Module – III

IP Addresses

- TCP/IP version 4 or IPv4 uses 32-bit for logical address and IPv6 uses 128-bit for logical address.
- An IP address represented in dotted decimal notation. Example- 123.22.33.44
- IP address is divided into net id or network id and host id.
- IP Addresses are divided into five classes: Class A, Class B, Class C, Class D, Class E.

IP Address Class	Starting Binary Value	First Address	Last Address	No. of Network	No. of Host
Class A	**0**	**1.0.0.0**	**126.255.255.254**	**2^7-1**	**2^{24}– 2**
Class B	**10**	**128.0.0.0**	**191.255.255.254**	**2^{14}**	**2^{16}-2**
Class C	**110**	**192.0.0.0**	**223.255.255.254**	**2^{21}**	**2^8-2**
Class D	**1110**	**224.0.0.0**	**239.255.255.254**	**Multicast**	
Class E	**1111**	**240.0.0.0**	**254.255.255.254**	**Undefined**	

Class A:

Network ID	Host ID		
←======8======>	←==========24==========>		
8 bit	**8 bit**	**8 bit**	**8 bit**

- It uses first octet for network address to uniquely identify the network and rest three octet for host address to uniquely identify the host on that network.
- An important rule is that network address cannot have all 8 bits 0 (zero).
- First bit is set to zero for class A, so following 7 bits in the first octet use to distinguish the network from other network.
- It means 2^7-1= 127 network i.e 1 to126
- Similar to the rule that the network portion of the address cannot be all 0s, the host portion of the address cannot be all 0s and it cannot be all 1s.
- A host portion with all 1s refers to an IP broadcast address.
- And the host portion with all 0s is a reference to the network.
- Class A network is: 2^{24}– 2 = 16,777,214 number of host.
- You subtract 2 because addresses with all 0s and all 1s are invalid.

Class B:

Network ID ←==========16==========>		**Host ID** ←==========16==========>	
8 bit	**8 bit**	**8 bit**	**8 bit**

- It uses first two octet for network address to uniquely identify the network and rest two octet for host address to uniquely identify the host on that network.
- "10" in the first 2 bits, the following 6 bits in the first octet and all 8 bits in the second octet for total 14 bits are used to distinguish this network from all other networks.
- Hence 2^{14}= 16,384 number of Class B networks.
- And 2^{16}-2= 65534 number of host on class B each network.

Class C:

Network ID ←==========24==========>			Host ID ←======8======>
8 bit	**8 bit**	**8 bit**	**8 bit**

- It uses first three octet for network address to uniquely identify the network and last octet for host address to uniquely identify the host on that network.
- "110" in the first 3 bits, the following 5 bits in the first octet , all 8 bits in the second octet and all 8 bits in the third octet for total 21 bits are used to distinguish this network from all other networks.
- Hence 2^{21}= 2,097,152 number of Class C networks.
- And 2^{8}-2=254 number of host on class C each network.

Class D:

- In the first octet, the first 4 bits are 1110.
- Class D addresses are called Multicast Address which cannot be used for host.
- The purpose of a multicast address is to enable a server somewhere to send data to a Class D address that no one host has so that several hosts can listen to that address at the same time. When you are watching TV on the Internet or listening to the radio on the Internet, your computer is listening to a Class D address. No server is sending data directly to your workstation; instead, a server is sending data to the multicast address. Any host can use software to listen for data at that address, and many hosts can be listening at once.

Class E:

- In the first octet, the first 4 bits are 11110.

- Class E addresses are reserved addresses and are invalid host addresses. They are used for experimental purposes by the IETF.

Special Address:

- Address use for Private use

  Class A: 10.0.0.0 to 10.255.255.255

  Class B: 172.16.0.0 to 172.31.255.255

  Class C: 192.168.0.0 to 192.168.255.255

- Loop Back Address

  127.0.0.0 to 127.255.255- For testing the TCP/IP connection.

  It cannot be used for host addressing.

Subnetting

Definition: It allows a network admin or user to create multiple networks from a single IP address block.

- Internet address (IP) is divided into different classes (such as A, B, C, D, E). Each IP class is associated with the default Subnet address. For example Class A its default subnet mask is 255.0.0.0.
- Default subnet mask also specifies the fixed number of Network and fixed number of Host for each class.
- So, in IP class didn't support flexibility to have more or less number of host per network.
- For example in class A IP address 2^{24}-2 host per network and it very difficult to manage in a single network.

Classless Inter Domain Routing (CIDR)

- It provide flexibility to have more or less number of host per network.
- Few bits of host part of an IP address can be borrowed and using them as network part of the Same IP address. The process of borrowing bits from host part to network part of an IP is called Subnetting**.**
- **Using** subnetting a large network can be subdivided into small sub-networks.

- **Subnetting increases** the hierarchy of routing.
- Below tables shows the possible numbers of subnets and host per subnet.

Class C Subnet

Network Bits	Subnet Mask	Bits Borrowed	Subnets	Hosts/Subnet
24	255.255.255.0	0	1	254
25	255.255.255.128	1	2	126
26	255.255.255.192	2	4	62
27	255.255.255.224	3	8	30
28	255.255.255.240	4	16	14
29	255.255.255.248	5	32	6
30	255.255.255.252	6	64	2

Class A Subnet

Network Bits	Subnet Mask	Bits Borrowed	Subnets	Hosts/Subnet
8	255.0.0.0	0	1	16777214
9	255.128.0.0	1	2	8388606
10	255.192.0.0	2	4	4194302
11	255.224.0.0	3	8	2097150
12	255.240.0.0	4	16	1048574
13	255.248.0.0	5	32	524286
14	255.252.0.0	6	64	262142
15	255.254.0.0	7	128	131070
16	255.255.0.0	8	256	65534
17	255.255.128.0	9	512	32766
18	255.255.192.0	10	1024	16382
19	255.255.224.0	11	2048	8190
20	255.255.240.0	12	4096	4094
21	255.255.248.0	13	8192	2046
22	255.255.252.0	14	16384	1022
23	255.255.254.0	15	32768	510
24	255.255.255.0	16	65536	254
25	255.255.255.128	17	131072	126
26	255.255.255.192	18	262144	62
27	255.255.255.224	19	524288	30
28	255.255.255.240	20	1048576	14
29	255.255.255.248	21	2097152	6
30	255.255.255.252	22	4194304	2

Class B Subnet

Network Bits	Subnet Mask	Bits Borrowed	Subnets	Hosts/Subnet
16	255.255.0.0	0	0	65534
17	255.255.128.0	1	2	32766
18	255.255.192.0	2	4	16382
19	255.255.224.0	3	8	8190
20	255.255.240.0	4	16	4094
21	255.255.248.0	5	32	2046
22	255.255.252.0	6	64	1022
23	255.255.254.0	7	128	510
24	255.255.255.0	8	256	254
25	255.255.255.128	9	512	126
26	255.255.255.192	10	1024	62
27	255.255.255.224	11	2048	30
28	255.255.255.240	12	4096	14
29	255.255.255.248	13	8192	6
30	255.255.255.252	14	16384	2

Advantage

- Increasing or maximizing the addressing efficiency
- Extending the life of IPv4 addresses (By setting required no. of Host per network).

Example:

Subnetting Base address 192.168.1.0/24 ("/" (Slash)Number – Tells no. of bit used for network part of a given IP address**)**

Here 3 octant are used as network part so, in default subnet mask's first 3 octant are 255 and the last octant is host part so 0.

IP	**192**	**168**	**1**	**0**	**/24**
Subnet Mask	**255**	**255**	**255**	**0**	
Subnet Mask (binary)	**11111111**	**11111111**	**11111111**	**00000000**	

- **Transform 1 host bit into a network bit**

So the new base IP is 192.168.1.0/25

New Subnet mask is 255.255.255.128

IP	192	168	1	0	/25
Subnet Mask	**255**	**255**	**255**	**128**	
Subnet Mask (binary)	**11111111**	**11111111**	**11111111**	**10000000**	

Two New networks are created one

1st – 192.168.1.0/25 (192.168.1.00000000/25)

2nd - 192.168.1.128/25(192.168.1.10000000/25)

Number of host per Subnet = **2^h-2 (h=7) =128-2= 126**

Formula

Number of subnetwork= 2^b (b- Number of bits borrowed)

Number of host per subnetwork= 2^h-2 (After borrowing remaining host bits-h)

- **Transform 2 bits into network bits.**

IP	192	168	1	0	/26
Subnet Mask	**255**	**255**	**255**	**0**	
Subnet Mask (binary)	**11111111**	**11111111**	**11111111**	**11000000**	

So , Number of subnetwork = 2^b (b=2)= 2^2=4 , those are

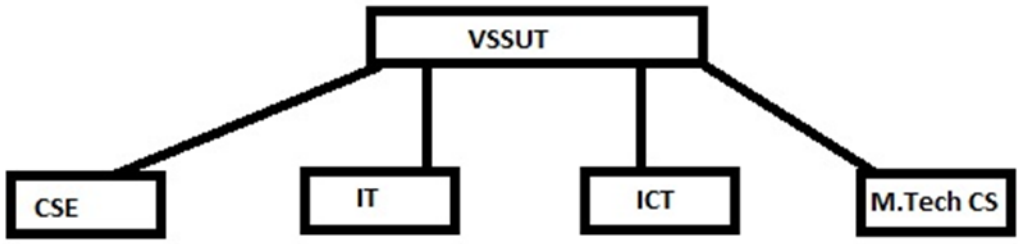
1. 192.168.1.0/26=====192.168.1.00000000000/26
2. 192.168.1.64/26 ====192.168.1.01000000000/26
3. 192.168.1.128/26 ===192.168.1.10000000000/26
4. 192.168.1.192/26=== 192.168.1.11000000000/26

Number of host per Subnet = **2^h-2 (h=6) =64-2= 62**

Similarly you can refer above table (For Class A, B, C) to check for number of host and number of subnet.

Question 1.

Create a network for following diagram with private IP 192.168.32.0. Each network at layer 2 can connect minimum of 50 hosts. **Subnet** the base address in order to yield maximum address utilization.



Find out following for VSSUT, CSE, IT, ICT and M.tech CS network

A. Range of IP address and subnet mask
B. IP address for broadcast and network

Answer :

- **Step 1** Find the smallest value of h that meet the criteria

  $2^h-2 >=50$

  Hit and trail

  - h=1, 0>=50 No
  - h=2, 2>=50 No
  - h=3, 6>=50 No
  - h=4, 14>=50 No
  - h=5, 30>=50 No
  - h=6, 62>=50 More than required but still ok
  - h=7, 126>=50 More than required

  So h==6 is ok

Step 2 Find number of borrow bit

b=32-(n+h)

Where, b= number of bit borrow

n= number of network bit

h=number of host bit (which is calculated in step 1)

So, b= 32-(24+6)=2

Number of new subnetwork = 2^2=4 and 62 host per network

- Answer **A** and B **New Subnet mask**

  255.255.255.11000000/26

Subnet Name	**Starting IP Address**	**Last IP Address**	**Subnet mask**	**Network IP Address**	**Broadcast address**
CSE	192.168.32.1	192.168.32.62	255.255.255.192	192.168.32.0	192.168.32.63
IT	192.168.32.65	192.168.32.126	255.255.255.192	192.168.32.64	192.168.32.127
ICT	192.168.32.129	192.168.32.190	255.255.255.192	192.168.32.128	192.168.32.191
M.tech CS	192.168.32.193	192.168.32.254	255.255.255.192	192.168.32.192	192.168.32.255

Supernet

- A supernetwork, or supernet, is formed by the combination of two or more networks (or subnets) .
- CIDR is used for supernet.
- The subnets required to create a single supernet must have prefix("/"number).
- The benefits of supernetting are conservation of address space and efficiencies gained in routers in terms of memory storage of route information and processing overhead when matching routes.

Example : (Source wikipedia)

A company that operates 150 accounting services in each of 50 districts has a router in each office connected with a Frame Relay link to its corporate headquarters. Without supernetting, the routing table on any given router might have to account for 150 routers in each of the 50 districts, or 7500 different networks. However, if a hierarchical addressing system is

implemented with supernetting, then each district has a centralized site as interconnection point. Each route is summarized before being advertised to other districts. Each router now only recognizes its own subnet and the other 49 summarized routes.

The determination of the summary route on a router involves the recognition of the number of highest-order bits that match all addresses. The summary route is calculated as follows. A router has the following networks in its routing table:

192.168.98.0
192.168.99.0
192.168.100.0
192.168.101.0
192.168.102.0
192.168.105.0

1. Firstly, the addresses are converted to binary format and aligned in a list:

Address	First Octet	Second Octet	Third Octet	Fourth Octet
192.168.98.0	11000000	10101000	01100010	00000000
192.168.99.0	11000000	10101000	01100011	00000000
192.168.100.0	11000000	10101000	01100100	00000000
192.168.101.0	11000000	10101000	01100101	00000000
192.168.102.0	11000000	10101000	01100110	00000000
192.168.105.0	11000000	10101000	01101001	00000000

2. Secondly, the bits at which the common pattern of digits ends are located. These common bits are shown in red. Lastly, the number of common bits is counted. The summary route is found by setting the remaining bits to zero, as shown below. It is followed by a slash and then the number of common bits.

First Octet	**Second Octet**	**Third Octet**	**Fourth Octet**	**Address**	**Subnet mask**
11000000	10101000	01100000	00000000	192.168.96.0	/20

The summarized route is 192.168.96.0/20. The subnet mask is 255.255.240.0.

This summarized route also contains networks that were not in the summarized group, namely, 192.168.96.0, 192.168.97.0, 192.168.103.0, 192.168.104.0, 192.168.106.0,

192.168.107.0, 192.168.108.0, 192.168.109.0, 192.168.110.0, 192.168.111.0. It must be assured that the missing network prefixes do not exist outside of this route.

In another example, an ISP is assigned a block of IP addresses by a regional Internet registry (RIR) of 172.1.0.0 to 172.1.255.255. The ISP might then assign subnetworks to each of their downstream clients, e.g., Customer A will have the range 172.1.1.0 to 172.1.1.255, Customer B would receive the range 172.1.2.0 to 172.1.2.255 and Customer C would receive the range 172.1.3.0 to 172.1.3.255, and so on. Instead of an entry for each of the subnets 172.1.1.x and 172.1.2.x, etc., the ISP could aggregate the entire 172.1.x.x address range and advertise the network 172.1.0.0/16 on the Internet community, which would reduce the number of entries in the global routing table.

Networking protocols in TCP/IP:

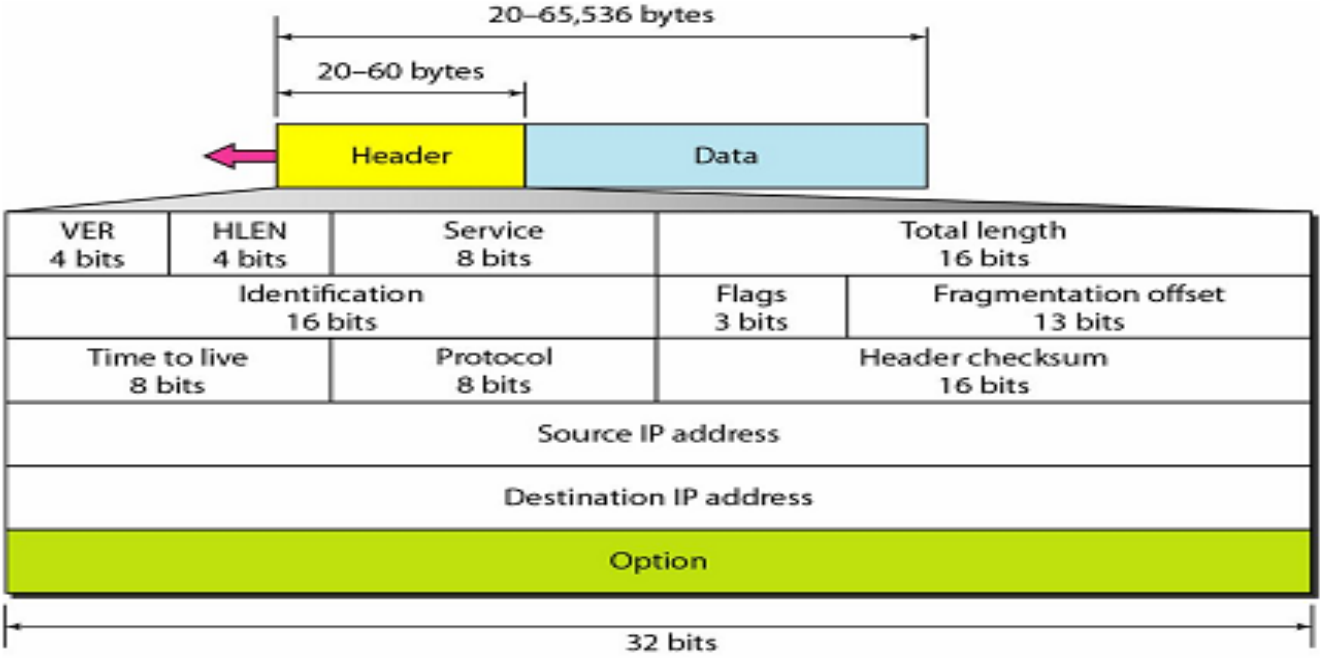
Internet Protocol (IP)

- The Internet Protocol (IP) is a network-layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed.
- IP is documented in RFC 791 and is the primary network-layer protocol in the Internet protocol suite. Along with the Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols.
- IP has two primary responsibilities: providing connectionless, best-effort delivery of datagrams through an internetwork; and providing fragmentation and reassembly of datagrams to support data links with different maximum-transmission unit (MTU) sizes.

IP Packet Format

Packets in the IP layer are called datagrams. A datagram divided into two parts : Header and Data

Header can be from 20 to 60 bytes and contains information for routing and delivery of data.



IP packet fields Details:

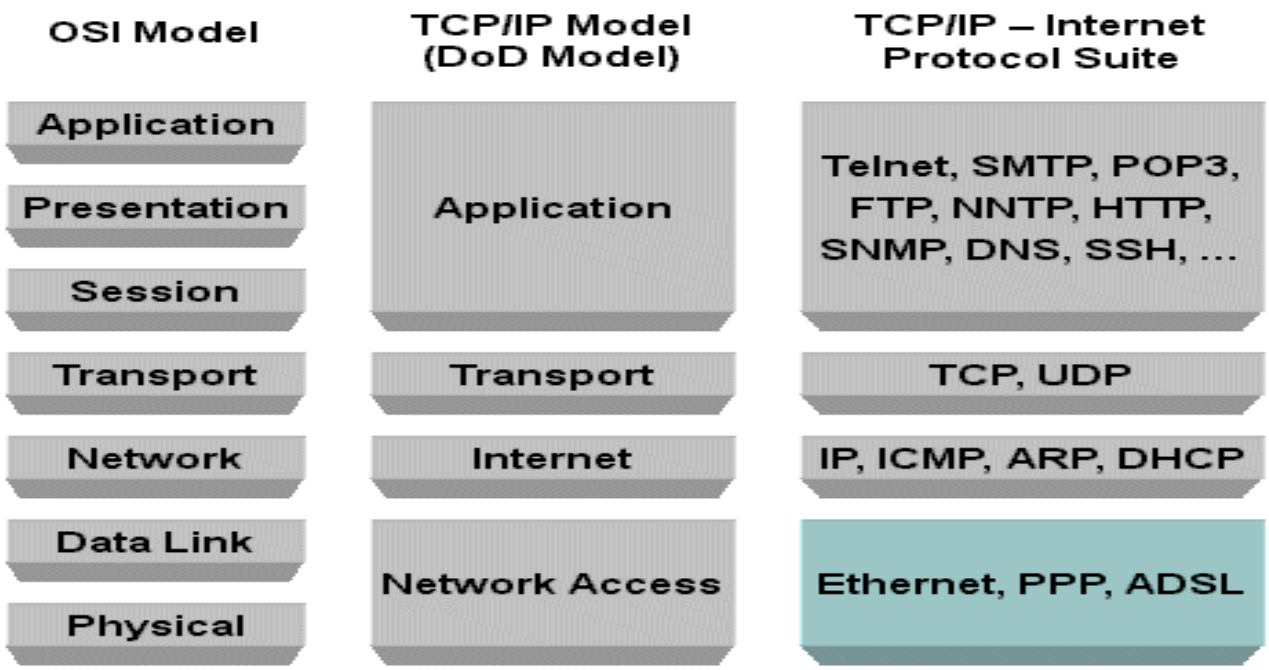
- **Version:** Indicates the version of IP currently used.
- **IP Header Length (IHL) :** Indicates the datagram header length in 32-bit words.
- **Type-of-Service:** Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.
- **Total Length:** Specifies the length, in bytes, of the entire IP packet, including the data and header.
- **Identification:** Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.
- **Flags:** Consists of a 3-bit field of which the two low-order (least-significant) bits control fragmentation. The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

- **Fragment Offset:** Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.
- **Time-to-Live:** Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.
- **Protocol:** Indicates which upper-layer protocol receives incoming packets after IP processing is complete.
- **Header Checksum:** Helps ensure IP header integrity.
- **Source Address:** Specifies the sending node.
- **Destination Address:** Specifies the receiving node.

IPv6 : Read from book

Network Protocol Overview:

- Network protocol is a set of rules to govern the communication between hosts (or computers) on a network.
- TCP/IP model protocol is widely used in the internet for communication.
- It is known as TCP/IP model because its most important protocols are TCP (Transmission control protocol) and IP (Internet protocol). And these protocols were first defined in this standard.
- In TCP/IP model provides set of protocols in each and every layers.
- Protocols in TCP/IP model are loosely coupled.
- Below figure shows the internet protocols suite used in TCP/IP model.



1. ARP and RARP

Introduction

- A data link such as Ethernet or a token ring has its own addressing scheme.
- When an Ethernet frame is sent from one host to another, it is the 48-bit Ethernet address that determines the destination.
- The first 28-bits are the organization that made the Ethernet card, the second 28-bits are randomly assigned by the manufacturer.
- The device driver software never looks at the destination IP address in the IP datagram.
- Address resolution provides a mapping between two different forms of addresses ie., 32-bit IP addresses and whatever the data link uses.

1.1. Address Resolution Protocol (ARP)

- A Network layer protocol used to associate a logical address (IP) to a physical (hardware) address (MAC). Obtains the hardware address (Ethernet) of another computer on the same network (subnet). This information is stored in ARP table for future reference.
- ARP table is created in the host system.
- ARP (address resolution protocol) is a protocol used to do address resolution in the TCP/IP protocol suite.
- ARP provides a dynamic mapping from an IP address to the corresponding hardware address.

- The protocol broadcast a packet containing the IP address of the destination machine. The machine with that address, or possibly a server, sends a reply containing the hardware address.
- This process is used by all network devices - computers, routers, printers. Address resolution provides a mapping between the two different forms of address ie., 32-bit Internet address and 48-bit Ethernet address.
- The term dynamic is used since it happens automatically and is normally not a concern of either the application user or the system administrator.

What happens if the device is unable to locate the destination MAC address in its ARP table? In other words, the source knows the destination IP address, but is unable to locate a MAC address for it in its own ARP table.

- The device sends an ARP request packet to all devices on the subnet asking for the MAC address corresponding to the IP address. The MAC address in this request is in the form of a broadcast: FF-FF-FF-FF-FF-FF. All devices will see the broadcast.
- The device with the destination IP address will send a reply back to the requesting device. Once the sending device has both of the destination's IP and Mac address in its ARP table, it can send data at any time.
- If network devices did not keep an ARP table in memory (cache), they would have to send an ARP request every time data needs to be transmitted.

The ARP request message contains the following details:

- MAC header of

  Destination: FF-FF-FF-FF-FF-FF
  Source: AA-AA-AA-AA-AA-AA
- IP Header of

  Destination: 192.168.2.1
  Source: 192.168.2.2
- ARP Request Message

  "What is your MAC address?"

The ARP reply contains the following:

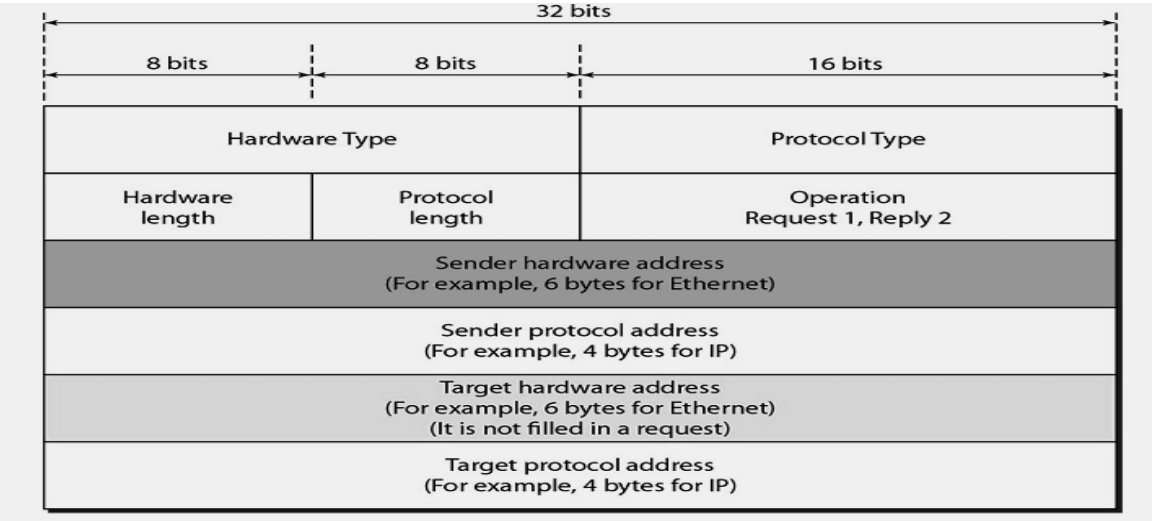
- MAC header of

  Destination: AA-AA-AA-AA-AA-AA
  Source: FF-FF-FF-FF-FF-FF
- IP Header of

  Destination: 192.168.2.2
  Source: 192.168.2.1
- ARP Request Message

  "What is your MAC address?"

1.1.1. ARP Packet format



1.1.2. ARP Cache

In each host ARP cache is maintained in main memory. This cache is updated automatically in each 20 minute interval.

1.1.3. Gratuitous ARP

It is a type of ARP request used to looking for own MAC address.

- **Reverse Address Resolution Protocol (RARP)**

- RARP does basically the opposite of ARP.
- A network device knows its MAC address but not its IP address. Used in diskless workstations or dumb terminals ARP tables are kept in RAM, and therefore lost after the power is turned off .
- Devices using RARP require that a RARP server be present on the network to answer RARP request Routers build tables that describe all devices and networks connected to them.
- In other words, ARP tables kept by routers can contain IP addresses and MAC addresses of devices located on more than one network. It used to require the Ethernet address of the IP address.
- The principle of RARP is for the diskless system to read its unique hardware address from the interface card and send an RARP request asking for someone to reply with the diskless system's IP address.

- The source device sends an RARP request packet to all devices on the subnet asking for an IP address. The IP address in this request is in the form of a broadcast: 197.15.22.255.
- All devices will see the broadcast, but the only RARP server will act upon it. The RARP server will send a reply packet containing the assigned IP address. That IP address is used for the session duration. RARP is used by systems without a disk drive but requires manual configuration by the system administrator.

The RARP request message contains the following details:

- MAC header of

  Destination: Any device who will listen (RARP server)
  Source: AA-AA-AA-AA-AA-AA
- IP Header of

  Destination: 192.168.2.1
  Source: --.--.--.--
- ARP Request Message
  "What is my IP address?"

The RARP reply contains the following:

- MAC header of

  Destination: AA-AA-AA-AA-AA-AA
  Source: BB-BB-BB-BB-BB-BB
- IP Header of

  Destination: 192.168.2.2
  Source: 192.168.2.1
- ARP Request Message
  "What is your IP address?"

- **RARP Packet format**
  - It is almost identical to an ARP packet.
  - The only differences are that the frame type for an RARP request or reply, and the operation field has a value of 3 for an RARP request and 4 for an RARP reply.
- **RARP Servers as User Processes**

  The complication with an RARP server is that the server normally provides the mapping from a hardware address to an IP address for many hosts. RARP requests are transmitted as Ethernet frames with a specific Ethernet frame type field.

ICMP

Problem with IP (Internet Protocol):

- Internet Protocol provides an unreliable, connectionless delivery of datagram from router to router. A datagram travels from router to router until it reaches to its final destination.
- As IP provides connectionless datagram delivery, datagram are travel independently in network from router to router without any coordination.
- Routers also work autonomously and deliver datagram using IP address.
- IP router work fine then there is no problem to use IP only.
- But, if a router cannot route or deliver a datagram due to fault or failure of router, or router buffer full or due to network congestion, then the router needs to inform the source to take action to avoid or correct the problem.

Solution:

Internet Control Message Protocol (ICMP):

- The Internet Control Message Protocol (ICMP) protocol is classic example of a client server application. The Internet Control Message Protocol (ICMP) is part of the Internet protocol suite and defined in RFC 792 .
- The ICMP server executes on all IP end system computers and all IP intermediate systems (i.e routers).
- The protocol is used to report problems with delivery of IP datagrams within an IP network. It can be sued to show when a particular End system is not responding, when an IP network i s not reachable, when a node is overloaded, when an error occurs in the IP header information, etc.
- The protocol is also frequently used by Internet managers to verify correct operations of End Systems and to check that routers are correctly routing packets to the specified destinations.

- The Internet Protocol (IP) is used for host-to-host datagram service in a system of interconnected networks called the Internet. The network connecting devices are called Gateways.

- These gateways communicate between themselves for control purposes via a Gateway to Gateway Protocol (GGP). Occasionally a gateway or destination host will communicate with a source host, for example, to report an error in datagram processing.

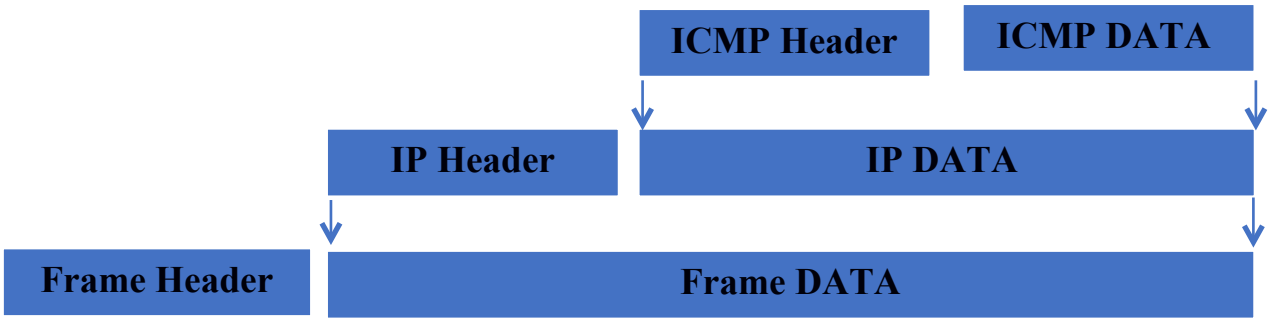


Figure 1 ICMP packet in IP

- ICMP, uses the basic support of IP as if it were a higher level protocol , however, ICMP is actually an integral part of IP, and must be implemented by every

IP module. Is a protocol for the exchange of error messages and other vital information between (Physical) Internet entities such as hosts and routers. ICMP is a network layer protocol; often it is placed next to the IP protocol.

ICMP6 : Read from book…..

DHCP

- A DHCP server stores all available IP addresses in a central database along with associated configuration information, including the subnet mask, gateways, and the addresses of DNS servers.
- This database enables automatic IP address configuration for hosts as they start up.
- DHCP saves network administrative time and the larger the network, the greater the savings. Without dynamic address assignment, network administrators must manage IP addresses to avoid duplicate use and apply configuration changes to workstations manually. The resulting lack of centralized configuration information makes it difficult for the administrator to ensure consistent client configurations.
- DHCP is derived from the Internet standard BOOTP, which allows dynamic assignment of IP addresses as well as remote booting of diskless workstations.
- In addition to supporting the dynamic assignment of IP addresses, DHCP supplies all configuration data required by TCP/IP, plus additional data required for specific services.
- Whenever a new computer starts on a network segment that is served by the DHCP server (or an existing computer is restarted), the computer asks for a unique IP address and the DHCP server assigns one from the pool of available addresses.

The process requires only four steps:

1. The DHCP client asks for an IP address (a DHCP Discover message).
2. The DHCP Server offers an address (a DHCP Offer message).
3. The DHCP client accepts the offer and requests the address (a DHCP Request message).
4. The DHCP Server officially assigns the address to the client (a DHCP Acknowledge message).

- DHCP server places an administrator-defined time limit, called a lease, on the address assignment. Halfway through the lease period, the DHCP client requests a lease renewal, and the DHCP server extends the lease. As a result, when a computer stops using its assigned IP address (for example, upon relocation to another network segment), the l ease expires and the address returns to the pool for reassignment.

Transmission Control Protocol (TCP)

Introduction:

- TCP is a transport layer protocol.
- It a connection oriented protocol.
- It creates a virtual circuit or connection between system before sending data.
- It is more reliable.

TCP Services

- **Process to Process communication** using port number. Few well known port number for process to process communication are shown in following table 1.

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FIP, Data	File Transfer Protocol (data connection)
21	FIP, Control	File Transfer Protocol (control connection)
23	TELNET	Tenninal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

Table 1 Well known port Number

- **Stream Delivery Service**
  - TCP is a stream-oriented protocol.

- It allows the sending process to deliver data as stream of bytes and allow receiver to obtained data as stream of bytes.
- Both the end system uses send and receiving buffer. One of the ways of implementing buffer is circular buffer.
- It keeps the data in the buffer until it receives an acknowledgement (ack). Sometimes receiver not able to receive the all the data due to slowness of the receiving process or due to congestion.
- **Segments**
  - The IP layer as a service provider for TCP needs to send data in packets not as a stream of bytes.
  - At transport layer, TCP groups a number of bytes together into packets called **Segment.**
  - TCP adds header to each segment (for control purpose) and deliver the segment to the IP layer for transmission.
  - The segments are encapsulated in IP datagrams and transmitted.
  - Each segment contains 1 to 1000 bytes.
- **Full-Duplex Communication**
  TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions.
- **Connection-Oriented Service**
  When a process at site A wants to send and receive data from another process at site B, the following occurs:
  1. The two TCPs establish a connection between them.
  2. Data are exchanged in both directions.
  3. The connection is terminated.

  Note that this is a virtual connection, not a physical connection. The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost, or corrupted, and then resent. Each may use a different path to reach the destination. There is no physical connection. TCP creates a stream-oriented environment in which it accepts the responsibility of delivering the bytes in order to the other site.

- **Reliable Service**
  TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data.

- TCP Features
  - **Numbering system**
    - TCP software keeps track of segments being transmitted or received.
    - There is no field called sequence number value.

- Instead there are two field called sequence number and acknowledgement number.
- These two field refers to the byte number not sequence number.
- Byte number
  - TCP numbers all data bytes that are transmitted in a connection. Numbering is independent in each direction.
  - When TCP receives bytes of data from a process, it stores them in the sending buffer and numbers them.
  - The numbering does not necessarily start from 0. Instead, TCP generates a random number between 0 and 2^{32}-1 for the number of the first byte.
  - For example, if the random number happens to be1057and the total data to be sent are 6000 bytes, the bytes are numbered from 1057 to 7056. Byte numbering is also used for flow and error control.
  - The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.
- Sequence Number

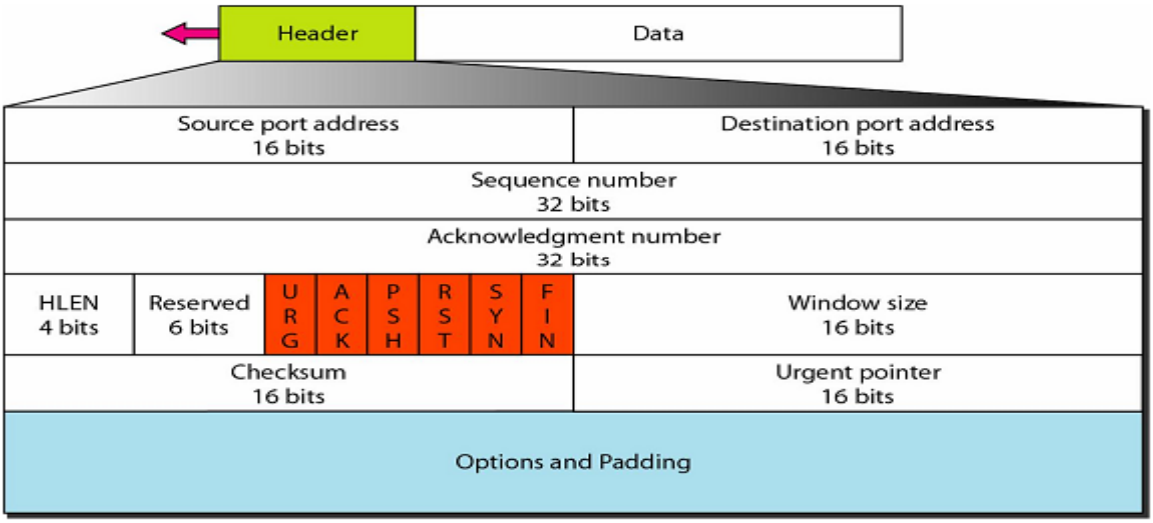
  After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment.
- Acknowledgment Number

  It defines the number of the next byte that receiver expected to receive. It uses a cumulative acknowledgement. If ack is 500 means before 500 all data are received.

- **Flow Control**
  - TCP provides flow control. The receiver of the data controls the amount of data that are to be sent by the sender.
  - This is done to prevent the receiver from being over whelmed with data.
  - The numbering system allows TCP to use a byte-oriented flow control.
- **Error Control**
  - To provide reliable service, TCP implements an error control mechanism.
  - Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented.
  -
- **Congestion Control**

  TCP takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network.

Segment Format



- **Source port address**. This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.
- **Destination port address**. This is a16-bit field that defines the port number of the application program in the host that is receiving the segment.
- **Sequence number.** This 32-bit field defines the number assigned to the first byte of data contained in this segment. As the TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment. During connection establishment, each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.
- **Acknowledgment number.** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number "x" from the other party, it defines "x+1" as the acknowledgment number. Acknowledgment and data can be piggy backed together.
- **Header length.** This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60bytes. Therefore, the value of this field can be between 5 (5 x 4=20) and 15 (15 x 4=60).
- **Reserved**. This is a 6-bit field reserved for future use.
- **Control Bits**. This field defines 6 different control bits or flags as tabulated in below

One or more of these bits can be set at a time.

Flag	*Description*
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

- **Window size**. This field defines the size of the window, in bytes, that the other party must maintain. Note that the length of this field is 16bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.
- **Checksum**.This16-bitfield contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one used for UDP. However, checksum for TCP is mandatory. The same pseudo header, serving the same purpose, is added to the segment. For the TCP pseudo header, the value for the protocol field is 6.
- **Urgent pointer.** This l6-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.
- **Options**. There can be upto 40 bytes of optional in formation in the TCP header.

Routing

- There are two distinct processes to delivering IP datagram: IP Forwarding and IP Routing.
- **IP Forwarding** tells how to pass a packet from an input interface to the output interface?
- **IP Routing** tells how to find and setup the routing tables? It determine route taken by packets from source to destination.

- Forwarding must be done as fast as possible. On routers, is often done with support of hardware. On PCs, is done in kernel of the operating system. Routing is less time-critical, On a PC, routing is done as a background process.
- Packets are transfer from one system to another system using for forwarding and transfer from the sender to the receiver using routing. Forwarding is a process of passing packets to the next hop. There is only one forwarding table and has prefix and next-hop information.
- Routing is a process of populating the forwarding table. You might have multiple routing databases - e.g. both OSPF and BGP Routing databases have more information. Routing is based on address lookup, maximum prefix match and search operation.
- Conceptually, IP routing is simple, especially for a host. If the destination is directly connected to the host (e.g., a point-to-point link) or on a shared network (e.g., Ethernet or token ring), then the IP datagram is sent directly to the destination. Otherwise the host sends the datagram to a default router, and lets the router deliver the datagram to its destination.

NOTE: The IP layer can be configured to act as a router in addition to acting as a host. Most multiuser systems today, including almost every Unix system, can be configured to act as a router.

Routing Protocol Vs Routed Protocol

Routing Protocols

Routing protocol used to update the routing table information. It will collect the information based on advertisement and also send routing information to other systems. Eg. RIP, OSPF

Routed Protocols

Routed protocols used to route the packets across network. It forward data to one network to another network Eg. IPX, DecNet

Types of routing table entries

- **Network route**: In this type of entry destination addresses is a network address (e.g.,10.0.2.0/24). Most entries are network routes.

- **Host route:** This type of entry destination address is an interface address (e.g., 10.0.1.2/32).This entry used to specify a separate route for certain hosts.
- **Default route:** This type of entry used when no network or host route matches. The router that is listed as the next hop of the default route is the default gateway.
- **Loopback address:** Routing table used the loopback address (127.0.0.1) which means the next hop lists the loopback (lo0) interface as outgoing interface.
- **Adding an interface**: Configuring an interface eth2 with 10.0.2.3/24 adds a routing table entry.
- **Adding a default gateway**: Configuring 10.0.2.1 as the default gateway adds the entry

The steps that IP performs when it searches its routing table:

1. Search for a matching host address.
2. Search for a matching network address.
3. Search for a default entry. (The default entry is normally specified in the routing table as a network entry, with a network ID of 0) .

A matching host address is always used before a matching network address.

Routing table

Routing Table is used by the Routing Protocols. It defines the topology of the network. It must be consistent with other router's tables.

Classification of routing table

In routing table two types of protocol are used, I e Interior Gateway protocol s (IGP) and Exterior Gateway protocol s (EGP). Kind of information that is carried and the way the routing table are calculated based on Distance-vector protocols or Link-state protocols.

Interior Gateway protocols Vs Exterior Gateway Protocols

- Interior Gateway Protocol s are used within a single autonomous system. Generally it have single network administration to administration. It has unique routing policy and makes best use of network resources. This class of protocols are used inside an autonomous system, ex. - IP, OSPF, IGRP, EIGRP.

- Exterior Gateway Protocols are used among different autonomous systems. It has independent administrative entities. It is used to communication between independent network infrastructures. This class of protocol s is used outside, or between, autonomous systems, ex. –BGP4, the current internet standard for EGP. BGP makes routing decisions based on network policies, or rules. In EBGP, session occurs between routers in two different Autonomous Systems.

- In IBGP, session occurs between routers in the same Autonomous Systems.

Static or Dynamic Routing

Routing can be either static or dynamic, depending on how routing information is generated and maintained. In static routing, routing information is entered manually by an administrator and remains constant throughout the router's operation. In dynamic routing, a router is configured to automatically generate routing information and share the information with neighboring routers.

1. **Static routing**
   - In static routing, a network administrator enters static routes in the routing table manually by indicating the Network ID, the hop count and the router interface. The network ID, consisting of a destination IP address and a subnet mask. The hop count is the distance between this router and the neighboring router. The router interfaces through which forward the packets to the destination.

- Static routing has significant drawbacks. Because a network administrator defines a static route, errors are more likely than wi th a dynamically assigned route. A simple typographical error can create chaos on the network. An even greater problem is the inability of a static route to adapt to topology changes. Whenever the topology changes, the administrator might have to make changes to the routing tables entries on every static router.

2. Dynamic Routing

- Dynamic routing method has two parts: the routing protocol that is used between neighboring routers to convey information about their network environment, and the routing algorithm that determines paths through that network.
- The protocol defines the method used to share the information externally, whereas the algorithm is the method used to process the information internally.
- The routing tables on dynamic routers are updated automatically based on the exchange of routing information with other routers. The most common dynamic routing protocol s are Di stance vector routing protocol s and Link state routing protocols.
- Dynamic routes are adapting to a failure in a network and work in large networks. The disadvantages are increase in complexity and overhead on the lines and routers.

Routing

- Routing is the process of selecting best paths in a network. In the past, the term routing was also used to mean forwarding network traffic among networks. However this latter function is much better described as simply forwarding.
- Routing is the act of moving information across an inter-network from a source to a destination. Along the way, at least one intermediate node typically is encountered. It's also referred to as the process of choosing a path over which to send the packets. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. The primary difference between the two is that bridging occurs at Layer 2 (the data link layer) of the OSI reference model, whereas routing occurs at Layer 3 (the network layer). This distinction provides routing and bridging with

different information to use in the process of moving information from source to destination, so the two functions accomplish their tasks in different ways. The routing algorithm is the part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on, i.e. what should be the next intermediate node for the packet.

Routing protocols use metrics to evaluate what path will be the best for a packet to travel. A metric is a standard of measurement; such as path bandwidth, reliability, delay, current load on that path etc; that is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain routing tables, which contain route information. Route information varies depending on the routing algorithm used.

Routing algorithms fill routing tables with a variety of information. Mainly Destination/Next hop associations tell a router that a particular destination can be reached optimally by sending the packet to a particular node representing the "next hop" on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop. Some of the routing algorithm allows a router to have multiple “next hop” for a single destination depending upon best with regard to different metrics. For example, let’s say router R2 is be best next hop for destination “D”, if path length is considered as the metric; while Router R3 is the best for the same destination if delay is considered as the metric for making the routing decision.

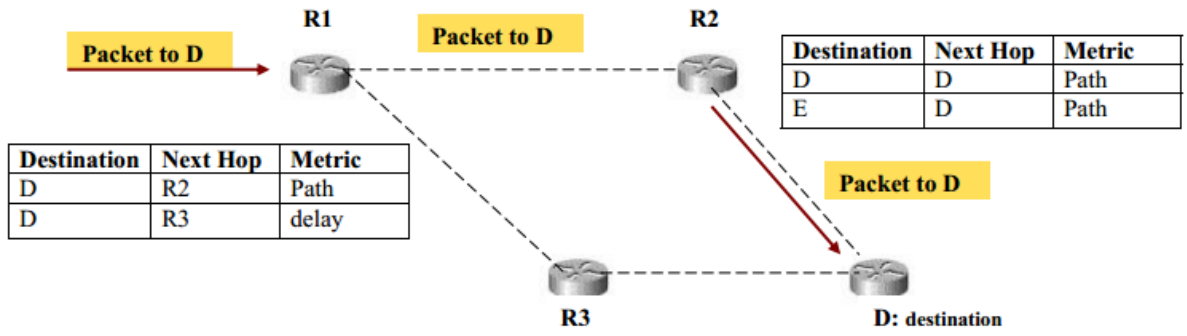


Figure.26 Typical Routing in a Small network

Figure.26 shows a small part of a network where packet destined for node “D”, arrives at router R1, and based on the path metric i.e. the shortest path to destination is forwarded to router R2 which forward it to the final destination. Routing tables also can contain other information, such as data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used. Routers communicate with one another and maintain their routing tables through the transmission of a variety of messages. The routing update message is one such message that generally consists of all or a portion of a routing table. By analysing routing updates from all other routers, a router can build a detailed picture of network topology. A link-state advertisement, another example of a message sent between routers, informs other routers of the state of the sender's links. Link information also can be used to build a complete picture of network topology to enable routers to determine optimal routes to network destinations.

While designing a routing protocol it is necessary to take into account the following design parameters:

- *Performance Criteria*: Number of hops, Cost, Delay, Throughput, etc
- *Decision Time*: Per packet basis (Datagram) or per session (Virtual-circuit) basis
- *Decision Place*: Each node (distributed), Central node (centralized), Originated node (source).
- *Network Information Source*: None, Local, Adjacent node, Nodes along route, All nodes.
- *Network Information Update Timing*: Continuous, Periodic, Major load change, Topology change.

There are different categories of routing protocols depicted by the following diagram.

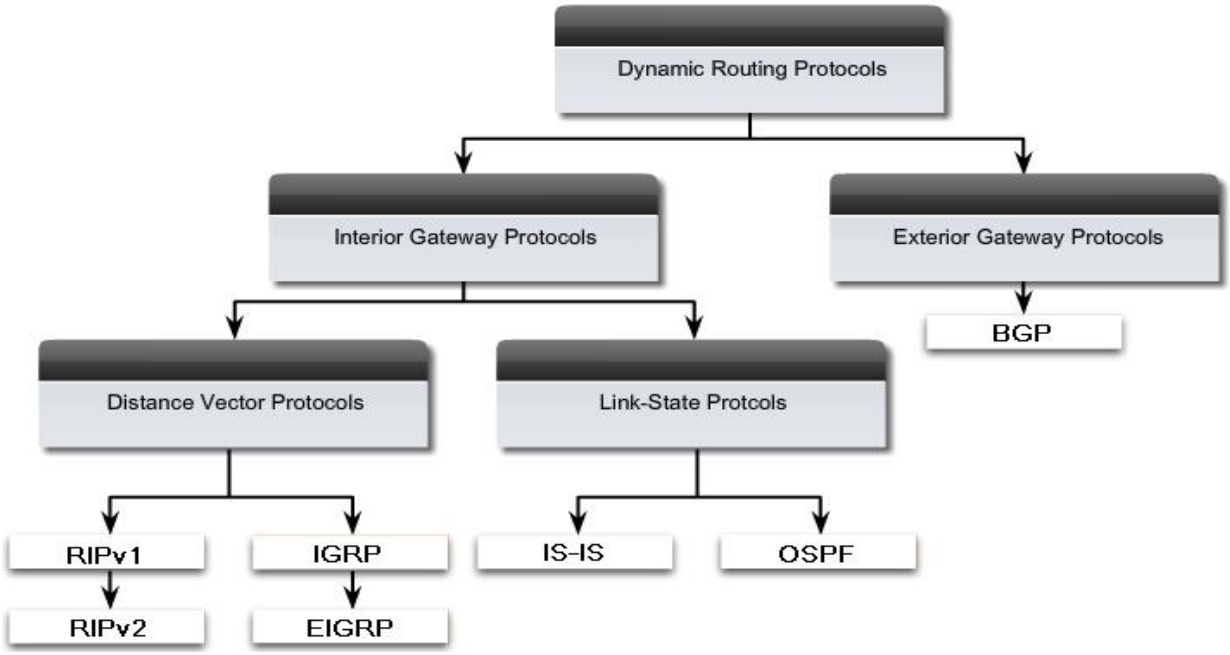


Figure.27 Routing protocol Categories

Distance vector routing protocols

- Distance vector routing protocols, the earliest dynamic routing protocols, are an improvement over static routing, but have some limitations. When the topology of the internetwork changes, distance vector routing protocols can take several minutes to detect the change and make the appropriate corrections.
- Distance vector protocols advertise periodically sends to his neighbours and find how far is the destination and how to the next hop to get there. It installs routes directly in tables.
- The distance vector algorithm, also known as the Bellman-Ford algorithm, enables a router to pass route updates to its neighbours at regularly scheduled intervals.
- Each neighbour then adds its own distance value and forwards the routing information on to its immediate neighbours.
- The result of this process is a table containing the cumulative distance to each network destination.
- One advantage of distance vector routing protocols is simplicity. Distance vector routing protocols are easy to configure and administer. They are well suited for small networks.

- Most distance vector routing protocols use a hop count as a routing metric. A routing metric is a number associated with a route that a router uses to select the best of several matching routes in the IP routing table.
- The hop count is the number of routers that a packet must cross to reach a destination.

Link state routing protocols

- Link state routing protocols are more reliable and require less bandwidth than do distance vector routing protocols, they are also more complex, more memory-intensive, and place a greater load on the CPU.
- Link state routing protocols address some of the limitations of distance vector routing protocols. For example, link state routing protocols provide faster convergence than do distance vector routing protocols. (Convergence is the process by which routers update routing tables after a change in network topology and the change is replicated to all routers that need to know about it)
- In Link-state protocols each router sends information about links to which it is attached state of these links. It is flooded throughout the network. Every router calculates its routing table.
- Unlike distance vector routing protocol s, which broadcast updates to all routers at regularly scheduled intervals, link state routing protocol s provide updates only when a network link changes state. When such an event occurs, a notification in the form of a link state advertisement is sent throughout the network.

Autonomous System

- As internet is a network of network that spans the entire world and because it's not under the control of a single organization or body, one cannot think of forcing a single policy for routing over it. Thus, comes the concept of autonomous system.
- An Autonomous System (AS) is a connected segment of a network topology that consists of a collection of subnetworks (with hosts attached) interconnected by a set of routes. The subnetworks and the routers are expected to be under the control of a single operations

and maintenance (O&M) organization i.e., an AS is under the same administrative authority.

- These ASs share a common routing strategy. An AS has a single "interior" routing protocol and policy. Internal routing information is shared among routers within the AS, but not with systems outside the AS. However, an AS announces the network addresses of its internal networks to other ASs that it is linked to. An AS is identified by an Autonomous System number.

RIP (Routing Information Protocol)

- It is used within an AS.
- Routing Information Protocol (RIP) first used in XNS (Xerox Network Systems). RIP first documented in RFC 1058.
- Routing Information Protocol (RIP) is the best known and most widely used of the distance vector routing protocols. RIP version 1 (RIP v1), which is now outmoded, was the first routing protocol accepted as a standard for TCP/IP. RIP version 2 (RIP v2) provides authentication support, multicast announcing, and better support for classless networks.

- The Windows Server 2003 Routing and Remote Access service supports both RIP v1 and RIP v2 (for IPv4 only).
- Using RIP, the maximum hop count from the first router to the destination is 15. Any destination greater than 15 hops away are considered unreachable. This limits the diameter of a RIP internetwork to 15.
- In RIP packets are sent every 30 seconds or faster when necessary. Route is considered down if it is not refreshed within 180 sec. (Distance set to infinity).
- It doesn't support classless routing.

OSPF (Open Shortest path First) Routing Protocol

Open Shortest Path First (OSPF) is a routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single autonomous system (AS).

OSPF is defined in RFCs 1245–1247, 1253 and 1583. OSPF was designed specifically for the TCP/ IP Internet environment, and supports the following features:

- Authentication of routing updates.
- Tagging of externally-derived routes.
- Fast response to topology changes with low overhead.
- Load sharing over meshed links.

Open Shortest Path First (OSPF) is an Interior Gateway Routing Protocol, based on Shortest Path First (SPF) or link-state technology. In SPF-based routing protocols, each router maintains a database describing the Autonomous System's (AS) topology. Each router has an identical database. Each piece of this database describes a particular router and its current state, which includes the state of the interfaces, reachable neighbours, and other information. The router distributes this information about the Autonomous System by "flooding".

Few of the important features of OSPF are as follows:

- OSPF is based on the SPF algorithm, which is also referred to as the Dijkstra's algorithm, named after the person credited with its creation.
- OSPF is a link-state routing protocol that calls for the sending of link-state advertisements (LSAs) to all other routers within the same hierarchical area. Information on attached interfaces, metrics used, and other variables are included in OSPF LSAs.
- OSPF specifies that all the exchanges between routers must be authenticated. It allows a variety of authentication schemes; even different areas can choose different authentication schemes. The idea behind authentication is that only authorized router are allowed to advertise routing information.
- OSPF include Type of service Routing. It can calculate separate routes for each Type of Service (TOS), for example it can maintain separate routes to a single destination based on hop-count and high throughput.
- OSPF provides Load Balancing. When several equal-cost routes to a destination exist, traffic is distributed equally among them.

- OSPF allows supports host-specific routes, Subnet-specific routes and also network-specific routes.
- OSPF allows sets of networks to be grouped together. Such a grouping is called an Area. Each Area is self-contained; the topology of an area is hidden from the rest of the Autonomous System and from other Areas too. This information hiding enables a significant reduction in routing traffic.
- OSPF uses different message formats to distinguish the information acquired from within the network (internal sources) with that which is acquired from a router outside (external sources).

Routing Hierarchy in OSPF

- Unlike RIP, OSPF can operate within a hierarchy. The largest entity within the hierarchy is the autonomous system (AS), which is a collection of networks under a common administration that share a common routing strategy. OSPF is an intra-AS (interior gateway) routing protocol, although it is capable of receiving routes from and sending routes to other ASs.
- A topological database is essentially an overall picture of networks in relationship to routers.
- The term domain sometimes is used to describe a portion of the network in which all routers have identical topological databases. Domain is frequently used interchangeably with AS. An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than it would if the AS were not partitioned.
- Area partitioning creates two different types of OSPF routing, depending on whether the source and the destination are in the same or different areas. Intra-area routing occurs when the source and destination are in the same area; inter-area routing occurs when they are in different areas.

An OSPF backbone is responsible for distributing routing information between areas. It consists of all Area Border Routers, networks not wholly contained in any area, and their attached routers. Figure.27 shows an example of an internet with several areas. In the Fig. 7.3.7, routers 9, 10, 11, 12 and 13 make up the backbone. If host H1 in Area 3 wants to send a packet to host H2 in Area 1, the packet is sent to Router 4, which then forwards the packet along the backbone to

Area Border Router 12, which sends the packet to Router 11, and Router 11 forwards it to Router 10. Router 10 then sends the packet through an intra-area router (Router 3) to be forwarded to Host H2.

The backbone itself is an OSPF area, so all backbone routers use the same procedures and algorithms to maintain routing information within the backbone that any area router would. The backbone topology is invisible to all intra-area routers, as are individual area topologies to the backbone. Areas can be defined in such a way that the backbone is not contiguous. In this case, backbone connectivity must be restored through virtual links. Virtual links are configured between any backbone routers that share a link to a nonbackbone area and function as if they were direct links.

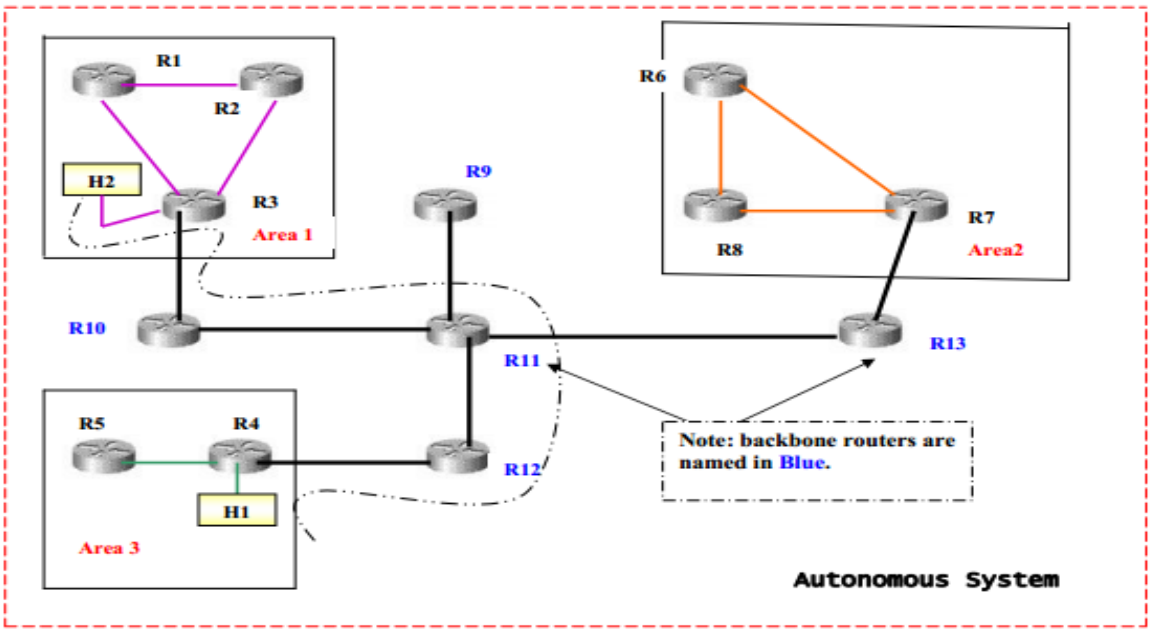


Figure.28 Different areas in autonomous systems

BGP (Boarder Gateway Protocol)

- BGP is an Exterior Gateway Protocol (EGP)- Performs routing between multiple autonomous systems or domains and exchanges routing and reachability information with other BGP systems.

- Border Gateway Protocol (BGP) is a standardized exterior gateway protocol designed to exchange routing and reachability information between autonomous systems (AS) on the Internet.The protocol is often classified as a path vector protocol but is sometimes also classed as a distance-vector routing protocol. The Border Gateway Protocol makes routing decisions based on paths, network policies, or rule-sets configured by a network administrator and is involved in making core routing decisions.
- It Performs inter-domain routing in Transmission-Control Protocol/Internet Protocol (TCP/IP) networks.

BGP is specified in several Request For Comments (RFCs):

- RFC 1771—Describes BGP4, the current version of BGP
- RFC 1654—Describes the first BGP4 specification
- RFC 1105, RFC 1163, and RFC 1267—Describes versions of BGP prior to BGP4.

The different versions of BGP range from 1–4; the industry standard is Version 4. You can, however, configure BGP Versions 2, 3, and 4 on a Cisco IOS router. The default standard is BGP Version 4 and is referred to as BGP4.

An autonomous system shares routing information with other autonomous systems using the Border Gateway Protocol (BGP). Previously, the Exterior Gateway Protocol (EGP) was used.

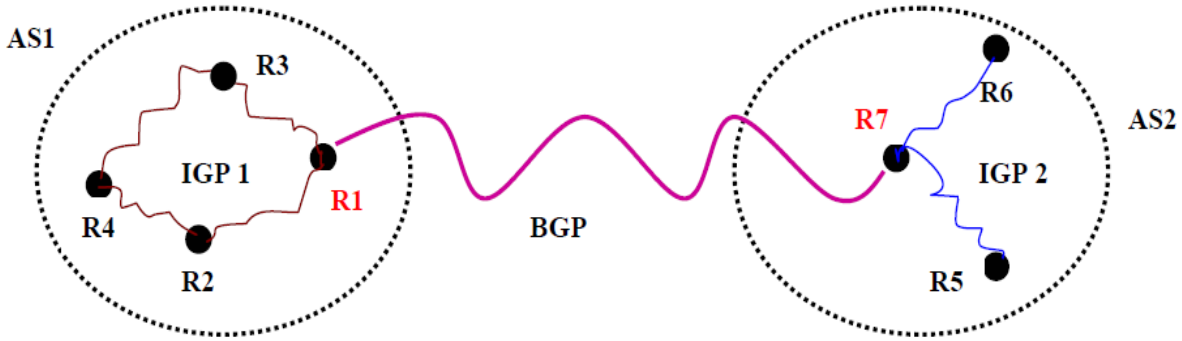


Figure.28 BGP

Two AS, each of which are using different IGPs internally and one BGP to communicate between each other.

BGP Characteristics

- *Inter-Autonomous System Configuration*: Provides communication between two autonomous systems.
- *Next-Hop paradigm*: Like RIP, BGP supplies next hop information for each destination.
- *Path information*:
    - BGP advertisements also include path information, along with the reachable destination and next destination pair, which allows a receiver to learn a series of autonomous system along the path to the destination.
    - BGP uses the path information to ensure the loop-free inter-domain routing.
- *Policy support*: Unlike most of the distance-vector based routing, BGP can implement policies that can be configured by the administrator
- *Runs over TCP*: BGP uses TCP for all communication. So the reliability issues are taken care by TCP.
- *Flexibility*: It can connect together any internetwork of autonomous systems using an arbitrary topology.
- *Conserve network bandwidth:*
    - BGP doesn't pass full information in each update message. Instead full information is just passed on once and thereafter successive messages only carries the incremental changes called deltas.
    - BGP also conserves bandwidth by allowing sender to aggregate route information and send single entry to represent multiple, related destinations.
- *Support for CIDR:* BGP supports classless addressing (CIDR). That it supports a way to send the network mask along with the addresses.
- *Security:* BGP allows a receiver to authenticate messages, so that the identity of the sender can be verified.

BGP Functionality and Route Information Management

It facilitates the exchange of route information between BGP devices, so that each router can determine efficient routes to each of the networks on an IP internetwork.

- *Route Storage:* Each BGP stores information about how to reach networks and holds routing information received from other devices.

- *Route Update:* When a BGP device receives an Update from one of its peers, it must decide how to use this information.
- *Route Selection*: Each BGP uses the information in its route databases to select good routes to each network on the internetwork.
- *Route Advertisement*: Each BGP speaker regularly tells its peers what it knows about various networks and methods to reach them by using BGP Update messages.

From Book correct mistakes typo error.

ICMP

The Internet Control Message Protocol (ICMP) has been designed only to report about the error or anymishappening occurred to the datagram ,since the IP doesn't take any care of flow and error control.

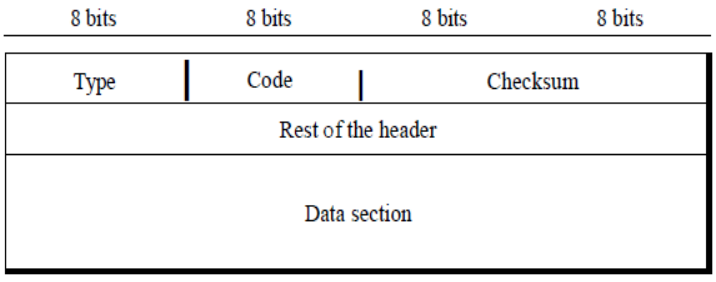
Types Of Message:-

ICMP messages are divided into two broad categories:

1. error-reporting messages:-The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.
2. query messages:- The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.

Message Format:-

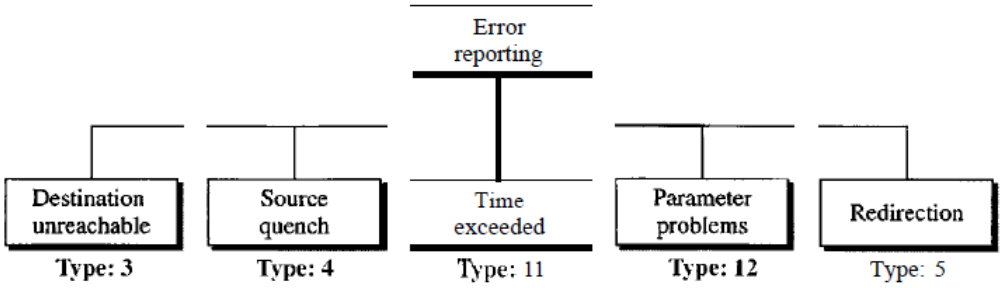
An ICMP message has an 8-byte header and a variable-size data section.Although the general format of the header is different for each message type, the first 4 bytes are common to all.



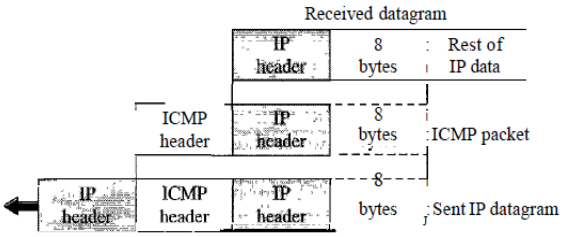
ICMP type, defines the type of themessage. The code field specifies the reason for the particular message type.The rest of theheader is specific for each message type.The data section in error messages carries information for finding the originalpacket that had the error. In query messages, the data section carries extra informationbased on the type of the query.

Error Reporting:-

- One of the main responsibilities of ICMP is to report errors. Although technology has produced increasingly reliable transmission media, errors still exist and must be handled.
- IP is an unreliable protocol. This means that error checking and error control are not a concern of IP. ICMP was designed, in part, to compensate for this shortcoming.
- However, ICMP does not correct errors-it simply reports them.
- Error correction is left to the higher-level protocols. Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses. ICMP uses the source IP address to send the error message to the source (originator) of the datagram.
- Five types of errors are handled: destination unreachable, source quench, time exceeded, parameter problems, and redirection



- The following are important points about ICMP error messages:
  1) No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
  2) No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
  3) No IeMP error message will be generated for a datagram having a multicastaddress.
  4) No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.
- All error messages contain a data section that includes the IP header of theoriginal datagram plus the first 8 bytes of data in that datagram.
- The original datagramheader is added to give the original source, which receives the error message, informationabout the datagram itself.
- UDP and TCP protocols, the first 8 bytes provide information aboutthe port numbers (UDP and TCP) and sequence number (TCP).
- This information isneeded so the source can inform the protocols (TCP or UDP) about the error.
- ICMPforms an error packet, which is then encapsulated in an IP datagram



[Contents of data field for the error messages]

Destination Unreachable

- When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.
- Destination-unreachable messages can be created by either a router or the destination host.

Source Quench

- The IP protocol is a connectionless protocol. There is no communication between the source host, which produces the datagram, the routers, which forward it, and the destination host, which processes it.
- One of the ramifications of this absence of communication is the lack of *flow control.* IP does not have a flow control mechanism embedded in the protocol. The lack of flow control can create a major problem in the operation of IP: congestion. The source host never knows if the routers or the destination host has been overwhelmed with datagrams.
- The source host never knows if it is producing datagrams faster than can be forwarded by routers or processed by the destination host.
- The lack of flow control can create congestion in routers or the destination host.
- A router or a host has a limited-size queue (buffer) for incoming datagrams waiting to be forwarded (in the case of a router) or to be processed (in the case of a host).
- If the datagrams are received much faster than they can be forwarded or processed, the queue may overflow. In this case, the router or the host has no choice but to discard some of the datagrams.
- The source-quench message in ICMP was designed to add a kind of flow control to the IP.
- When a router or host discards a datagram due to congestion,it sends a source-quench message to the sender of the datagram.
- This messagehas two purposes. First, it informs the source that the datagram has been discarded.Second, it warns the source that there is congestion somewhere in the path and that thesource should slow down (quench) the sending process.

Time Exceeded

- The time-exceeded message is generated in two cases: routersuse routing tables to find the next hop (next router) that must receive the packet. If thereare errors in one or more routing tables, a packet can travel in a loop or a cycle, goingfrom one router to the next or visiting a series of routers endlessly.
- Each datagram contains a field called *time to live* that controls this situation.

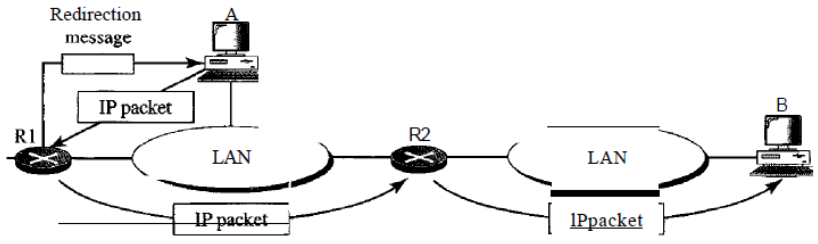
- When a datagram visits a router, the value of this field is decremented by 1. When thetime-to-live value reaches 0, after decrementing, the router discards the datagram. However,when the datagram is discarded, a time-exceeded message must be sent by therouter to the original source.
- Second, a time-exceeded message is also generated whennot all fragments that make up a message arrive at the destination host within a certaintime limit.

Parameter Problem

- Any ambiguity in the header part of a datagram can create serious problems as the datagram travels through the Internet.
- Ifa router or the destination host discovers an ambiguousor missing value in any field of the datagram, it discards the datagram and sends aparameter-problem message back to the source.

Redirection

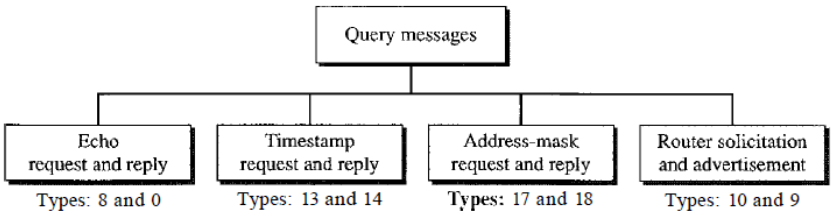
- When a router needs to send a packet destined for another network, it must know the IPaddress of the next appropriate router. The same is true if the sender is a host. Both routersand hosts, then, must have a routing table to find the address of the router or the nextrouter.
- Routers take part in the routing update process
- Hosts do not take part in the routing update process becausethere are many more hosts in an internet than routerS.
- Updating the routing tables of hostsdynamically produces unacceptable traffic.
- The hosts usually use static routing.
- When ahost comes up, its routing table has a limited number of entries. It usually knows the IPaddress of only one router, the default router.
- For this reason, the host may send a datagram,which is destined for another network, to the wrong router. In this case, the routerthat receives the datagram will forward the datagram to the correct router.
- However, toupdate the routing table of the host, it sends a redirection message to the host.



- Host A wants to send a datagram to host B. Router R2 is obviously the most efficient routing choice, but host A did not chooserouter R2.
- The datagram goes to R1 instead. Router R1, after consulting its table, findsthat the packet should have gone to R2. It sends the packet to R2 and, at the same time,sends a redirection message to host A.

Query

- ICMP can diagnose some network problems.
- This is accomplished through the query messages, a group of four different pairs of messages.
- In this type of ICMP message, a node sends a message thatis answered in a specific format by the destination node.
- A query message is encapsulated in an IP packet.



[query messages]

1) Echo Request and Reply

- The echo-request and echo-reply messages are designed for diagnostic purposes. Networkmanagers and users utilize this pair of messages to identify network problems.
- Thecombination of echo-request and echo-reply messages determines whether two systems(hosts or routers) can communicate with each other.
- The echo-request and echo-replymessages can be used to determine if there is communication at the IP level.

2) Timestamp Request and Reply

- Two machines (hosts or routers) can use the timestamp request and timestamp replymessages to determine the round-trip time needed for an IP datagram to travel betweenthem.
- It can also be used to synchronize the clocks in two machines.

3) Address-Mask Request and Reply

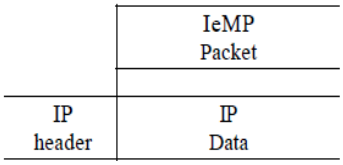
- A host may know its IP address, but it may not know the corresponding mask.
- To obtain its mask, a host sends an address-mask-requestmessage to a router on the LAN.
- Ifthe host knows the address of the router, it sends therequest directly to the router. If it does not know, it broadcasts the message.
- The routerreceiving the address-mask-request message responds with an address-mask-replymessage, providing the necessary mask for the host.
- This can be applied to its full IPaddress to get its subnet address.

4) Router Solicitation and Advertisement

- A host that wants to send data to a hoston another network needs to know the address of routers connected to its own network.Also, the host must know if the routers are alive and functioning.

- The router-solicitationand router-advertisement messages can help in this situation. A host can broadcast (ormulticast) a router-solicitation message.
- The router or routers that receive the solicitationmessage broadcast their routing information using the router-advertisement message. Arouter can also periodically send router-advertisement messages even if no host has solicited.
- when a router sends out an advertisement, it announces not only its ownpresence but also the presence of all routers on the network of which it is aware.

Encapsulation ofICMP query messages



5) Debugging Tools

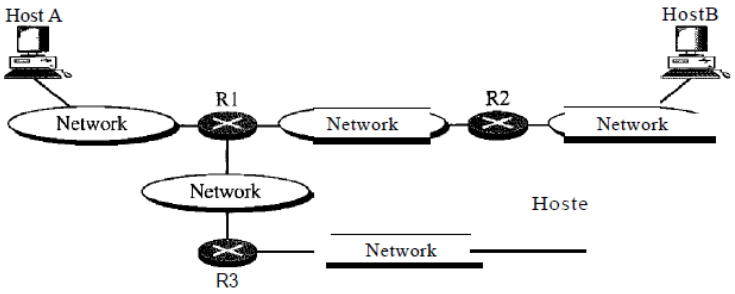
- There are several tools that can be used in the Internet for debugging. We can determinethe viability of a host or router.
- We can trace the route of a packet.
- We introduce twotools that use ICMP for debugging

a) Ping

- We can use the *ping* program to find if a host is alive and responding.
- The source host sends ICMP echo-request messages (type: 8, code: 0); the destination,if alive, responds with ICMP echo-reply messages.
- The *ping* program sets the identifierfield in the echo-request and echo-reply message and stans the sequence number from 0; this number is incremented by 1 each time a new message is sent.
- *Ping* can calculate the round-trip time. It inserts the sending time in the data section of the message.
- When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (RTT).

b) Traceroute

- The *traceroute* program in UNIX or *tracert*in Windows can be used to trace the route of a packet from the source to the destination.
- We have seen an application of the *traceroute* program to simulate the loose source route and strict source route options of an IP datagram .
- The program elegantly uses two ICMP messages, time exceeded and destination unreachable, to find the route of a packet. This is a program at the application level that uses the services of UDP



TRANSPORT LAYER

The transport layer is responsible for the delivery of a message from one process to another. 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 on one computer to a specific process on the other. The transport layer header must therefore include a type of address called a *service-point address* in the OSI model and port number or port addresses in the Internet and TCP/IP protocol suite.

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

In the transport layer, a message is normally divided into transmittable segments. A connectionless protocol, such as UDP, treats each segment separately. A connection oriented protocol, such as TCP and SCTP, creates a relationship between the segments using sequence numbers. Like the data link layer, the transport layer may be responsible for flow and error control. However, flow and error control at this layer is performed end to end rather than across a single link.

Process to process delivery:

The transport layer is responsible for process-to-process delivery. The delivery of a packet, part of a message, from one process to another. Two processes communicate in a client/server relationship.

Client/Server Paradigm:

Although there are several ways to achieve process-to-process communication, the most common one is through the client/server paradigm. A process on the local host, called a client, needs services from a process usually on the remote host, called a server. Both processes (client and server) have the same name. For example, to get the day and time from a remote machine, we need a Daytime client process running on the local host and a Daytime server process running on a remote machine. Operating systems today support both multiuser and multiprogramming environments. A remote computer can run several server programs at the same time, just as local computers can run one or more client programs at the same time. For communication, we must define the following:

1. Local host

2. Local process
3. Remote host
4. Remote process

Addressing:

Whenever we need to deliver something to one specific destination among many, we need an address.

At the transport layer, we need a transport layer address, called a port number, to choose among multiple processes running on the destination host. The destination port number is needed for delivery; the source port number is needed for the reply. In the Internet model, the port numbers are 16-bit integers between 0 and 65,535.

The client program defines itself with a port number, chosen randomly by the transport layer software running on the client host. This is the ephemeral port number.

The server process must also define itself with a port number. This port number, however, cannot be chosen randomly. If the computer at the server site runs a server process and assigns a random number as the port number, the process at the client site that wants to access that server and use its services will not know the port number. Of course, one solution would be to send a special packet and request the port number of a specific server, but this requires more overhead. The Internet has decided to use universal port numbers for servers; these are called well-known port numbers. There are some exceptions to this rule; for example, there are clients that are assigned well-known port numbers. Every client process knows the well-known port number of the corresponding server process. For example, while the Daytime client process, discussed above, can use an ephemeral (temporary) port number 52,000 to identify itself, the Daytime server process must use the well-known (permanent) port number 13.

lANA Ranges

The lANA (Internet Assigned Number Authority) has divided the port numbers into three ranges: well known, registered, and dynamic or private.

* Well-known ports. The ports ranging from 0 to 1023 are assigned and controlled by lANA. These are the well-known ports.
* Registered ports. The ports ranging from 1024 to 49,151 are not assigned or controlled by lANA. They can only be registered with lANA to prevent duplication.
* Dynamic ports. The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used by any process. These are the ephemeral ports.

Socket Addresses:

Process-to-process delivery needs two identifiers, IP address and the port number, at each end to make a connection. The combination of an IP address and a port number is called a socket address. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely a transport layer protocol needs a pair of socket addresses: the client socket address and the server socket address. These four pieces of information are part of the IP header and the transport layer protocol header. The IP header contains the IP addresses; the UDP or TCP header contains the port numbers.

Multiplexing and Demultiplexing:

The addressing mechanism allows multiplexing and demultiplexing by the transport layer.

Multiplexing:

At the sender site, there may be several processes that need to send packets. However, there is only one transport layer protocol at any time. This is a many-to-one relationship and requires multiplexing. The protocol accepts messages from different processes, differentiated by their assigned port numbers. After adding the header, the transport layer passes the packet to the network layer.

Demultiplexing:

At the receiver site, the relationship is one-to-many and requires demultiplexing. The transport layer receives datagrams from the network layer. After error checking and dropping of the header, the transport layer delivers each message to the appropriate process based on the port number.

Connectionless Versus Connection-Oriented Service:

A transport layer protocol can either be connectionless or connection-oriented.

Connectionless Service:

In a connectionless service, the packets are sent from one party to another with no need for connection establishment or connection release. The packets are not numbered; they may be delayed or lost or may arrive out of sequence. There is no acknowledgment either. We will see shortly that one of the transport layer protocols in the Internet model, UDP, is connectionless.

Connection Oriented Service:

In a connection-oriented service, a connection is first established between the sender and the receiver. Data are transferred. At the end, the connection is released. We will see shortly that TCP and SCTP are connection-oriented protocols.

Reliable Versus Unreliable:

The transport layer service can be reliable or unreliable. If the application layer program needs reliability, we use a reliable transport layer protocol by implementing flow and error control at the transport layer. This means a slower and more complex service. On the other hand, if the application program does not need reliability because it uses its own flow and error control mechanism or it needs fast service or the nature of the service does not demand flow and error control (real-time applications), then an unreliable protocol can be used. In the Internet, there are three common different transport layer protocols, as we have already mentioned. UDP is connectionless and unreliable; TCP and SCTP are connection oriented and reliable. These three can respond to the demands of the application layer

programs.

USER DATAGRAM PROTOCOL (UDP):

The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to process communication instead of host-to-host communication. Also, it performs very limited error checking.UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about

reliability, it can use UDP. Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP or SCTP.

User Datagram:

UDP packets, called user datagram, have a fixed-size header of 8 bytes.

The fields are as follows:

-Source port number. This is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 to 65,535. If the source host is the client (a client sending a request), the port number, in most cases, is an ephemeral port number requested by the process and chosen by the UDP software running on the source host. If the source host is the server (a server sending a response), the port number, in most cases, is a well-known port number.

- Destination port number. This is the port number used by the process running on the destination host. It is also 16 bits long. If the destination host is the server (a client sending a request), the port number, in most cases, is a well-known port number. If the destination host is the client (a server sending a response), the port number, in most cases, is an ephemeral port number. In this case, the server copies the ephemeral port number it has received in the request packet.

-Length. This is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be much less because a UDP user datagram is stored in an IP datagram with a total length of 65,535 bytes.

The length field in a UDP user datagram is actually not necessary. A user datagram is encapsulated in an IP datagram. There is a field in the IP datagram that defines the total length. There is another field in the IP datagram that defines the length of the header. So if we subtract the value of the second field from the first, we can deduce the length of a UDP datagram that is encapsulated in an IP datagram.

-Checksum. This field is used to detect errors over the entire user datagram (header plus data). The checksum is discussed next.

UDP Operation:

UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program. The user datagrams are not numbered. Also, there is no connection establishment and no connection termination, as is the case for TCP. This means that each user datagram can travel on a different path. One of the ramifications of being connectionless is that the process that uses UDP cannot send a stream of data to UDP and expect UDP to chop them into different related user datagrams. Instead each request must be small enough to fit into one user datagram. Only those processes sending short messages should use UDP.

Flow and Error Control:

UDP is a very simple, unreliable transport protocol. There is no flow control and hence no window mechanism. The receiver may overflow with incoming messages. There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded. The lack

of flow control and error control means that the process using UDP should provide these mechanisms.

Queuing:

In UDP, queues are associated with ports.

- At the client site, when a process starts, it requests a port number from the operating system. Some implementations create both an incoming and an outgoing queue associated with each process. Other implementations create only an incoming queue associated with each process.
- Note that even if a process wants to communicate with multiple processes, it obtains only one port number and eventually one outgoing and one incoming queue. The queues opened by the client are, in most cases, identified by ephemeral port numbers. The queues function as long as the process is running. When the process terminates, the queues are destroyed.
- The client process can send messages to the outgoing queue by using the source port number specified in the request. UDP removes the messages one by one and, after adding the UDP header, delivers them to IP. An outgoing queue can overflow. If this happens, the operating system can ask the client process to wait before sending any more messages.
- When a message arrives for a client, UDP checks to see if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue. If there is no such queue, UDP discards the user datagram and asks the ICMP protocol to send a *port unreachable* message to the server. All the incoming messages for one particular client program, whether coming from the same or a different server, are sent to the same queue. An incoming queue can overflow. If this happens, UDP drops the user datagram and asks for a port unreachable message to be sent to the server.
- At the server site, the mechanism of creating queues is different. In its simplest form, a server asks for incoming and outgoing queues, using its well-known port, when it starts running. The queues remain open as long as the server is running.
- When a message arrives for a server, UDP checks to see if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue. If there is no such queue, UDP discards the user datagram and asks the ICMP protocol to send a port unreachable message to the client. All the incoming messages for one particular server, whether coming from the same or a different client, are sent to the same queue. An incoming queue can overflow. If this happens, UDP drops the user datagram and asks for a port unreachable message to be sent to the client.
- When a server wants to respond to a client, it sends messages to the outgoing queue, using the source port number specified in the request. UDP removes the messages one by one and, after adding the UDP header, delivers them to IP. An outgoing queue can overflow. If this happens, the operating system asks the server to wait before sending any more messages.

Use of UDP:

The following lists some uses of the UDP protocol:

- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as FrP that needs to send bulk data (see Chapter 26).
- UDP is suitable for a process with internal flow and error control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control. It can easily use UDP.
- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- UDP is used for management processes such as SNMP .
- UDP is used for some route updating protocols such as Routing Information Protocol (RIP).

TCP:

Transmission Control Protocol (TCP) is called a *connection-oriented, reliable* transport protocol. It adds Connection-oriented and reliability features to the services of IP.

Stream Delivery Service:

TCP, unlike UDP, is a stream-oriented protocol. In UDP, a process (an application program) sends messages, with predefined boundaries, to UDP for delivery. UDP adds its own header to each of these messages and delivers them to IP for transmission. Each message from the process is called a user datagram and becomes, eventually, one IP datagram. Neither IP nor UDP recognizes any relationship between the datagrams. TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes. TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their data across the Internet. The sending process produces (writes to) the stream of bytes, and the receiving process consumes (reads from) them.

Sending and Receiving Buffers Because the sending and the receiving processes may not write or read data at the same speed, TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer.

Full-Duplex Communication:

TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions.

Connection-Oriented Service:

TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

1. The two TCPs establish a connection between them.
2. Data are exchanged in both directions.
3. The connection is terminated.

Note that this is a virtual connection, not a physical connection. The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost, or corrupted, and then resent. Each may use a different path to reach the destination.

There is no physical connection. TCP creates a stream-oriented environment in which it accepts the responsibility of delivering the bytes in order to the other site. The situation is similar to creating a bridge
that spans multiple islands and passing all the bytes from one island to another in one single connection. We will discuss this feature later in the chapter.

Flow Control:

TCP, unlike UDP, provides *flow control.* The receiver of the data controls the amount of data that are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

Error Control:

To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented.

Congestion Control:

TCP, unlike UDP, takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network.

Segment:

Before we discuss TCP in greater detail, let us discuss the TCP packets themselves. A packet in TCP is called a segment.

The segment consists of a 20- to 60-byte header, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options. We will discuss some of the header fields in this section. The meaning and purpose of these will become clearer as we proceed through the chapter.

- Source port address. This is a 16-bit field that defines the port number of the application program in the host that is sending the segment. This serves the same purpose as the source port address in the UDP header.
- Destination port address. This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment. This serves the same purpose as the destination port address in the UDP header.
- Sequence number. This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment. During connection establishment, each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.
- Acknowledgment number. This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number *x* from the other party, it defines *x* + I as the acknowledgment number. Acknowledgment and data can be piggybacked together.

A TCP Connection:

TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All the segments belonging to a

message are then sent over this virtual path. Using a single virtual pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.

In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

Connection Establishment:

TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously. This implies that each party must initialize communication and get approval from the other party before any data are transferred. Three-Way Handshaking The connection establishment in TCP is called three way handshaking. In our example, an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport layer protocol. The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This is called a request for a *passive open.* Although the server TCP is ready to accept any connection from any machine in the world, it cannot make the connection itself. The client program issues a request for an *active open.* A client that wishes to connect to an open server tells its TCP that it needs to be connected to that particular server. TCP can now start the three-way handshaking process.

The three steps in this phase are as follows:-

1. A SYN segment cannot carry data, but it consumes one sequence number.
2. A SYN +ACK segment cannot carry data, but does consume one sequence number.
3. An ACK segment, if carrying no data, consumes no sequence number.

Data Transfer:

After connection is established, bidirectional data transfer can take place. The client and server can both send data and acknowledgments. Data traveling in the same direction as an acknowledgment are carried on the same segment. The acknowledgment is piggybacked with the data.

Connection Termination:

Any of the two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client. Most implementations today allow two options for connection termination: three-way handshaking and four-way handshaking with a half-close option. Three-Way Handshaking Most implementations today allow *three-way handshaking* for connection termination.

1. The FIN segment consumes one sequence number ifit does not carry data.
2. The FIN +ACK segment consumes one sequence number if it does not carry data.
3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgment number, which is 1 plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

Some points about TCP sliding windows:

- The size of the window is the lesser of *rwnd* and *cwnd.*
- The source does not have to send a full window's worth of data.
- The window can be opened or closed by the receiver, but should not be shrunk.

- The destination can send an acknowledgment at any time as long as it does not result in a shrinking window.
- The receiver can temporarily shut down the window; the sender, however, can always send a segment of 1 byte after the window is shut down.

Error Control:

TCP is a reliable transport layer protocol. This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without any part lost or duplicated.

TCP provides reliability using error control. Error control includes mechanisms for detecting corrupted segments, lost segments, out-of-order segments, and duplicated segments. Error control also includes a mechanism for correcting errors after they are detected. Error detection and correction in TCP is achieved through the use of three simple tools: checksum, acknowledgment, and time-out.

Checksum:

Each segment includes a checksum field which is used to check for a corrupted segment. If the segment is corrupted, it is discarded by the destination TCP and is considered as lost. TCP uses a 16-bit checksum that is mandatory in every segment. In that the 16-bit checksum is considered inadequate for the new transport layer, SCTP. However, it cannot be changed for TCP because this would involve reconfiguration of the entire header format.

Acknowledgment:

TCP uses acknowledgments to confirm the receipt of data segments. Control segments that carry no data but consume a sequence number are also acknowledged. ACK segments are never acknowledged.

CONGESTION CONTROL:

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion control (removal).

Congestion Control

Open-loop	Closed-loop
1. *Retransmission Policy*	1. *Backpressure*
2. *Window Policy*	2. *Choke Packet*
3. *Acknowledgment Policy*	3. *Implicit Signaling*

4. *Discarding Policy* 4. *Explicit Signaling*

5. *Admission Policy*

Open-Loop Congestion Control:

In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination. We give a brief list of policies that can prevent congestion.

1. **Retransmission Policy:**

Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion. For example, the retransmission policy used by TCP (explained later) is designed to prevent or alleviate congestion.

2. **Window Policy:**

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the *Go-Back-N* window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse. The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.

3. **Acknowledgment Policy:**

The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion. Several approaches are used in this case. A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires. A receiver may decide to acknowledge only *N* packets at a time. We need to know that the acknowledgments are also part of the load in a network. Sending fewer acknowledgments means imposing less load on the network.

4. **Discarding Policy:**

A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission. For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

5. **Admission Policy:**

An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks. Switches in a flow first check the resource requirement of a flow before admitting it to the network. A router can deny establishing a virtual circuit connection if there is congestion in the network or if there is a possibility of future congestion Closed-Loop Congestion Control: Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols. We describe a few of them here.

a. **Backpressure:**

The technique of *backpressure* refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream nodes or nodes. And so on. Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is corning.

b. 2.Choke Packet:

A choke packet is a packet sent by a node to the source to inform it of congestion. Note the difference between the backpressure and choke packet methods. In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station. In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned. It informs the source host, using a source quench ICMP message. The warning message goes directly to the source station; the intermediate routers, and does not take any action.

3.Implicit Signaling*:*

In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is a congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested. The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down.

4.Explicit Signaling*:*

The node that experiences congestion can explicitly send a signal to the source destination. The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data. Explicit signaling, as we will see in Frame Relay congestion control, can occur in either the forward or the backward direction. Backward Signaling A bit can be set in a packet moving in the direction opposite to the congestion.

This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets. Forward Signaling A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

QUALITY OF SERVICE:

We can informally define quality of service as something a flow seeks to attain.

Flow Characteristics:

Traditionally, four types of characteristics are attributed to a flow:

1. reliability
2. delay

3. jitter
4. Bandwidth

Reliability:

Reliability is a characteristic that a flow needs. Lack of reliability means losing a packet or acknowledgment, which entails retransmission. However, the sensitivity of application programs to reliability is not the same. For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.

1. **Delay:**

   Source-to-destination delay is another flow characteristic. Again applications can tolerate delay in different degrees. In this case, telephony, audio conferencing, video conferencing, and remote log-in need minimum delay, while delay in file transfer or e-mail is less important.

2. **Jitter:**

   Jitter is the variation in delay for packets belonging to the same flow. For example, if four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time. On the other hand, if the above four packets arrive at 21, 23, 21, and 28, they will have different delays: 21,22, 19, and 24.

3. **Bandwidth:**

   Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million. Techniques To Improve QOS: we discuss some techniques that can be used to improve the quality of service. We briefly discuss three common methods: scheduling, admission control, and resource reservation.

Scheduling:

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Several scheduling techniques are designed to improve the quality of service. We discuss three of them here: FIFO queuing, priority queuing, and weighted fair queuing.

FIFO Queuing:

In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them. If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded. A FIFO queue is familiar to those who have had to wait for a bus at a bus stop.

Priority Queuing:

In priority queuing, packets are first assigned to a priority class. Each priority class has its own queue. The packets in the highest-priority queue are processed first. Packets in the lowest-priority queue are processed last. Note that the system does not stop serving a queue until it is empty.

Weighted Fair Queuing:

A better scheduling method is weighted fair queuing. In this technique, the packets are still assigned to different classes and admitted to different queues.

Resource Reservation:

A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The quality of service is improved if these resources are reserved beforehand. We discuss in this section one QoS model called Integrated Services, which depends heavily on resource reservation to improve the quality of service.

Admission Control:

Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

Host to Host Delivery: Internetworking, addressing and Routing Network Layer Protocols: ARP, IPV4, ICMP, IPV6 ad ICMPV6 Transport Layer: Process to Process Delivery: UDP; TCP congestion control and Quality of service.