Unit 4: The Network Layer

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links) i.e., it ensures that each packet gets from its point of origin to its final destination. The network layer is considered the backbone of the OSI Model. It selects and manages the best logical path for data transfer between nodes. The routing information contained within a packet includes the source of the sending host and the eventual destination of the remote host. This information is contained within the network layer header that encapsulates network frames at the data link layer. The primary function of the network layer is to permit different networks to be interconnected. It does this by forwarding packets to network routers, which rely on algorithms to determine the best paths for the data to travel. The network layer can support either connection-oriented or connectionless networks, but such a network can only be of one type and not both.

Virtual Circuits:

A virtual circuit (VC) is a means of transporting data over a packet switched computer network in such a way that it appears as though there is a dedicated physical layer link between the source and destination end systems of this data. In all major computer network architectures to date (Internet, ATM, frame relay, and so on), the network layer provides either a host-to-host connectionless service or a host-to-host connection service, but not both. Computer networks that provide only a connection service at the network layer are called **virtual-circuit** (VC) networks; computer networks that provide only a connectionless service at the network layer are called datagram networks. While the Internet is a datagram network, many alternative network architectures— including those of ATM and frame relay— are virtual-circuit networks and, therefore, use connections at the network layer. These network-layer connections are called **virtual circuits (VCs)**. A VC consists of:

(1) A path (that is, a series of links and routers) between the source and destination hosts,
(2) VC numbers, one number for each link along the path, and
(3) Entries in the forwarding table in each router along the path.
A packet belonging to a virtual circuit will carry a VC number in its header. Because a virtual circuit may have a different VC number on each link, each intervening router must replace the VC number of each traversing packet with a new VC number. The new VC number is obtained from the forwarding table.

There are three identifiable phases in a virtual circuit:

- VC Setup: During this setup phase, the sending transport layer contacts the network layer, specifies the receiver's address, and waits for the network to set up the VC. The network layer determines the path between sender and receiver, that is, the series of links and routers through which all packets of the VC will travel. The network layer also determines the VC number for each link along the path. Finally, the network layer adds an entry in the forwarding table in each router along the path. During VC setup, the network layer may also reserve resources (for example, bandwidth) along the path of the VC.
- Data Transfer: As shown in the figure below, once the VC has been established, packets can begin to flow along the VC.
- VC Teardown: This is initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC. The network layer will then typically inform the end system on the other side

of the network of the call termination and update the forwarding table sin each of the packet routers on the path to indicate that the VC no longer exists.

To illustrate the concept, consider the network shown in the figure. The numbers next to the links of R1 in figure are the link interface numbers. Suppose now that Host A requests that the network establish a VC between itself and Host B. Suppose also that the network chooses the path A-R1-R2-B and assigns VC numbers 12, 22, and 32 to the three links in this path for this virtual circuit. In this case, when a packet in this VC leaves Host A, the value in the VC number field in the packet header is 12; when it leaves R1, the value is 22; and when it leaves R2, the value is 32.

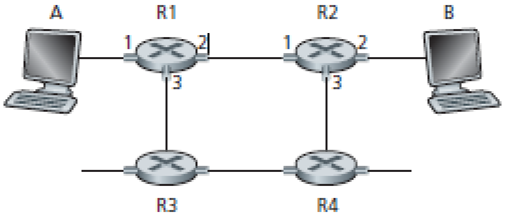


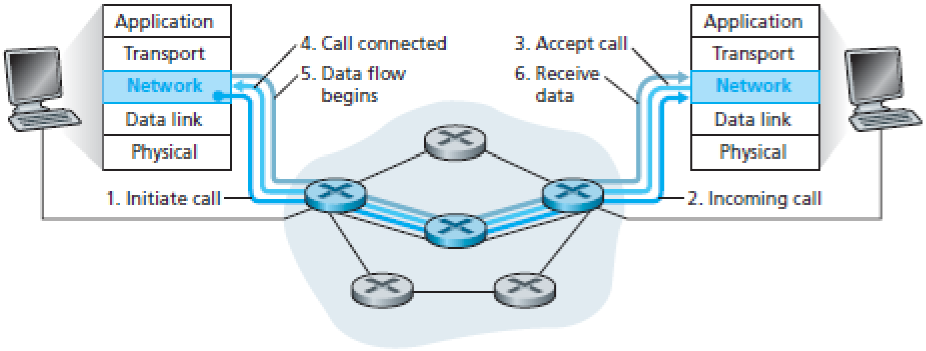
Fig: A simple virtual circuit network

VC forwarding table:

For a VC network, each router's forwarding table includes VC number translation; for example, the forwarding table in R1 might look something like this:

Incoming interface	**Incoming VC #**	**Outgoing interface**	**Outgoing VC #**
1	**12**	**2**	**22**
2	**63**	**1**	**18**
3	**7**	**2**	**17**
1	**97**	**3**	**87**
…	**…**	**…**	**…**

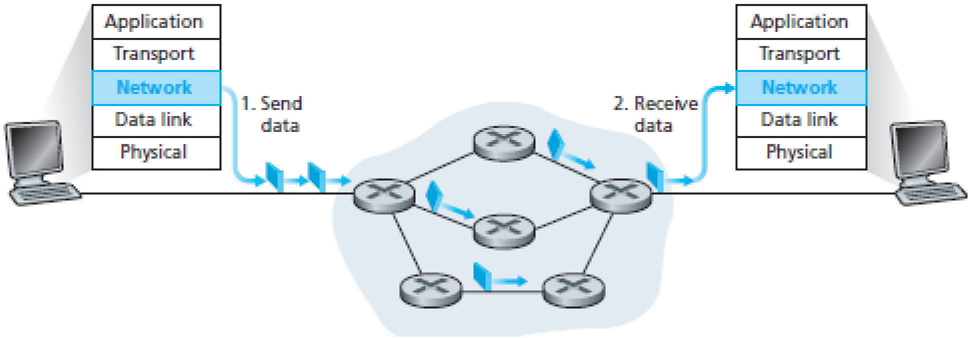
Whenever a new VC is established across a router, an entry is added to the forwarding table. Similarly, whenever a VC terminates, the appropriate entries in each table along its path are removed.



Datagram Subnet:

The connectionless services at the network layer are called datagram networks. Connectionless service means that a terminal or node can send data packets to its destination without establishing a connection to the destination. A session connection between the sender and the receiver is not required, the sender just starts sending the data. The message or datagram is sent without prior arrangement, which is less reliable but faster transaction than a connection-oriented service. This works because of error handling protocols, which allow for error correction like requesting retransmission. It is similar to the postal services, as it carries the full address where the message (letter) is to be carried. Each message is routed independently from source to destination. The order of message sent can be different from the order received.

In a datagram network, each time an end system wants to send a packet, it stamps the packet with the address of the destination end system and then pops the packet into the network. As shown in Figure, there is no VC setup and routers do not maintain any VC state information (because there are no VCs).



Internet Protocol:

The Internet Protocol (IP) is the method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP address that uniquely identifies it from all other computers on the Internet. When you send or receive data (for example, an e-mail note or a Web page), the message gets divided into little chunks called packets. Each of these packets contains both the sender's Internet address and the receiver's address. Any packet is sent first to a gateway computer that understands a small part of the Internet. The gateway computer reads the destination address and forwards the packet to an adjacent gateway that in turn reads the destination address and so forth across the Internet until one gateway recognizes the packet as belonging to a computer within its immediate neighborhood or domain. That gateway then forwards the packet directly to the computer whose address is specified.

IP Address:

An Internet Protocol address (IP address) is a logical numeric address that is assigned to every single computer, printer, switch, router or any other device that is part of a TCP/IP-based network. The IP address is the core component on which the networking architecture is built; no network exists without it. An IP address is a logical address that is used to uniquely identify every node in the network. Because

IP addresses are logical, they can change. They are similar to addresses in a town or city because the IP address gives the network node an address so that it can communicate with other nodes or networks.

The numerals in an IP address are divided into 2 parts:

- The network part specifies which networks this address belongs to and
- The host part further pinpoints the exact location.

IPv4:

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device to the internet. If a device operating in the network layer has m connections, then it needs to have m addresses. Router is such type of device.

An IPV4 address consists of 4 bytes in the form a.b.c.d (E.g. 173.14.2.225, 11.12.13.3). It can be logically divided into a network and a host portion. While the network portion identifies the network to which the end node belongs to, the host portion uniquely identifies the end node, from the other end nodes, inside the network.

Binary Format	**Dotted Decimal Notation**
11000000 10101000 00000011 00011000	192.168.3.24

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

IPv4 ADDRESSING SCHEME

IP addresses falls into two types:

- Classful IP addressing is a legacy scheme which divides the whole IP address pools into 5 distinct classes—A, B, C, D and E.
- Classless IP addressing has an arbitrary length of the prefixes.

Classful Addressing

Class A

The first octet denotes the network address, and the last three octets are the host portion. Any IP address whose first octet is between 1 and 126 is a Class A address. Note that 0 is reserved as a part of the default address and 127 is reserved for internal loopback testing.

Format: network.host.host.host

Default subnet mask = 255.0.0.0 or (slash notation) /8

Class B

The first two octets denote the network address, and the last two octets are the host portion. Any address whose first octet is in the range 128 to 191 is a Class B address.

Format: network.network.host.host

Default subnet mask =255.255.0.0 or /16

Class C

The first three octets denote the network address, and the last octet is the host portion. The first octet range of 192 to 223 is a Class C address.

Format: network.network.network.host

Default subnet mask = 255.255.255.0 or /24

Class D

Used for multicast. Multicast IP addresses have their first octets in the range 224 to 239.

Class E

Reserved for future use or research purpose and includes the range of addresses with a first octet from 240 to 255.

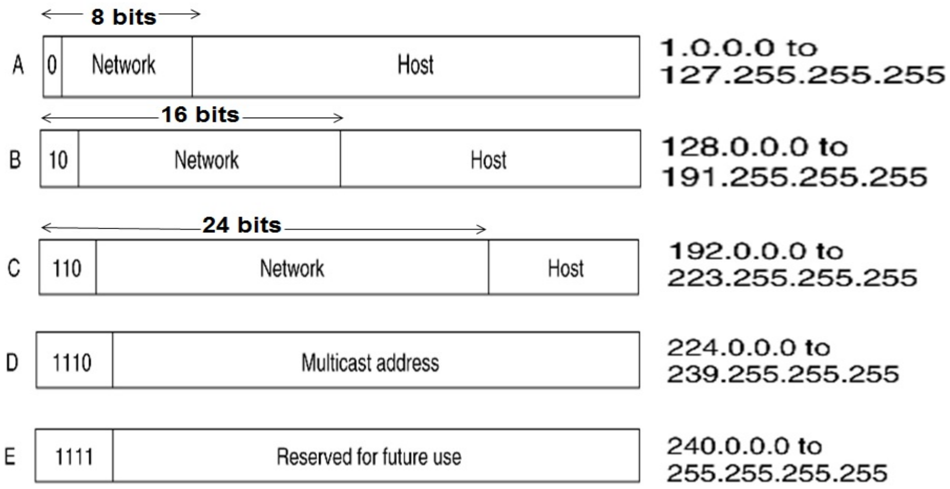
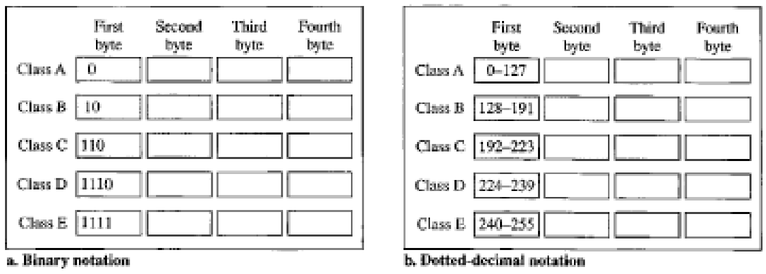


Figure 19.2 *Finding the classes in binary and dotted-decimal notation*

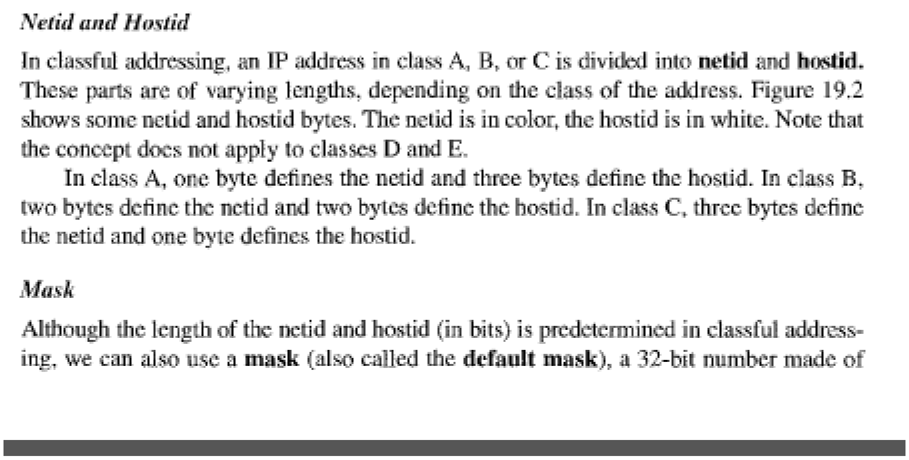


	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

a. Binary notation

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

b. Dotted-decimal notation



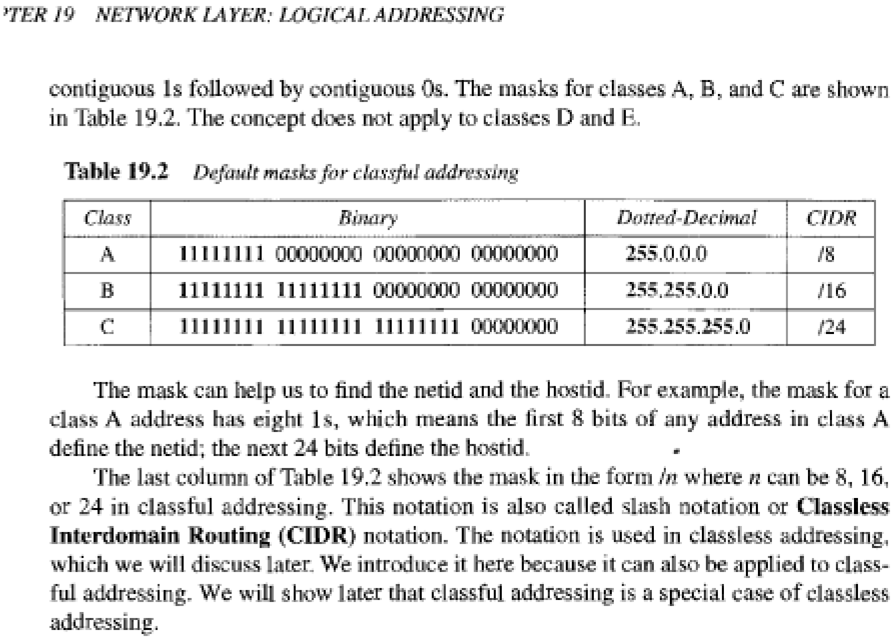
Netid and Hostid

In classful addressing, an IP address in class A, B, or C is divided into **netid** and **hostid.** These parts are of varying lengths, depending on the class of the address. Figure 19.2 shows some netid and hostid bytes. The netid is in color, the hostid is in white. Note that the concept does not apply to classes D and E.

In class A, one byte defines the netid and three bytes define the hostid. In class B, two bytes define the netid and two bytes define the hostid. In class C, three bytes define the netid and one byte defines the hostid.

Mask

Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a **mask** (also called the **default mask**), a 32-bit number made of



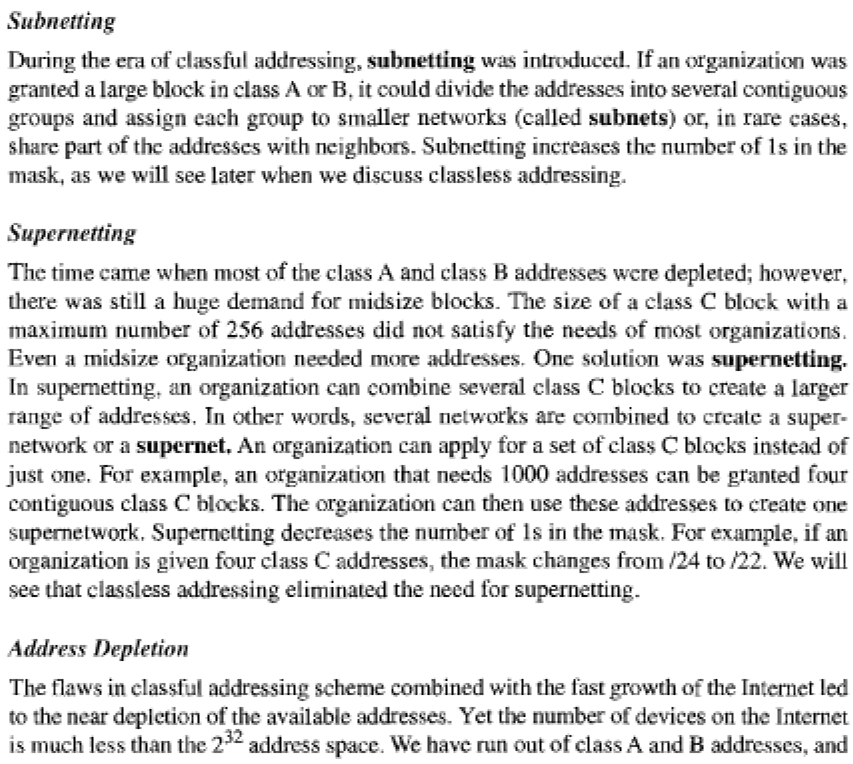
contiguous 1s followed by contiguous 0s. The masks for classes A, B, and C are shown in Table 19.2. The concept does not apply to classes D and E.

Table 19.2 *Default masks for classful addressing*

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	**11111111** 00000000 00000000 00000000	255.0.0.0	/8
B	**11111111 11111111** 00000000 00000000	255.255.0.0	/16
C	**11111111 11111111 11111111** 00000000	255.255.255.0	/24

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

The last column of Table 19.2 shows the mask in the form */n* where *n* can be 8, 16, or 24 in classful addressing. This notation is also called slash notation or **Classless Interdomain Routing (CIDR)** notation. The notation is used in classless addressing, which we will discuss later. We introduce it here because it can also be applied to classful addressing. We will show later that classful addressing is a special case of classless addressing.



Subnetting

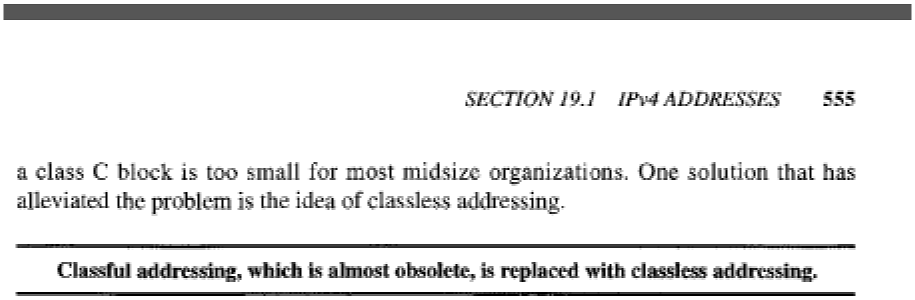
During the era of classful addressing, **subnetting** was introduced. If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called **subnets**) or, in rare cases, share part of the addresses with neighbors. Subnetting increases the number of 1s in the mask, as we will see later when we discuss classless addressing.

Supernetting

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

Address Depletion

The flaws in classful addressing scheme combined with the fast growth of the Internet led to the near depletion of the available addresses. Yet the number of devices on the Internet is much less than the 2^{32} address space. We have run out of class A and B addresses, and



a class C block is too small for most midsize organizations. One solution that has alleviated the problem is the idea of classless addressing.

Classful addressing, which is almost obsolete, is replaced with classless addressing.

Classless IP addresses

Classful IP addresses is no longer popular and instead has been replaced with the concept of classless IP address, where there is no concept of IP address classes and no strict network and host boundaries. In classless IP addressing, there is no concept of Classful addressing like Classes A, B, C, D and E. IPv4 address range 0.0.0.0 to 223.255.255.255 treated as a single class. No strict 8-byte boundaries for the network and host portions. A Subnet masks defines network & host boundaries. This approach is very useful for optimizing address usage.

Examples of Classless Addressing

Network address – 22.10.0.0 /16

Network address – 173.2.224.0 / 21

In the above address, 16 and 21 denote the subnet masks respectively. This means that in the first address 22.10.0.0, the first 16 bits are reserved for the network portion and the rest of the 16 bits are reserved for the host portion. Similarly, in the second address 173.2.224.0/21, the first 21 bits are reserved for the network portion and the remaining 13 bits are reserved for the host portion. Thus, it can be seen that classless addressing gives a flexible boundary between the network and host portions, thereby allowing lot of flexibility in partitioning the networks.

Subnetting:

Subnetting is the practice of dividing a network into two or more smaller networks. The major advantage of subnetting is to reduce the address wastage. It increases routing efficiency, enhances the security of the network and reduces the size of the broadcast domain.

The reasons to use subnetting are:

- Conservation of IP addresses
- Reduced network traffic
- Simplified troubleshooting

FLSM vs VLSM:

FLSM stands for Full Length Subnet Mask. It means all the subnets are of the same size. In FLSM, the subnet mask remains the same for all the subnets.

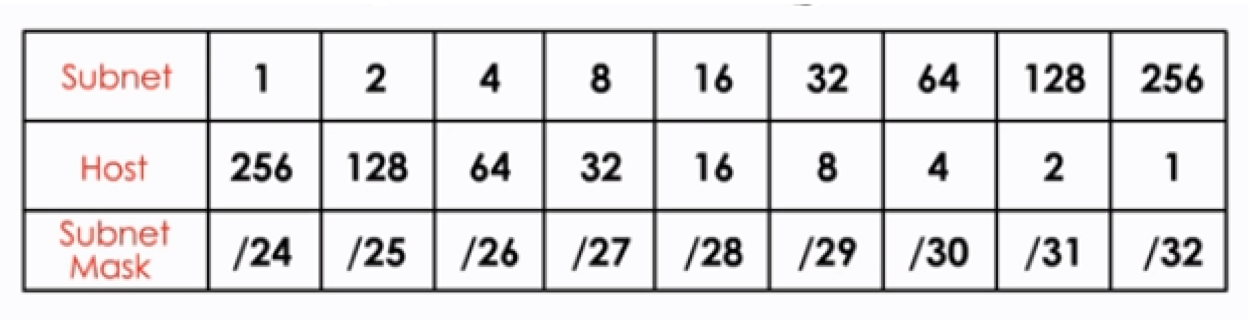
VLSM stands for Variable Length Subnet Mask. It means the size of the subnet varies according to the needs. In VLSM, the subnet mask is different normally but it can be same for any two or more subnets depending upon the situation.

Internet Service Providers may face a situation where they need to allocate IP subnets of different sizes as per the requirement of customer. One customer may ask Class C subnet of 3 IP addresses and another may ask for 100 IPs. For an ISP, it is not feasible to divide the IP addresses into fixed size subnets, rather he may want to subnet the subnets in such a way which results in minimum wastage of IP addresses.

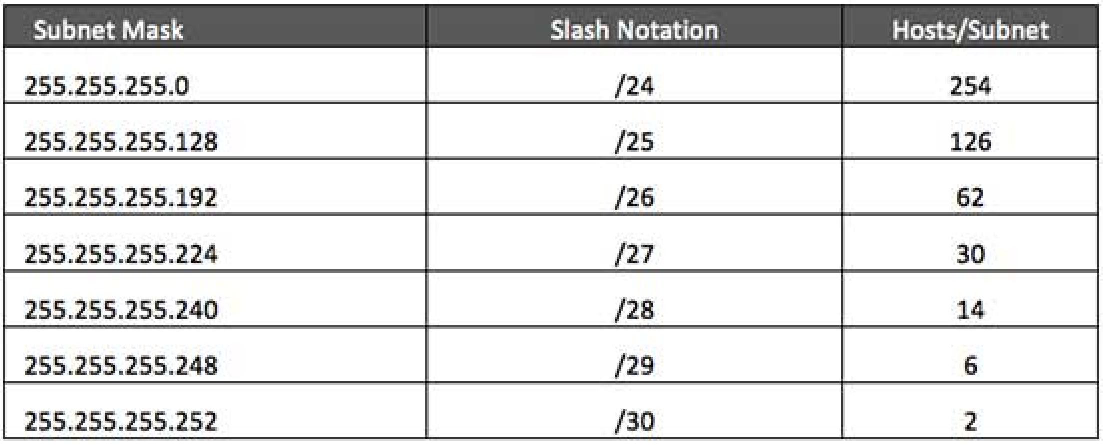
For example, an administrator has 192.168.1.0/24 network. The suffix /24 (pronounced as "slash 24") tells the number of bits used for network address. In this example, the administrator has three different departments with different number of hosts. Sales department has 100 computers, Purchase department has 50 computers, Accounts has 25 computers and Management has 5 computers. In CIDR, the subnets are of fixed size. Using the same methodology, the administrator cannot fulfill all the requirements of the network.

The following procedure shows how VLSM can be used in order to allocate department-wise IP addresses as mentioned in the example.

Step 1: Make a list of Subnets possible.



Subnet	1	2	4	8	16	32	64	128	256
Host	256	128	64	32	16	8	4	2	1
Subnet Mask	/24	/25	/26	/27	/28	/29	/30	/31	/32



Subnet Mask	Slash Notation	Hosts/Subnet
255.255.255.0	/24	254
255.255.255.128	/25	126
255.255.255.192	/26	62
255.255.255.224	/27	30
255.255.255.240	/28	14
255.255.255.248	/29	6
255.255.255.252	/30	2

Step 2: Sort the requirements of IPs in descending order (Highest to Lowest).

Sales 100

Purchase 50

Accounts 25

Management 5

Step 3: Allocate the highest range of IPs to the highest requirement, so let's assign 192.168.1.0 /25 (255.255.255.128) to the Sales department. This IP subnet with Network number 192.168.1.0 has 126 valid Host IP addresses which satisfy the requirement of the Sales department. The subnet mask used for this subnet has 10000000 as the last octet.

Step 4: Allocate the next highest range, so let's assign 192.168.1.128 /26 (255.255.255.192) to the Purchase department. This IP subnet with Network number 192.168.1.128 has 62 valid Host IP Addresses which can be easily assigned to all the PCs of the Purchase department. The subnet mask used has 11000000 in the last octet.

Step 5: Allocate the next highest range, i.e., Accounts. The requirement of 25 IPs can be fulfilled with 192.168.1.192 /27 (255.255.255.224) IP subnet, which contains 30 valid host IPs. The network number of Accounts department will be 192.168.1.192. The last octet of subnet mask is 11100000.

Step 6: Allocate the next highest range to Management. The Management department contains only 5 computers. The subnet 192.168.1.224 /29 with the Mask 255.255.255.248 has exactly 6 valid host IP addresses. So, this can be assigned to Management. The last octet of the subnet mask will contain 11111000.

By using VLSM, the administrator can subnet the IP subnet in such a way that least number of IP addresses are wasted. Even after assigning IPs to every department, the administrator, in this example, is still left with plenty of IP addresses which was not possible if he has used FLSM.

Numerically, we can show the VLSM subnetting process as:

The given network address is: 192.168.1.0/24

Given requirement in descending order is:

Sales	100
Purchase	50
Accounts	25
Management	5

The complete range of the address in the above provided network is:

192.168.1.0 to 192.168.1.255

Divide the given network consisting 256 hosts into 2 networks with 128 hosts each:

192.168.1.0-192.168.1.127 (192.168.1.0/25)

192.168.1.128-192.168.1.255 (192.168.1.128/25)

The largest network requirement is of 100 hosts for Sales department. For this, we need to assign subnetwork with 128 hosts.

Let us assign the first divided subnetwork 192.168.1.0/25 to Sales Department.

We now have remaining subnetwork 192.168.1.128/25.

Dividing this subnetwork, two subnetworks with 64 hosts each are formed.

192.168.1.128 to 192.168.1.191 (192.168.1.128/26)

192.168.1.192 to 192.168.1.255 (192.168.1.192/26)

Our second network requirement is of 50 hosts for Purchase department. We need to assign subnetwork consisting of 64 hosts.

Assigning 192.168.1.128/26 to Purchase department.

The remaining subnetwork available is 192.168.192/26.

Dividing this subnetwork, two subnetworks with 32 hosts each are formed.

192.168.1.192 to 192.168.1.223 (192.168.1.192/27)

192.168.1.224 to 192.168.1.255 (192.168.1.224/27)

The third largest requirement is of 25 hosts for Account department.

Assigning 192.168.1.192/27 to Account Department.

Remaining subnetwork is 192.168.1.224/27

Dividing this subnetwork, two subnetworks with 16 hosts each are formed.

192.168.1.224 to 192.168.1.239 (192.168.1.224/28)

192.168.1.240 to 192.168.1.255 (192.168.1.240/28)

Our fourth network requirement is of 5 hosts for Management department. We need to assign subnetwork consisting of 8 hosts, which is sufficient.

So, again dividing the subnetwork 192.168.1.240/28, two subnetworks with 8 hosts each are formed.

192.168.1.240 to 192.168.1.247 (192.168.1.240/29)

192.168.1.248 to 192.168.1.255 (192.168.1.248/29)

Our fourth network requirement is of 5 hosts for Management department. We need to assign subnetwork consisting of 8 hosts.

We can Assign either of the subnetwork to Management department.

Summarizing the subnetting results,

Network Name	**Network ID**	**Subnet mask**	**No. of usable hosts**	**Usable Host ID Range**	**Broadcast address**
Sales	192.168.1.0	/25	126	192.168.1.1 to 192.168.1.126	192.168.1.127
Purchase	192.168.1.128	/26	62	192.168.129 to 192.168.1.190	192.168.1.191
Account	192.168.1.192	/27	30	192.168.1.193 to 192.168.1.222	192.168.1.223
Management	192.168.1.240	/29	6	192.168.1.241 to 192.168.1.246	192.168.1.247
Unused	192.168.1.224/28 (192.168.1.224 to 192.168.1.239)				
Unused	192.168.1.247/29 (192.168.1.247 to 192.168.1.255)				

FLSM Numerical Example:

Q1. If you are given a network 210.25.23.0 with the subnet mask 255.255.255.0, assign the networks to four different departments with 50 hosts each.

Ans: The complete range of the address in the above provided network is:

210.25.23.0 to 210.25.23.255

Total no of hosts available: 256 hosts

Each subnetwork requires 50 usable hosts. So, we need to assign n/w with 64 hosts each to the four departments.

Since we are using FLSM, the divided networks will be of same size. The given network consists of 256 hosts which needs to be divided into four subnetworks with 64 hosts each.

The process is as follows:

First of all, divide the given network range into four equal parts.

210.25.23.0 to 210.25.23.63	(210.25.23.0/26)
210.25.23.64 to 210.25.23.127	(210.25.23.64/26)
210.25.23.128 to 210.25.23.191	(210.25.23.128/26)
210.25.23.192 to 210.25.23.255	(210.25.23.192/26)

Now, as per the requirement, there are four networks required and we can assign the above networks to each of the four departments.

Network Name	**Network ID**	**Subnet mask**	**No. of usable hosts**	**Usable Host ID Range**	**Broadcast address**
Dept 1	210.25.23.0	/26	62	210.25.23.1 to 210.25.23.62	210.25.23.63
Dept 2	210.25.23.64	/26	62	210.25.23.65 to 210.25.23.126	210.25.23.127
Dept 3	210.25.23.128	/26	62	210.25.23.129 to 210.25.23.190	210.25.23.191
Dept 4	210.25.23.192	/26	62	210.25.23.193 to 210.25.23.254	210.25.23.255

Q2. Suppose you are network administrator with provided network 172.16.0.0/24. You need to manage the entire n/w by dividing into subnetworks so that each of the Development, Sales, Reception, HR and Production. How would you do so?

Ans: Provided network: 172.16.0.0/24. Here, /24 indicates 256 hosts are contained in the given network.

There are five departments to address the networks with. So, we divide the given network into 8 networks. 256/8 = 32

Each of the 8 subnetworks will contain 32 hosts each. The divided networks will be:

172.16.0.0 to 172.16.0.31	(172.16.0.0/27)
172.16.0.32 to 172.16.0.63	(172.16.0.32/27)
172.16.0.64 to 172.16.0.95	(172.16.0.64/27)
172.16.0.96 to 172.168.0.127	(172.16.0.96/27)
172.16.0.128 to 172.16.0.159	(172.16.0.128/27)
172.16.0.160 to 172.16.0.191	(172.16.0.160/27)
172.16.0.192 to 172.16.0.223	(172.16.0.192/27)
172.16.0.224 to 172.16.0.255)	(172.16.0.224/27)

Now, we can assign 5 of the above 8 subnetworks to the departments of our requirement.

The result will be as follows:

Network Name	**Network ID**	**Subnet mask**	**No. of usable hosts**	**Usable Host ID Range**	**Broadcast address**
Development	172.16.0.0	/27	30	172.16.0.1 to 172.16.0.30	172.16.0.31
Sales	172.16.0.32	/27	30	172.16.0.33 to 172.16.0.62	172.16.0.63
Reception	172.16.0.64	/27	30	172.16.0.65 to 172.16.0.94	172.16.0.95
HR	172.16.0.96	/27	30	172.16.0.97 to 172.168.0.126	172.168.0.127
Production	172.16.0.128	/27	30	172.16.0.129 to 172.16.0.158	172.16.0.159
Unused	172.16.0.160 to 172.16.0.191 (172.16.0.160/27)				
Unused	172.16.0.192 to 172.16.0.223 (172.16.0.192/27)				
Unused	172.16.0.224 to 172.16.0.255) (172.16.0.224/27)				

VLSM Numerical Example:

Q. If you are assigned an IP address 92.16.1.0/24 and plans to deploy CIDR. Here are some requirements which you have to fulfill for Subnet A= 120 hosts, Subnet B=60 hosts, Subnet C=30 hosts, Subnet D= 10 hosts, Subnet E= 5. You are also required to calculate subnet mask, range, netid, broadcast id for each subnet.

Ans: The given network address is: 92.16.1.0/24

Given requirement in descending order is:

Subnet A: 120

Subnet B: 60

Subnet C: 30

Subnet D: 10

Subnet E: 5

The complete range of the address in the above provided network is:

92.16.1.0 to 92.16.1.255

The largest network requirement is of 120 hosts for Subnet A. For this, we need to assign subnetwork with 128 hosts.

Divide the given network consisting 256 hosts into 2 networks with 128 hosts each:

92.16.1.0-92.16.1.127	(92.16.1.0/25)
92.16.1.128-92.16.1.255	(92.16.1.128/25)

Let us assign the first divided subnetwork 92.16.1.0/25 to Subnet A.

We now have remaining subnetwork 92.16.1.128/25.

Our second network requirement is of 60 hosts for Subnet B. We need to assign subnetwork consisting of 64 hosts.

Dividing this subnetwork, two subnetworks with 64 hosts each are formed.

92.16.1.128 to 92.16.1.191	(92.16.1.128/26)
92.16.1.192 to 92.16.1.255	(92.16.1.192/26)

Assigning 92.16.1.128/26 to Subnet B.

The remaining subnetwork available is 92.16.1.192/26.

The third largest requirement is of 30 hosts for Subnet C.

Dividing this subnetwork, two subnetworks with 32 hosts each are formed.

92.16.1.192 to 92.16.1.223 (92.16.1.192/27)

92.16.1.224 to 92.16.1.255 (92.16.1.224/27)

Assigning 92.16.1.192/27 to Subnet C.

Remaining subnetwork is 92.16.1.224/27

Our fourth network requirement is of 10 hosts for Subnet D. We need to assign subnetwork consisting of 16 hosts.

Dividing this subnetwork, two subnetworks with 16 hosts each are formed.

92.16.1.224 to 92.16.1.239 (92.16.1.224/28)

92.16.1.240 to 92.16.1.255 (92.16.1.240/28)

Assigning 92.16.1.224/28 to Subnet D.

Remaining subnetwork is 92.16.1.240/28

Our fifth network requirement is of 5 hosts for Subnet E. We need to assign subnetwork consisting of 8 hosts.

So, again dividing the subnetwork 92.16.1.240/28, two subnetworks with 8 hosts each are formed.

92.16.1.240 to 92.16.1.247 (92.16.1.240/29)

92.16.1.248 to 92.16.1.255 (92.16.1.248/29)

We can Assign either of the subnetwork to Subnet E. Let us assign 92.16.1.240/29 to Subnet E.

Summarizing the subnetting results,

Network Name	**Network ID**	**Subnet mask**	**No. of usable hosts**	**Usable Host ID Range**	**Broadcast address**
Subnet A	92.16.1.0	/25	126	92.16.1.1 to 92.16.1.126	92.16.1.127
Subnet B	92.16.1.128	/26	62	92.16.129 to 92.16.1.190	92.16.1.191
Subnet C	92.16.1.192	/27	30	92.16.1.193 to 92.16.1.222	92.16.1.223
Subnet D	92.16.1.224	/28	14	92.16.1.225 to 92.16.1.238	92.16.1.239
Subnet E	92.16.1.240	/29	6	92.16.1.241 to 92.16.1.246	92.16.1.247
Unused	92.16.1.248/29 (92.16.1.248 to 92.16.1.255)				

Note:

1. Network: 192.168.0.0/24, 2^8, 256 hosts

   Total Range: 192.168.0.0 to 192.168.0.255

2. Network: 192.168.1.0/25, 2^7, 128 hosts
   Total Range: 192.168.1.0 to 192.168.1.127

3. Network: 192.168.3.0/26, 2^6, 64 hosts
   Total Range: 192.168.3.0 to 192.168.3.63

4. Network: 192.168.0.0/23, 2^9, 512 hosts
   Total Range: 192.168.0.0 to 192.168.0.255, 192.168.1.0 to 192.168.1.255

5. Network: 192.168.1.0/23, 2^9, 512 hosts
   Total Range: 192.168.1.0 to 192.168.1.255, 192.168.2.0 to 192.168.2.255

6. Network: 172.16.10.0/23, 2^9, 512 hosts
   Total Range: 172.16.10.0 to 172.16.10.255, 172.16.11.0 to 172.168.11.255

7. Network: 172.16.10.0/22, 2^10, 1024 hosts
   Total Range: 172.16.10.0 to 172.16.10.255
   172.16.11.0 to 172.16.11.255
   172.16.12.0 to 172.16.12.255
   172.16.13.0 to 172.16.13.255

8. Network: 172.16.10.0/21, 2^11, 2048 hosts
   Total Range: 172.16.10.0 to 172.16.10.255
   172.16.11.0 to 172.16.11.255
   172.16.12.0 to 172.16.12.255
   172.16.13.0 to 172.16.13.255
   172.16.14.0 to 172.16.14.255
   172.16.15.0 to 172.16.15.255
   172.16.16.0 to 172.16.16.255
   172.16.17.0 to 172.16.17.255

Q2. Given Network: 192.168.0.0/23

Requirement:

A: 128 hosts, B: 64 hosts, C: 31 hosts, D: 15 hosts

Solution: Total Range= 192.168.0.0 to 192.168.0.255 (192.168.0.0/24)

192.168.1.0 to 192.168.1.255 (192.168.1.0/24)

A-> 128 hosts, need to assign n/w of 256 hosts

Let us assign: 192.168.0.0/24

B-> 64 hosts, need to assign n/w of 128 hosts

Divide 192.168.1.0/24,

192.168.1.0 to 192.168.1.127 (192.168.1.0/25)

192.168.1.128 to 192.168.1.255 (192.168.1.128/25)

Assign 192.168.1.0/25 to B.

Remaining: 192.168.1.128/25

C->31 hosts, need to assign n/w of 64 hosts

Divide 192.168.1.128/25,

192.168.1.128 to 192.168.1.191 (192.168.1.128/26)

192.168.1.192 to 192.168.1.255 (192.168.1.192/26)

Assign 192.168.1.128/26 to C.

Remaining: 192.168.1.192/26

D-> 15 hosts, need to assign n/w of 32 hosts

Divide 192.168.1.192/26,

192.168.1.192 to 192.168.1.223 (192.168.1.192/27)

192.168.1.224 to 192.168.1.255 (192.168.1.224/27)

Assign 192.168.1.192/27 to D

Unused: 192.168.1.224/27

Network Address Translation (NAT):

The number of home users and small businesses that want to use the Internet is ever increasing. An ISP with a block of addresses could dynamically assign an address to this user. Earlier, an address was given to a user when needed. But the situation is different now. Home users and small businesses need more than one IP addresses since they have several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem. A quick solution to this problem is called Network Address Translation. NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set, the traffic outside can use the small set.

To separate the addresses used inside the home or business and the ones used for the internet, the internet authorities have reserved three sets of addresses as private addresses.

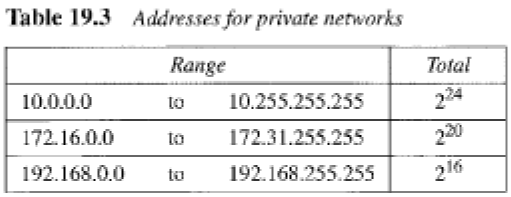
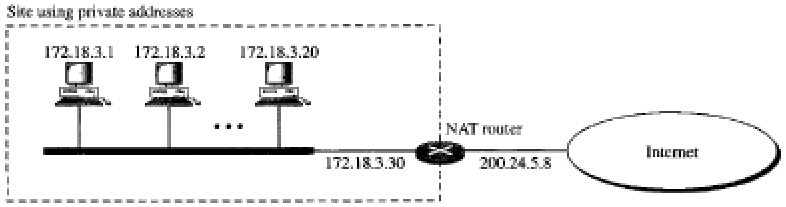


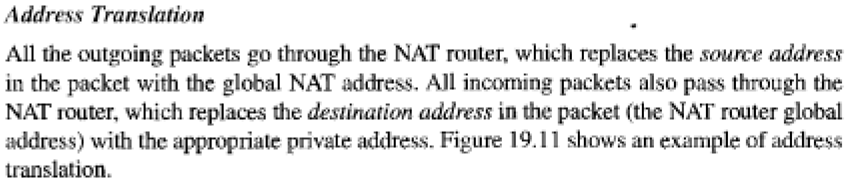
Table 19.3 *Addresses for private networks*

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

Any organization can use an address out of this set without permission from the internet authorities. Everyone knows that these reserved addresses are for private networks. They are unique inside the organization, but they are not unique globally.

Figure 19.10 *A NAT implementation*

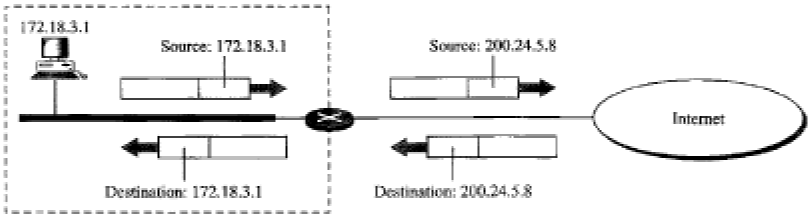




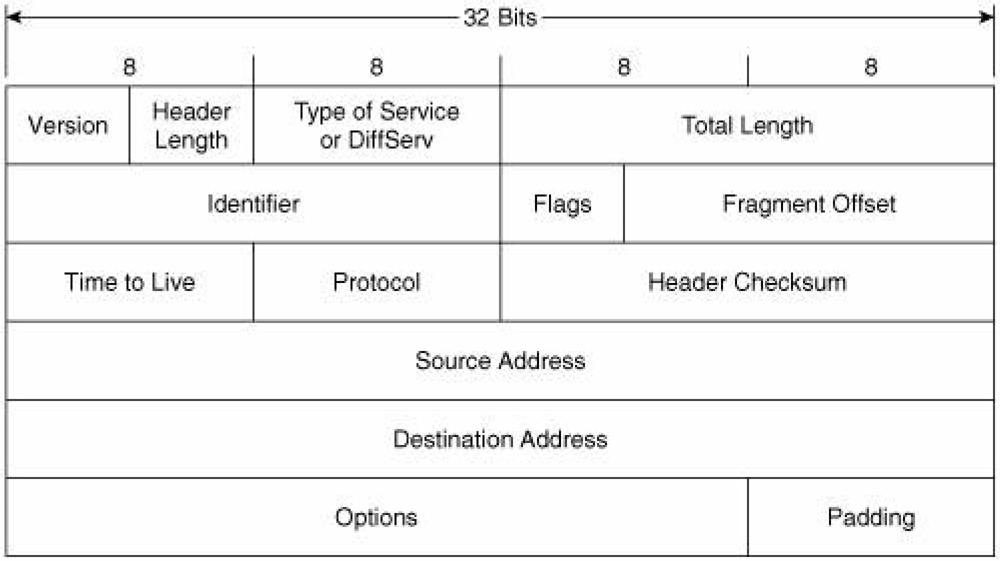
Address Translation

All the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address. All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet (the NAT router global address) with the appropriate private address. Figure 19.11 shows an example of address translation.

Figure 19.11 *Addresses in a NAT*



IPv4 Packet Structure:



- Version: Version no. of Internet Protocol used (e.g. IPv4).
- IHL: Internet Header Length; Length of entire IP header.
- Type of service: This provides network service parameters.
- Total Length: Length of entire IP Packet (including IP header and IP Payload).
- Identification: If IP packet is fragmented during the transmission, all the fragments contain same identification number to identify original IP packet they belong to.
- Flags: As required by the network resources, if IP Packet is too large to handle, these 'flags' tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.
- Fragment Offset: This offset tells the exact position of the fragment in the original IP Packet.
- Time to Live: To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- Protocol: Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example, protocol number of ICMP is 1, TCP is 6 and UDP is 17.
- Header Checksum: This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.
- Source Address: 32-bit address of the Sender (or source) of the packet.
- Destination Address: 32-bit address of the Receiver (or destination) of the packet.
- Options: This is optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc.

Packet Fragmentation:

IP fragmentation is an Internet Protocol (IP) process that breaks packets into smaller pieces (fragments), so that the resulting pieces can pass through a link with a smaller maximum transmission unit (MTU) than the original packet size. The fragments are reassembled by the receiving host.

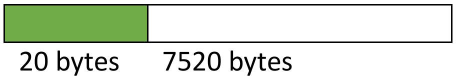
Fragmentation is necessary for data transmission, as every network has a unique limit for the size of datagrams that it can process. If a datagram is being sent that is larger than the receiving server's MTU, it has to be fragmented in order to be transmitted completely.

a) **With IP Options**

Assume a link layer protocol, called Newnet with MTU 1600 bytes. A TCP segment with 7540 bytes of user data is to be sent over Newnet link. There are 20 bytes IP options field involved. How many IP fragments are transmitted and what is the offset and IP payload length of each fragment?

Given,

Packet size: 7540 bytes



MTU: 1600 bytes

(20 bytes header + 1580 bytes data)

Data bytes should be in multiple of 8. So, consider 1576 bytes data in each packet.

No. of fragments = 7520/1576 = 4.775 => 5

Fragments	Payback length	Flag	Offset
1	1576	1	0
2	1576	1	197
3	1576	1	394
4	1576	1	591
5	1216	0	788

b) **Without IP Options**

Assume a link layer protocol, called Newnet with MTU 1400 bytes. A TCP segment with 6000 bytes of user data is to be sent over Newnet link. There are no IP options field involved. How many IP fragments are transmitted and what is the offset and IP payload length of each fragment?

Given,

Packet size: 6000 bytes

MTU: 1400 bytes

6000 bytes

Data bytes should be in multiple of 8. 1400 is divisible by 8. So, we can transmit 1400 bytes in each packet

No. of fragments = 6000/1400 = 4.28 => 5

Fragments	Payback length	Flag	Offset
1	1400	1	0
2	1400	1	175
3	1400	1	350
4	1400	1	525
5	400	0	700

Issues with IPv4:

Changes since IPv4 was developed (mid 70's)

- Provider market has changed dramatically
- Immense increase in user and traffic on the Internet
- Rapid technology advancement
- Bandwidth increase from kb/s to Tb/s

IPv4 issues: The major issues in IPv4 are

- Deficiency of address space - The devices connected to the Internet grows exponentially. The size of address space 2^32 is quickly exhausted;
- Too large routing tables

Some more issues are:

- Weak expansibility of the protocol - the insufficient size of heading IPv4 doesn't allow to place demanded quantity of additional parameters in it;
- Problem of safety of communications - it is not stipulated any means for differentiation of access to the information placed in a network;
- Absence of support of quality of service (QoS) - accommodation of the information about throughput, the delays and demanded for normal work of some network appendices is not supported;
- The problems connected with the mechanism of a fragmentation - the size of the maximal block of data transmission on each concrete way is not defined;
- Absence of the auto-configuration IP addresses mechanism.

Overview of IPv6:

To respond to the need for a large IP address space, a new IP protocol, IPv6, was developed. Also, major issues of IPv4 are addressed in this version.

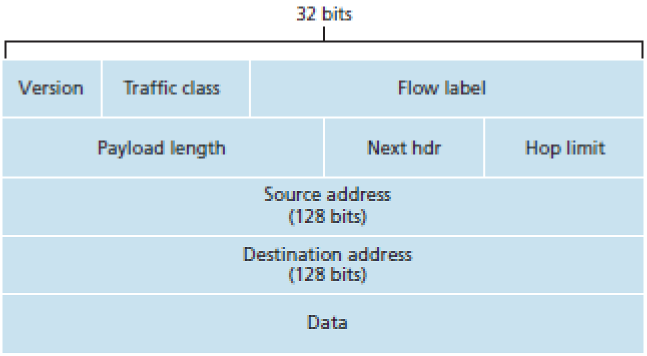
The most important changes introduced in IPv6 are evident in the datagram format:

- *Expanded addressing capabilities.* IPv6 increases the size of the IP address from 32 to 128 bits. This ensures that the world won't run out of IP addresses. Now, every grain of sand on the planet can be IP-addressable. In addition to unicast and multicast addresses, IPv6 has introduced a new type of address, called an **any-cast address**, which allows a datagram to be delivered to any one of a group of hosts.
- *A streamlined 40-byte header.* As discussed below, a number of IPv4 fields have been dropped or made optional. The resulting 40-byte fixed-length header allows for faster processing of the IP datagram. A new encoding of options allows for more flexible options processing.
- *Flow labeling and priority.* IPv6 has an elusive definition of a flow. This allows "labeling of packets belonging to particular flows for which the sender requests special handling, such as a non-default quality of service or real-time service." For example, audio and video transmission might likely be treated as a flow.

IPv6 Simplifications:

- Remove header checksum: Because the transport-layer (for example, TCP and UDP) and link-layer (for example, Ethernet) protocols in the Internet layers perform check summing, the designers of IP probably felt that this functionality was sufficiently redundant in the network layer that it could be removed.
- Remove hop-by-hop segmentation: IPv6 does not allow for fragmentation and reassembly at intermediate routers; these operations can be performed only by the source and destination. If an IPv6 datagram received by a router is too large to be forwarded over the outgoing link, the router simply drops the datagram and sends a "Packet Too Big" ICMP error message back to the sender.
- Options. An options field is no longer a part of the standard IP header. However, it has not gone away. Instead, the options field is one of the possible next headers pointed to from within the IPv6 header. The removal of the options field results in a fixed-length, 40-byte IP header.

IPv6 Header:



S.N.	Field & Description
1	Version (4-bits): It represents the version of Internet Protocol, i.e. 0110
2	Traffic Class (8-bits): These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router know what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN).
3	Flow Label (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information.
4	Payload Length (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data.
5	Next Header (8-bits): This field is used to indicate either the type of Extension Header, or if the Extension Header is not present, then it indicates the Upper Layer PDU.
6	Hop Limit (8-bits): This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0, the packet is discarded.
7	Source Address (128-bits): This field indicates the address of originator of the packet.
8	Destination Address (128-bits): This field provides the address of intended recipient of the packet.

IPv6 Addresses :(IPv6 Format)

IPv6 address is 128 bits long and is arranged in eight groups, each of which is 16 bits. Each group is expressed as four hexadecimal digits and the groups are separated by colons.

An example of a full IPv6 address: FE80:CD00: 0000:0CDE: 1257:0000:211E:729C

IPv6 has three address categories:

- Unicast - identifies exactly one interface
- Multicast - identifies a group; packets get delivered to all members of the group
- Anycast - identifies a group; packets normally get delivered to nearest member of the group

IPv6 Address Abbreviations and CIDR:

Even after converting into Hexadecimal format, IPv6 address remains long. An IPv6 address may be abbreviated to shorter notations by application of the following rules:

Rule 1: Discard leading zero (es)

That address can be shortened because the addressing scheme allows the omission of any leading zero, as well as any sequences consisting only of zeroes.

E.g.: FE80:CD00:0000:0CDE:1257:0000:211E:729C

Here's the short version:

FE80:CD00:0:CDE:1257:0:211E:729C

Rule 2: If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign **::**

2001:0000:3238:DFE1:63:0000:0000:FEFB

can be written as

2001:0000:3238:DFE1:63::FEFB

The IPv6 addressing architecture allows you use the two-colon (::) notation to represent contiguous 16-bit fields of zeros.

CIDR Notation is similar to IPv4 addresses, IPv6 addresses consist of NetworkID + HostID, and use classless notation to identify (distinguish between) the two. Network ID is also referred to as prefix, and the number of bits allocated to Network ID as prefix length. Information on the prefix is provided together with each IPv6 address as a slash (/) at the end of the address followed by the prefix length.

For example, the site prefix of the IPv6 address 2001:db8:3c4d:0015:0000:0000:1a2f:1a2b/48 is contained in the leftmost 48 bits, 2001:db8:3c4d. You use the following representation, with zeros compressed, to represent this prefix: 2001:db8:3c4d::/48

IPv6 vs IPv4:

IPv4	IPv6

IPv4 addresses are 32 bit length.	IPv6 addresses are 128 bit length.
IPv4 addresses are binary numbers represented in decimals.	IPv6 addresses are binary numbers represented in hexadecimals.
IPSec support is only optional.	Inbuilt IPSec support.
Fragmentation is done by sender and forwarding routers.	Fragmentation is done only by sender.
No packet flow identification	Packet flow identification is available within the IPv6 header using the Flow Label field.
Checksum field is available in IPv4 header	No checksum field in IPv6 header
Options fields are available in IPv4 header	No option fields, but IPv6 Extension headers are available.

Routing:

Routing is a process which is performed by layer 3 (or network layer) devices in order to deliver the packet by choosing an optimal path from one network to another.

Routing Algorithm:

A routing algorithm is a procedure that lays down the route or path to transfer data packets from source to the destination**.**

Routing Protocol:

A routing protocol specifies how routers communicate with each other to distribute information that enables them to select routes between nodes on a computer network.

Types of Routing:

There are 3 major types of routing:

Static routing:

Static routing is a process in which we have to manually add routes in routing table. No routing overhead for router CPU which means a cheaper router can be used to do routing. It adds security because only network administrator can allow routing to particular networks only. It does not require any bandwidth usage between routers. But, for large networks, adding each route manually in the routing table on each router is a hectic task.

Default Routing

This is the method where the router is configured to send all packets towards a single router (next hop). It doesn't matter to which network the packet belongs, it is forwarded out to router which is configured for default routing.

Dynamic Routing

Dynamic routing makes automatic adjustment of the routes according to the current state of the route in the routing table. Dynamic routing uses protocols to discover network destinations and the routes to reach it. RIP and OSPF are the best examples of dynamic routing protocol. Automatic adjustment will be made to reach the network destination if one route goes down.

A dynamic protocol has following features:

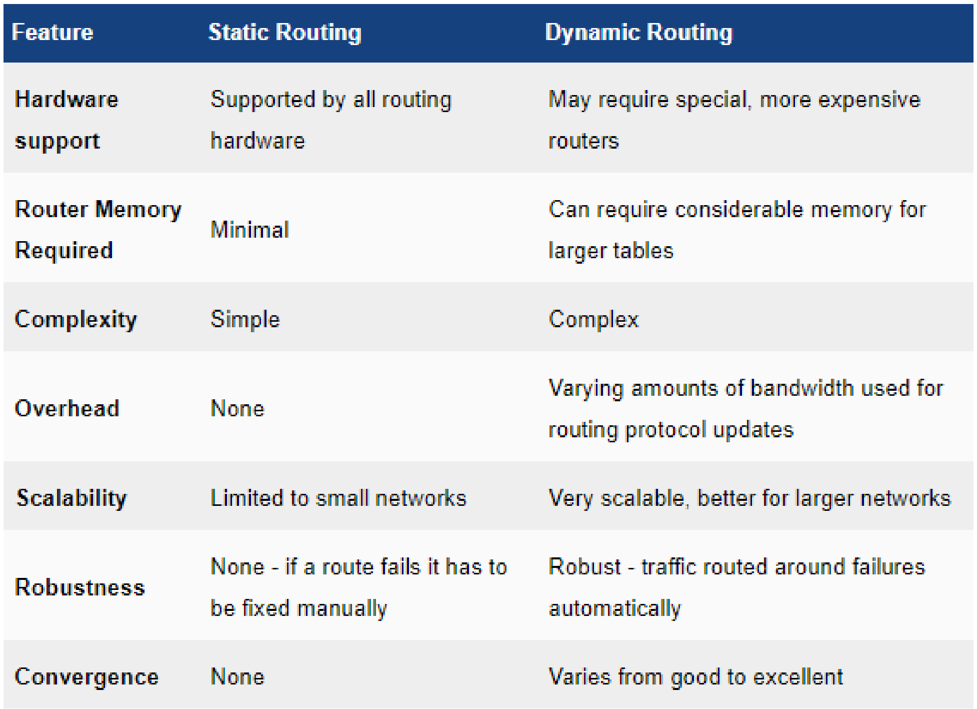
- The routers should have the same dynamic protocol running in order to exchange routes.
- When a router finds a change in the topology then router advertises it to all other routers.

Advantages

- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.

Disadvantage

- Consumes more bandwidth for communicating with other neighbors.
- Less secure than static routing.



Feature	Static Routing	Dynamic Routing
Hardware support	Supported by all routing hardware	May require special, more expensive routers
Router Memory Required	Minimal	Can require considerable memory for larger tables
Complexity	Simple	Complex
Overhead	None	Varying amounts of bandwidth used for routing protocol updates
Scalability	Limited to small networks	Very scalable, better for larger networks
Robustness	None - if a route fails it has to be fixed manually	Robust - traffic routed around failures automatically
Convergence	None	Varies from good to excellent

Fixed Path Routing:

A route is selected for each source and destination pair of nodes in the network. The routes are fixed and changes only if topology of the network changes. It is sometimes also referred to as static routing since the routes are fixed as in static routing.

Flooding:

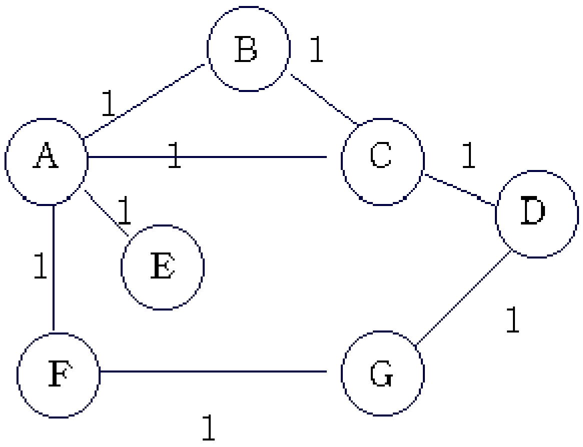
Flooding adapts the technique in which every incoming packet is sent on every outgoing line expect from which it arrived. One problem with this is that packets may go in a loop and as a result of which a node may receive duplicate packets. These problems can overcome with the help of sequence numbers and hop count. No routing table is required for flooding and no network information like topology, load condition, cost of different paths is required. All possible routes between source and destination is tried, and there will be at least one route which is the shortest.

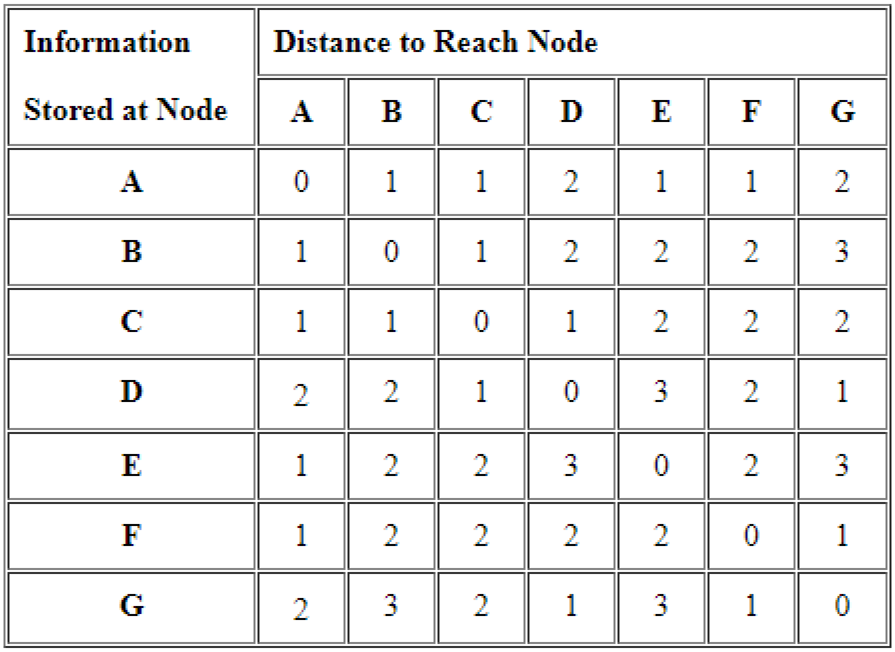
Shortest Path Routing:

Shortest path routing refers to the process of finding paths through a network that have a minimum of distance or other cost metric. Shortest-path routing algorithms have existed since two independent research works by Bellman and Ford, and Dijkstra in 1950's. The difference between these two algorithms is the way information needed for computing the shortest-path is used. In the context of packet-switched networks and Internet routing, in particular, Bellman-Ford's algorithm has enabled the development of distance-vector routing protocols while Dijkstra's algorithm has paved the way to the introduction of link-state routing protocols.

Distance Vector Routing Algorithm:

A distance-vector routing protocol in data networks determines the best route for data packets based on distance. Distance-vector routing protocols measure the distance by the number of routers a packet has to pass, one router counts as one hop. The vector describes the route of the message over a given set of network nodes. To determine the best route across a network router on which a distance-vector protocol is implemented exchange information with one another, usually routing tables plus hop counts for destination networks and possibly other traffic information. The basic idea here is that each node receives some information from one or more of its directly attached neighbors, performs a calculation, and then distributes the results of its calculation back to its neighbors.

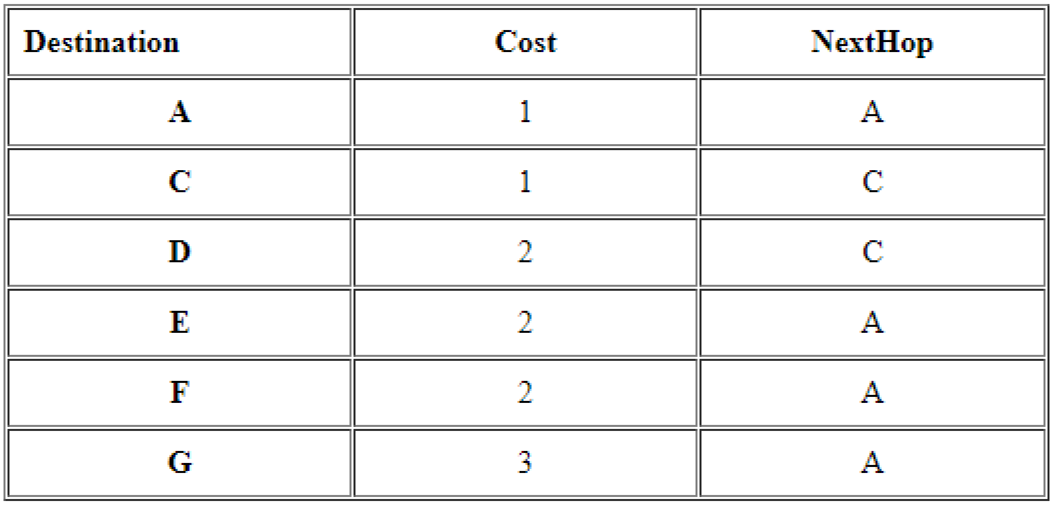




Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Table 2. final distances stored at each node (global view).

For example, Table below shows the complete routing table maintained at node B for the network in figure above.



Destination	Cost	NextHop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

Table 3. Routing table maintained at node B.

Link State Routing:

The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. Each collection of best paths will then form each node's routing table. Link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology. Based on this learned topology, each router is then able to compute its routing table by using a shortest path computation.

Calculation of shortest path:

To find shortest path, each node needs to run the famous **Dijkstra algorithm**. Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph. This famous algorithm uses the following steps:

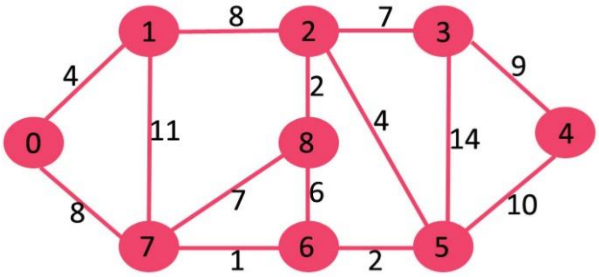
Step-1: The node is taken and chosen as a root node of the tree, this creates the tree with a single node, and now set the total cost of each node to some value based on the information in Link State Database

Step-2: Now the node selects one node, among all the nodes not in the tree like structure, which is nearest to the root, and adds this to the tree. The shape of the tree gets changed.

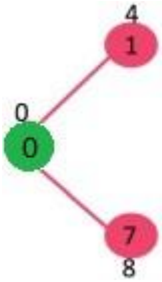
Step-3: After this node is added to the tree, the cost of all the nodes not in the tree needs to be updated because the paths may have been changed.

Step-4: The node repeats the Step 2 and Step 3 until all the nodes are added in the tree.

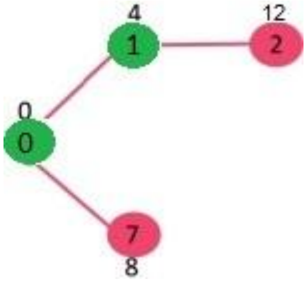
Let us understand with the following example:



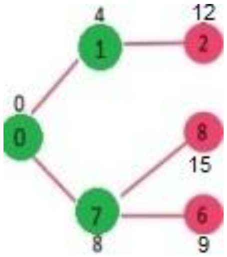
Adjacent vertices of 0 are 1 and 7. The distance values of 1 and 7 are updated as 4 and 8. Following subgraph shows vertices and their distance values, only the vertices with finite distance values are shown.



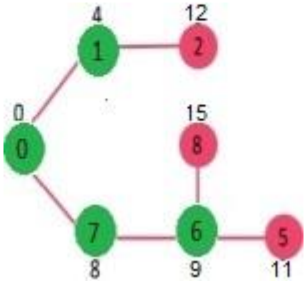
The vertex 1 is picked and added. So the set now becomes {0, 1}. Update the distance values of adjacent vertices of 1. The distance value of vertex 2 becomes 12.



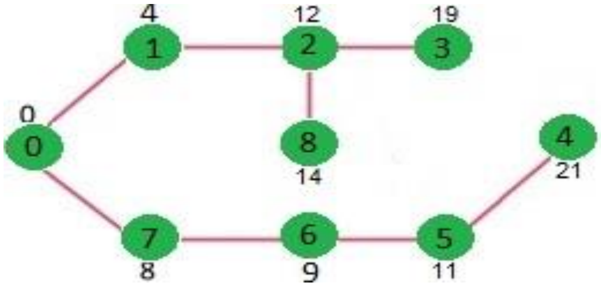
Vertex 7 is picked. So set now becomes {0, 1, 7}. Update the distance values of adjacent vertices of 7. The distance value of vertex 6 and 8 becomes finite (15 and 9 respectively).



Vertex 6 is picked. So set now becomes {0, 1, 7, 6}. Update the distance values of adjacent vertices of 6. The distance value of vertex 5 and 8 are updated.



We repeat the above steps until set doesn't include all vertices of given graph. Finally, we get the following Shortest Path Tree (SPT).



OSPF (Open Path Shortest First):

Open Shortest Path First (OSPF) is a link state routing protocol (LSRP) that uses the Shortest Path First (SPF) network communication algorithm (Dijkstra's algorithm) to calculate the shortest connection path between known devices.

Using OSPF, a router that learns of a change to a routing table (when it is reconfigured by network staff, for example) or detects a change in the network immediately multicasts the information to all other OSPF hosts in the network so they will all have the same routing table information. OSPF sends only the part that has changed and only when a change has taken place. When routes change -- sometimes due to equipment failure -- the time it takes OSPF routers to find a new path between endpoints with no loops (which is called "open") and that minimizes the length of the path is called the convergence time. OSPF bases its path choices on "link states" that take into account additional network information, including assigned cost metrics that give some paths higher assigned costs.

For example, a person in city A wants to travel to city M and is given two options:

Travel via cities B and C. The route would be ABCM. And the distance (or bandwidth cost in the networking case) for A-B is 10 miles, B-C is 5 miles and C-M is 10 miles.

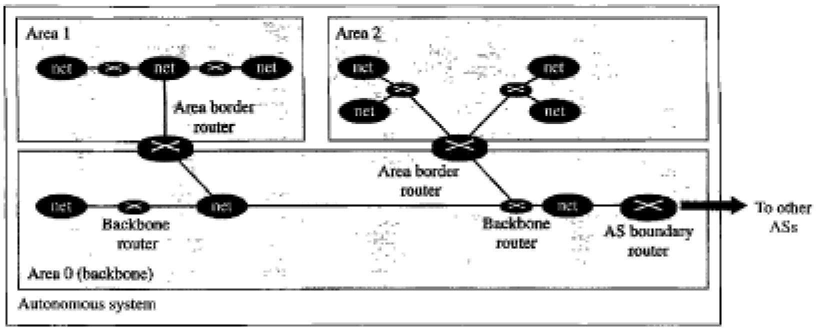
Travel via city F. The route would be AFM. And the distance for A-F is 20 miles and F-M is 10 miles.

The shortest route is always the one with least amount of distance covered in total. Thus, the ABCM route is the better option (10+5+10=25), even though the person has to travel to two cities as the associated total cost to travel to the destination is less than the second option with a single city (20+10=30). OSPF performs a similar algorithm by first calculating the shortest path between the source and destination based on link bandwidth cost and then allows the network to send and receive IP packets via the shortest route.

OSPF Network Topology:

Two routers communicating OSPF to each other exchange information about the routes they know about and the cost for them to get there. When many OSPF routers are part of the same network, information about all of the routes in a network are learned by all of the OSPF routers within that network—technically called an area. Each OSPF router passes along information about the routes and costs they’ve heard about to all of their adjacent OSPF routers, called neighbors.

Figure 22.24 *Areas in an autonomous system*

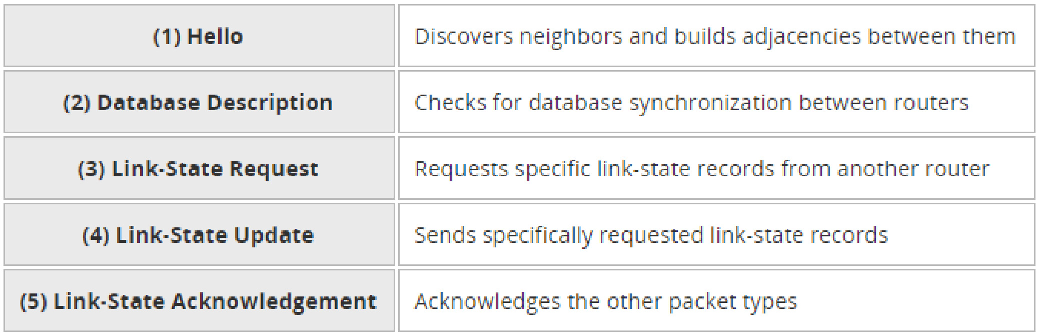


OSPF Protocols (hello, exchange, flooding):

Routers periodically send hello packets on all interfaces to establish and maintain neighbor relationships. Hello packets are multicast on physical networks that have a multicast or broadcast capability, which enables dynamic discovery of neighboring routers. Hello packets are sent out every 10 seconds which helps to detect failed neighbors. RouterDeadInterval (default 40 seconds) is specified for detecting such neighbors. Also, hello message ensures that link between neighbors is bidirectional. Neighboring routers agree on intervals where hello interval is set so that a link is not accidentally brought down.

OSPF uses hello packets and two timers to check if a neighbor is still alive or not:

- Hello interval: this defines how often we send the hello packet.
- Dead interval: this defines how long we should wait for hello packets before we declare the neighbor dead.



(1) Hello	Discovers neighbors and builds adjacencies between them
(2) Database Description	Checks for database synchronization between routers
(3) Link-State Request	Requests specific link-state records from another router
(4) Link-State Update	Sends specifically requested link-state records
(5) Link-State Acknowledgement	Acknowledges the other packet types

Path Vector Routing:

Distance vector and Link State routing are both intradomain routing protocols. They can be used inside an autonomous system, but not between the autonomous systems. These two protocols are not suitable for interdomain routing mostly because of scalability. There is a need for a third routing protocol which we call path vector routing.

Path Vector Routing is a routing algorithm of network layer, and it is useful for interdomain routing. The principle of path vector routing is similar to that of distance vector routing. In path vector routing, we assume that there is one node (there can be more, but one is enough) in each autonomous system that acts on behalf of the entire autonomous system, referred to as speaker node. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring autonomous systems. A speaker node advertises the path, not the metrics of the nodes, in its autonomous system or other autonomous systems.

Border Gateway Protocol (BGP):

Border gateway protocol (BGP) is an interdomain routing protocol using path vector routing. BGP is protocol that manages how packets are routed across the internet through the exchange of routing and reachability information between edge routers. BGP directs packets between autonomous systems (AS) -- networks managed by a single enterprise or service provider. Traffic that is routed within a single network AS is referred to as internal BGP, or iBGP. More often, BGP is used to connect one AS to other autonomous systems, and it is then referred to as an external BGP, or eBGP.

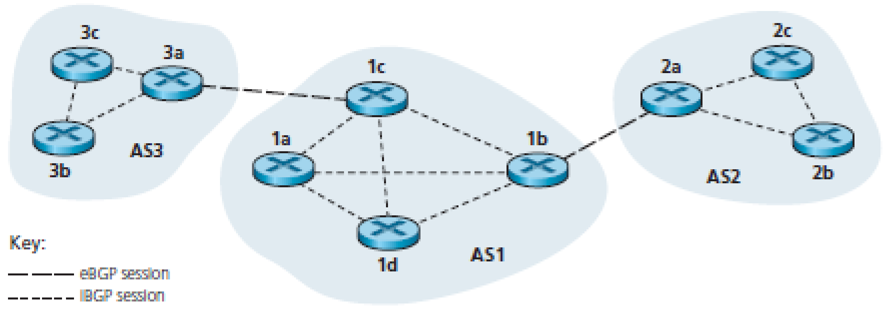


Figure 4.40 ♦ eBGP and iBGP sessions

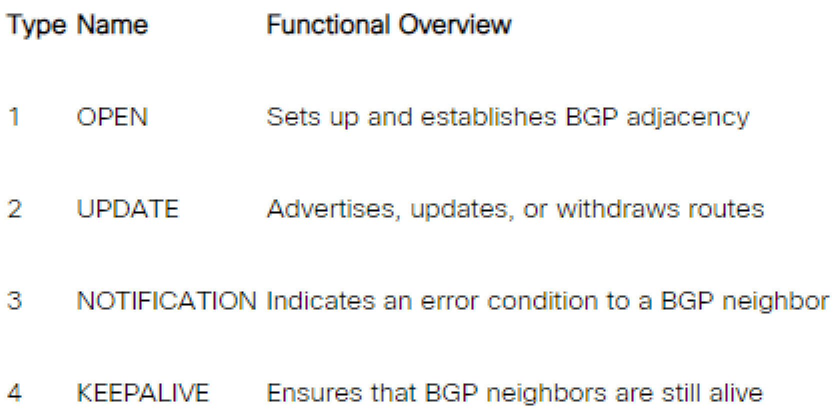
All other routing protocols are concerned solely with finding the optimal path towards all known destinations. BGP cannot take this simplistic approach because the peering agreements between ISPs almost always result in complex routing policies. To help network operators implement these policies, BGP carries a large number of attributes with each IP prefix:

- Local Preference – The local preference attribute is used to dictate how traffic prefers to leave a specific BGP ASN. This attribute is passed between neighbors within the same ASN. The highest local preference gets priority.
- Local Routes – Routes which have been sourced from the local router will be preferred over those sourced from other routers.
- Shortest AS_PATH – With BGP, the path is notated by the ASN of the external BGP networks that must be traversed to reach the destination network; e.g. 10 20 30 means that the traffic must pass through ASNs 10, 20, and 30 to reach the destination. If multiple options exist to a specific network, the one with the shortest AS path will be preferred.
- Origin – With origin, BGP is looking for the source of the initial network advertisement, for example if it was redistributed from an IGP, an EGP or through an unknown source. When analyzing this attribute, routes that have originated from an IGP are preferred to those from an EGP.
- BGP Neighbor Type – There are two different types of BGP neighborship: internal and external. A BGP neighborship that exists within the same ASN between two devices is considered internal, and a BGP neighborship that exists between devices from different ASNs is considered external. External (or eBGP) routes are preferred to Internal (iBGP) routes.
- Lowest Router-ID – The route with the lowest BGP router ID will be preferred
- Lowest Neighbor Address – The route coming through a neighbor with the lowest address will be preferred.

BGP Message (Packet) Types:

BGP communication uses four message types: Open, Update, Keep Alive, Notification.

There is a 5th message type defined in BGP called Route-Refresh to support the route refresh capability. 'Route Refresh Capability', which would allow the dynamic exchange of route refresh request between BGP speakers and subsequent re-advertisement. One possible application of this capability is to facilitate non-disruptive routing policy changes.



Type	Name	Functional Overview
1	OPEN	Sets up and establishes BGP adjacency
2	UPDATE	Advertises, updates, or withdraws routes
3	NOTIFICATION	Indicates an error condition to a BGP neighbor
4	KEEPALIVE	Ensures that BGP neighbors are still alive

Open Message:

Once two BGP routers have completed a TCP 3-way handshake they will attempt to establish a BGP session, this is done using open messages. In the open message we will find some information about the BGP router, these have to be negotiated and accepted by both routers before we can exchange any routing information.

Update Message:

Once two routers have become BGP neighbors, they can start exchanging routing information. This is done with the update message. In the update message you will find information about the prefixes that are advertised.

Notification Message:

A Notification message is sent when an error is detected with the BGP session, such as a hold timer expiring, neighbor capabilities change, or a BGP session reset is requested. This causes the BGP connection to close.

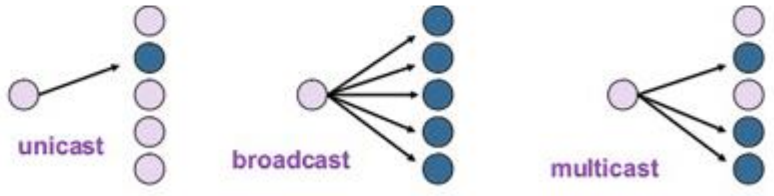
Keep Alive Message:

BGP does not rely on the TCP connection state to ensure that the neighbors are still alive. Keepalive messages are exchanged every one-third of the Hold Timer agreed upon between the two BGP routers. Cisco devices have a default Hold Time of 180 seconds, so the default Keepalive interval is 60 seconds. If the Hold Time is set for zero, no Keepalive messages are sent between the BGP neighbors.

Unicast, Multicast and Broadcast Routing:

Packets are routed across a network by three simple methods i.e., Unicast, Broadcast, and Multicast.

- Unicast: from one source to one destination i.e. One-to-One
- Broadcast: from one source to all possible destinations i.e. One-to-All
- Multicast: from one source to multiple destinations stating an interest in receiving the traffic i.e. One-to-Many



Unicast routing is a type of routing where data is forwarded from one single computer to another single computer over the network. In Unicast type of communication, there is only one sender, and one receiver. Distance vector, Link State and Path Vector Routing are unicast routing algorithms. One of the simplest everyday examples of unicast transmission would be a phone call between two people.

Multicast routing is a type of routing where multicast traffic is addressed for a group of devices on the network. IP multicast traffic are sent to a group and only members of that group receive and/or process the Multicast traffic. Devices which are interested in a particular Multicast traffic must join to that Multicast group to receive the traffic. IP Multicast Groups are identified by Multicast IP Addresses (IPv4 Class D Addresses). In Multicast, the sender transmit only one copy of data and it is delivered and/or processed to many devices (Not as delivered and processed by all devices as in Broadcast) who are interested in that traffic. Multicast IP Routing protocols are used to distribute data (for example, audio/video streaming broadcasts) to multiple recipients.

Broadcast routing is a type of routing where data is sent from one computer once and a copy of that data will be forwarded to all the devices. In Broadcast, there is only one sender and the data is sent only once. But the Broadcast data is delivered to all connected devices. Television signals sent from a public network to viewers across the country or globe are a simple example of broadcast transmission.

Internet Control Protocols:

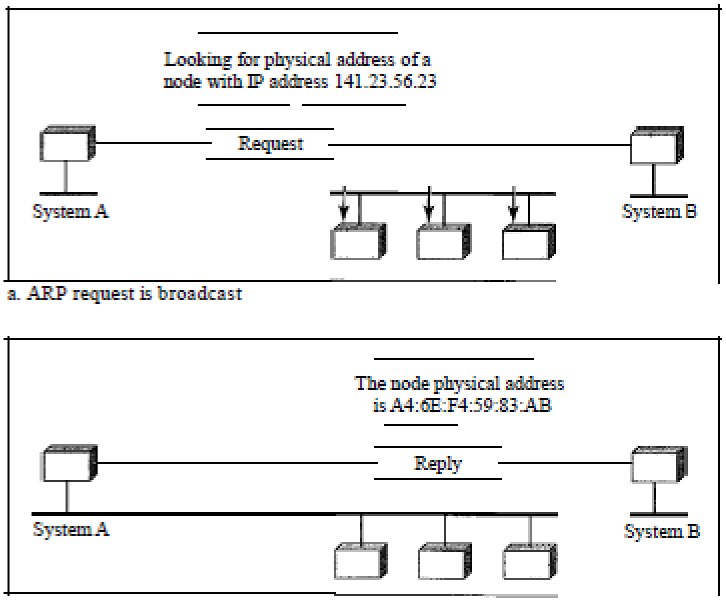
Address Resolution Protocol (ARP):

If a machine talks to another machine in the same network, it requires its physical or MAC address. But since the application has given the destination's IP address it requires some mechanism to bind the IP address with its MAC address. This is done through Address Resolution protocol (ARP).

Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network.

Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and physical addresses. The packet is unicast directly to the inquirer by using the physical address received in the query packet.

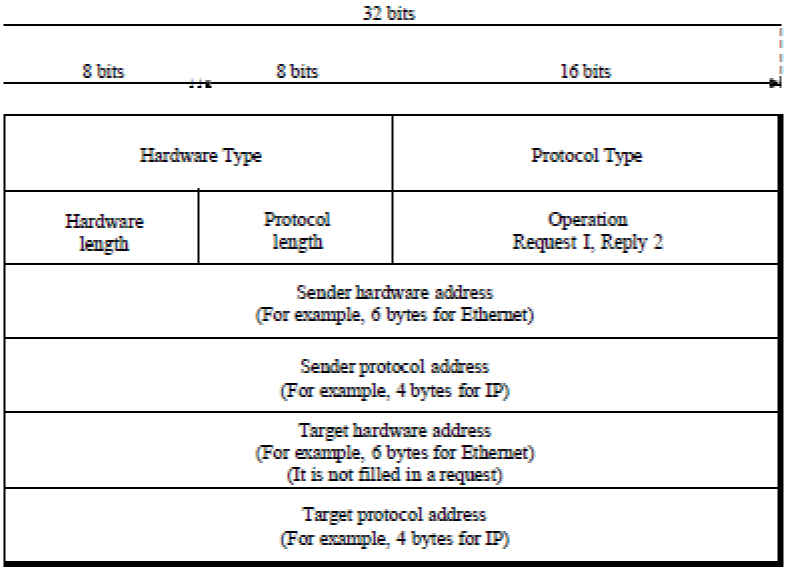
ARP operation


Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B
The node physical address is A4:6E:F4:59:83:AB
Reply
System A
System B

a. ARP request is broadcast

b. ARP reply is unicast

Figure 21.2 *ARP packet*



8 bits	8 bits	16 bits
Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request I, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

Reverse Address Resolution Protocol (RARP):

Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address. Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine.

The machine can get its physical address (by reading its NIC, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol. A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply. The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program.

There is a serious problem with RARP: Broadcasting is done at the data link layer. The physical broadcast address, allis in the case of Ethernet, does not pass the boundaries of a network. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet. This is the reason that RARP is almost obsolete.

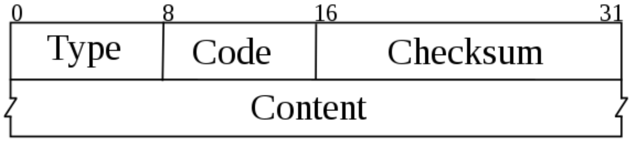
Internet Control Message Protocol (ICMP):

- ICMP is a TCP/IP network layer protocol that provides troubleshooting, control and error message services.
- Internet Control Message Protocol is also known as RFC 792.
- While ICMP is not used regularly in end-user applications, it is used by network administrators to troubleshoot Internet connections in diagnostic utilities.
- An ICMP message is created as a result of errors in an IP datagram. These errors are reported to the originating datagram's source IP address.
- An ICMP message is encapsulated directly within a single IP datagram and reports errors in the processing of datagrams.
- ICMP messages are transmitted as datagrams and consist of an IP header that encapsulates the ICMP data.
- ICMP packets are IP packets with ICMP in the IP data portion. ICMP messages also contain the entire IP header from the original message, so the end system knows which packet failed.

There can be several reasons behind reporting the error like:

- A router with a datagram for a host in another network, may not find the next hop (router) to the final destination host.
- Datagram’s time-to-live field has become zero.
- There may be ambiguity in the header of IP datagram.
- It may happen that all the fragments of datagram if do not arrive within a time limit to the destination host.

ICMP Header Format:



ICMP is available for both IPv4 and IPv6. The header format is similar for both versions of ICMP. ICMPv4 is the messaging protocol for IPv4. ICMPv6 provides these same services for IPv6 but includes additional functionality.

ICMPv4: ICMP for IPv4

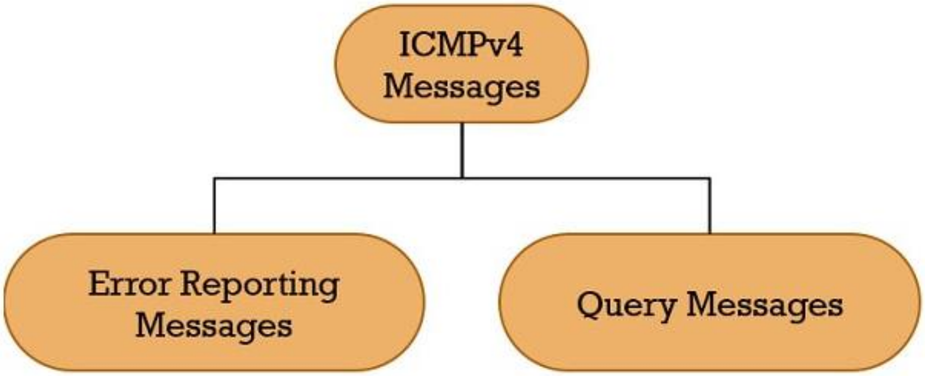


Figure 21.9 *Error-reporting messages*

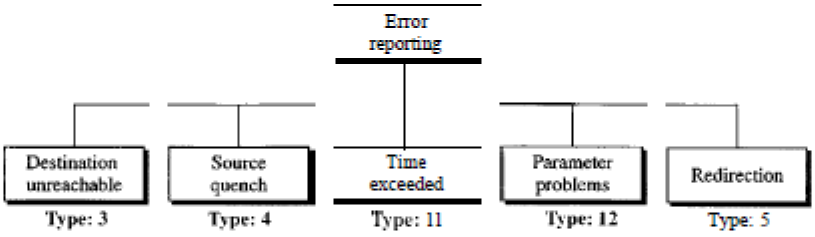
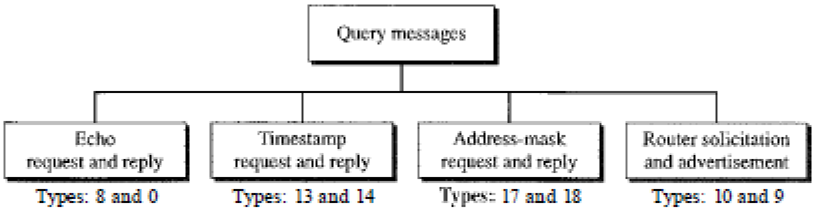


Figure 21.12 *Query messages*



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.

Source Quench:

Whenever a device is sending too much data for the destination host to process, the recipient can send an ICMP Source Quench error message back to the sender, suggesting that the sender throttle back on the rate at which it is sending data.

Time Exceeded

The time-exceeded message is generated in two cases:

- Routers use routing tables to find the next hop (next router) that must receive the packet. If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly. When the time-to-live expires, the routers discard the datagram. However, when the datagram is discarded, a time-exceeded message must be sent by the router to the original source.
- A time-exceeded message is also generated when not all fragments that make up a message arrive at the destination host within a certain time limit.

Parameter Problem

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

Redirection

A router sends a redirect error to the sender of an IP datagram when the sender should have sent the datagram to a different router or directly to an end host.

Echo request and Reply

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

Timestamp Request and Reply

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

Address-Mask Request and Reply

A host may know its IP address, but it may not know the corresponding mask. For example, a host may know its IP address as 159.31.17.24, but it may not know that the corresponding mask is /24. To obtain its mask, a host sends an address-mask-request message to a router on the LAN. If the host knows the address of the router, it sends the request directly to the router. If it does not know, it broadcasts the message. The router receiving the address-mask-request message responds with an address-mask-reply message, providing the necessary mask for the host. This can be applied to its full IP address to get its subnet address.

Router Advertisement and Solicitation

Rather than initializing a routing table with static routes, we can use the router ICMP advertisement and solicitation messages. A host can transmit a broadcast or multicast a solicitation message to know if the routers are alive and functioning. A router or routers responds with a router advertisement. This allows communicating hosts to learn of available routes dynamically and update their routing tables.