristics are provided by transport layer**
5)Separate session layer	**5)No presentation layer, characteristics are provided by application layer**
6)Network layer provides both connectionless and connection oriented services	**6)Network layer provides only connection less services**
7)It defines the services, interfaces and protocols very clearly and makes a clear distinction between them	**7)It does not clearly distinguishes between service interface and protocols**
8)The protocol are better hidden and can be easily replaced as the technology changes	**8)It is not easy to replace the protocols**
9)OSI truly is a general model	**9)TCP/IP cannot be used for any other application**
10)It has a problem of protocol filtering into a model	**10) The model does not fit any protocol stack.**