UNIT 2 INTERNET PROTOCOL

Structure **Page Nos.**

2.0 INTRODUCTION

Internet protocols were first developed in the mid-1970s, when the Defence Advanced Research Projects Agency (DARPA) became interested in establishing a packet-switched network that would facilitate communication between dissimilar computer systems at research institutions. 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 TCP/IP 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.

2.1 OBJECTIVES

Our objective is to introduce you with basic concept of Internet Protocol. On successful completion of this unit, you should be able to:

- have a reasonable understanding of the IP protocol architecture;
- desdribe the operation of IP protocol and its header format;
- understand the role and meaning of IP addressing and classes;
- describe and understand how to use subnet addressing, and
- understand the simple routing protocols.

2.2 OVERVIEW OF INTERNET PROTOCOL,

Internet Protocol is the network (internet) layer protocol used by both TCP and UDP, transport layer protocols. The entire TCP or UDP datagrams (header + payload) travel through the network as a part of the IP datagrams. TCP or UDP datagram is encapsulated in IP datagram as illustrated in following *Figure 1*. It is also clear from the figure that the application layer packet is encapsulated in transport layer datagram and network layer datagram is encapsulated in link layer frame.

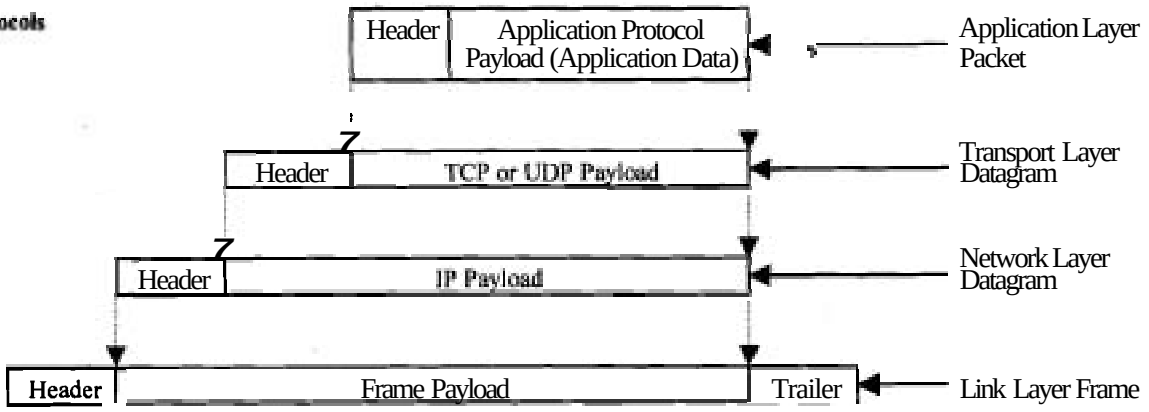

Figure 1: TCP/IP Protocols and Encapsulation

2.3 IP HEADER

The IP datagram consists of a header part and a data part. The IP header is six 32-bit words in length (24 bytes total) when all the optional fields are included in the header. The shortest header allowed by IP uses five words (20 bytes total). To understand all the fields in the header, it is useful to remember that IP has no hardware dependence but must account for all versions of IP software it can encounter (providing full backward-compatibility with previous versions of IP). The IP header layout is shown schematically in Figure 2. The different fields in the IP header are examined in more detail in the following subsections.

Version	HL	TOS	Total Length	
Identifier			Flags	Fragment Offset
TTL	Protocol		Header Checksum	
Source Address				
Destination Address				
Options				

Figure 2: IPv4 Header Format

Version Number

This is a 4-bit field that contains the IP version number the protocol software is using. The version number is required so that receiving IP software knows how to decode the rest of the header, which changes with each new release of the IP standards. The most widely used version is 4, although several systems are now testing version 6 (called IPng). The Internet and most LANs do not support IP version 6 at present.

Part of the protocol definition stipulates that the receiving software must first check the version number of incoming datagrams before proceeding to analyse the rest of the header and encapsulated data. If the software cannot handle the version used to build the datagram, the receiving machine's IP layer rejects the datagram and ignores the contents completely.

Header Length

This 4-bit field reflects the total length of the IP header built by the sending machine; it is specified in 32-bit words. The shortest header is five words (20 bytes), but the use of optional fields can increase the header size to its maximum of six words (24 bytes). To properly decode the header, IP must know when the header ends and the data begins, which is why this field is included. (There is no start-of-data marker to show where the data in the datagram begins. Instead, the header length is used to compute an offset from the start of the IP header to give the start of the data block).

Type of Service

The 8-bit (1 octet) Service Type field instructs IP how to process the datagram properly. The field's 8 bits are read and assigned as shown in ***Figure 3,*** which shows the layout of the Service Type field inside the larger IP header shown in ***Figure*** 2. The first 3 bits indicate the datagram's precedence, with a value from 0 (normal) through 7 (network control). The higher the number, the more important the datagram and, in theory at least, the faster the datagram should be routed to its destination. In practice, though, most implementations of TCP/IP and practically all hardware that use TCP/IP ignore this field, treating all datagrams with the same priority.

Figure 3: Type of Service

The next three bits are I-bit flags that control the delay, throughout, and reliability of the datagram. If the bit is set to 0, the setting is normal. A bit set to 1 implies low delay, high throughout, and high reliability for the respective flags. The last two bits of the field are not used. Most of these bits are ignored by current IP implementations, and all datagrams are treated with the same delay, throughput, and reliability settings.

For most purposes, the values of all the bits in the Service Type field are set to 0 because differences in precedence, delay, throughput, and reliability between machines are virtually nonexistent unless a special network has been established. Although these flags would be useful in establishing the best routing method for a datagram, no currently available UNIX-based IP system bothers to evaluate the bits in these fields. (Although it is conceivable that the code could be modified for high security or high reliability networks.)

'Total Length (or Packet Datagram Length)

This field gives the total length of the datagram, including the header, in octets. The length of the data area itself can be computed by subtracting the header length from this value. The size of the total datagram length field is 16 bits, hence the 65,535 bytes are the maximum length of a datagram (including the header). This field is used to determine the length value to be passed to the transport protocol to set the total packet length.

Identification

This field holds a number that is a unique identifier created by the sending node. This number is required when reassembling fragmented messages, ensuring that the fragments of one message are not intermixed with others. Each chunk of data received by the IP layer from a higher protocol layer is assigned one of these identification numbers when the data arrives. If a datagram is fragmented, each fragment has the same identification number.

Flags

l'he Flags field is a 3-bit field, the first bit of which is left unused (it is ignored by the protocol and usually has no value written to it). The remaining two bits are dedicated to flags called DF (Don't Fragment) and MF (More Fragments), which control the handling of the datagrams when fragmentation is desirable.

If the DF flag is set to 1, the datagram cannot be fragmented under any circumstances. If the current IP layer software cannot send the datagram to another machine without fragmenting it, and this bit is set to 1, the datagram is discarded and an error message is sent back to the sending device.

If the MF flag is set to I, the current datagram is followed by more fragments (sometimes called *subpackets*), which must be reassembled to re-create the full message. The last fragment that is sent as part of a larger message has its MF flag set to 0 (off) so that the receiving device knows when to stop waiting for datagrams. Because the order of the fragments' arrival might not correspond to the order in which they were sent, the MF flag is used in conjunction with the Fragment Offset field (the

next field in the IP header) to indicate to the receiving machine the full extent of the message.

Fragment Offset

If the MF (More Fragments) flag bit is set to 1 (indicating fragmentation of a larger datagram), the fragment offset contains the position of the fragments contained in the complete message within the current datagram. This enables IP to reassemble fragmented packets in the proper order.

Offsets are always given relative to the beginning of the message. This is a 13-bit field, so offsets are calculated in units of 8 bytes, corresponding to the maximum packet length of 65,535 bytes. Using the identification number to indicate which message a receiving datagram belongs to, the IP layer on a receiving machine can then use the fragment offset to reassemble the entire message.

Time to Live (TTL)

This field gives the amount of time in maximum number of hops that a datagram can remain on the network before it is discarded. This is set by the sending node when the datagram is assembled. Usually the TTL field is set to 15 or 30 hops.

The TCP/IP standards stipulate that the TTL field must be decreased by at least one hop for each node that processes the packet, even if the processing time is less. Also, when a datagram is received by a gateway, the arrival time is tagged so that if the datagram must wait to be processed, that time counts against its TTL. Hence, if a gateway is particularly overloaded and can't get to the datagram in short order, the TTL timer can expire while awaiting processing, and the datagram is abandoned.

If the TTL field reaches 0, the datagram must be discarded by the current node, but a message is sent back to the sending machine when the packet is dropped. The sending machine can then resend the datagram. The rules governing the TTL field are designed to prevent IP packets from endlessly circulating through networks.

Protocol

This field holds the identification number of the protocol to which the packet is to be handed. The numbers are defined by the Network Information Centre (NIC), which governs the Internet. There are currently about 50 protocols defined and assigned a transport protocol number. The two most important protocols are ICMP (detailed in the section titled "Internet Control Message Protocol (ICMP)" later today), which is number 1, and TCP, which is number 6. The full list of numbers is not necessary here because most of the protocols are never encountered by users.

Header Checksum

The number in this field of the IP header is a checksum for the protocol header field (but not the data fields) to enable faster processing. Because the Time to Live (TTL) field is decremented at each node, the checksum also changes with every machine the datagram passes through. The checksum algorithm takes the ones-complement of the 16-bit sum of all 16-bit words. This is a fast, efficient algorithm, but it misses some unusual corruption circumstances such as the loss of an entire 16-bit word that contains only 0s. However, because the data checksums used by both TCP and UDP cover the entire packet, these types of errors usually can be caught as the packet is assembled for the network transport.

Sending Address and Destination Address

These fields contain the 32-bit IP addresses of the sending and destination devices. These fields are established when the datagram is created and are not altered during the routing.

Options

The Options field is optional, composed of several codes of variable length. If more than one option is used in the datagram, the options appear consecutively in the IP

header. All the options are controlled by a byte that is usually divided into three fields: a I-bit copy flag, a 2-bit option class, and a 5-bit option number. The copy flag is used to stipulate how the option is handled when fragmentation is necessary in a gateway. When the bit is set to 0, the option should be copied to the first datagram but not subsequent ones. If the bit is set to 1, the option is copied to all the datagrams.

The option class and option number indicate the type of opticn and its particular value. At present, there are only two option classes set. (With only 2 bits to work with in the field, a maximum of four options could be set.) When the value is 0, the option applies to datagram or network control. A value of 2 means the option is for debugging or administration purposes. Values of 1 and 3 are unused. Currently supported values for the option class and number are given in Table 1.

Table 1: Valid option class and numbers for IP headers

Option Class	Option Number	Description
0	0	Marks the end of the options list
0	1	No option (used for padding)
0	2	Security options (military purposes only)
0	3	Loose source routing
0	7	Activates routing record (adds fields)
0	9	Strict source routing
2	4	Timestamping active (adds fields)

Of most interest to you are options that enable the routing and timestamps to be recorded. These are used to provide a record of a datagram's passage across the internetwork, which can be useful for diagnostic purposes. Both these options add information to a list contained within the datagram. (The timestamp has an interesting format: it is expressed in milliseconds since midnight, Universal Time. Unfortunately, because most systems have widely differing time settings — even when corrected to Universal Time — the timestamps should be treated with more than a little suspicion).

There are two kinds of routing indicated within the options field: loose and strict. ***Loose routing*** provides a series of IP addresses that the machine must pass through, but it enables any route to be used to get to each of these addresses (usually gateways). ***Strict routing*** enables no deviations from the specified route. If the route can't be followed, the datagram is abandoned. Strict routing is frequently used for testing routes but rarely for transmission of user datagrams because of the higher chances of the datagram being lost or abandoned.

Padding

The content of the padding area depends on the options selected. The padding is usually used to ensure that the datagram header is a round number of bytes.

2.4 IP ADDRESSES

The Internet Protocol moves data between hosts in the form of datagrams. Each datagram is delivered to the address contained in the Destination Address (word 5) of the datagram's header. The Destination Address is a standard 32-bit IP address that contains sufficient information to uniquely identify a network and a specific host on that network.

An IP address contains a network part and a host part, but the format of these parts is not the same in every IP address. The number of address bits used to identify the network, and the number used to identify the host, vary according to the prefix length of the address. There are two ways the prefix length is determined: by address class or by a CIDR address mask. We begin with a discussion of traditional IP address classes.

2.4.1 IP Address Classes

Originally, the IP address space was divided into a few fixed-length structures called address classes. The three main address classes are class A, class B, and class C. By examining the first few bits of an address, IP software can quickly determine the class,

and therefore the structure, of an address. IP follows these rules to determine the address class:

- **Class A:** If the first bit of an IP address is 0, it is the address of a class A network. The first bit of a class A address identifies the address class. The next 7 bits identify the network, and the last 24 bits identify the host. There are fewer than 128 class A network numbers, but each class A network can be composed of millions of hosts.
- **Class B:** If the first 2 bits of the address are 1 0, it is a class B network address. The first 2 bits identify class; the next 14 bits identify the network, and the last 16 bits identify the host. There are thousands of class B network numbers and each class B network can contain thousands of hosts.
- **Class C:** If the first 3 bits of the address are 1 1 0, it is a class C network address. In a class C address, the first 3 bits are class identifiers; the next 21 bits are the network address, and the last 8 bits identify the host. There are millions of class C network numbers, but each class C network is composed of fewer than 254 hosts.
- **Class D:** If the first 4 bits of the address are 1 1 1 0, it is a multicast address. These addresses are sometimes called class D addresses, but they don't really refer to specific networks. Multicast addresses are used to address groups of computers all at one time. Multicast addresses identify a group of computers that share a common application, such as a video conference, as opposed to a group of computers that share a common network.
- **Class E:** If the first four bits of the address are 1 1 1 1, it is a special reserved address. These addresses are called class E addresses, but they don't really refer to specific networks. No numbers are currently assigned in this range.

IP addresses are usually written as four decimal numbers separated by dots (periods). Each of the four numbers is in the range 0-255 (the decimal values possible for a single byte). Because the bits that identify class are contiguous with the network bits of the address, we can lump them together and look at the address as composed of full bytes of network address and full bytes of host address. If the value of the first byte is:

- Less than 128, the address is class A; the first byte is the network number, and the next three bytes are the host address.
- From 128 to 191, the address is class B; the first two bytes identify the network, and the last two bytes identify the host.
- From 192 to 223, the address is class C; the first three bytes are the network address, and the last byte is the host number.
- From 224 to 239, the address is multicast. There is no network part. The entire address identifies a specific multicast group.
- Greater than 239, the address is reserved.

The following table depicts each class range with other details.

Tabk 2: IP Address Classes in dotted decimal format with their ranges

IP Address Class	High Order Bit(s)	Format	Range	No. of Network Bits	No. of Host Bits	Max. Hosts	Purpose
A	0	N.H.H.H	1.0.0.0 to 126.0.0.0	7	24	2^{24}-2	Few large organisations
B	1,0	N.N.H.H	128.1.0.0 to 191.254.0.0	14	16	2^{16}-2	Medium-size organisations
C	1,1,0	N.N.N.H	192.0.1.0 to 223.255.254.0	21	8	2^{8}-2	Relatively small organisations
D	1,1,1,0	N/A	224.0.0.0 to 239.255.255.255	N/A	N/A	N/A	Multicast groups (RFC 1112)
E	1,1,1,1	N/A	240.0.0.0 to 254.255.255.255	N/A	N/A	N/A	Future Use (Experimental)

The IP address, which provides universal addressing across all of the networks of the Internet, is one of the great strengths of the TCP/IP protocol suite. However, the original class structure of the IP address has weaknesses. The TCPIIP designers did not envision the enormous scale of today's network. When TCPIIP was being designed, networking was limited to large organisations that could afford substantial computer systems. The idea of a powerful UNIX system on every desktop did not exist. At that time, a 32-bit address seemed so large that it was divided into classes to reduce the processing load on routers, even though dividing the address into classes sharply reduced the number of host addresses actually available for use. For example, assigning a large network a single class B address, instead of six class C addresses, reduced the load on the router because the router needed to keep only one route for that entire organisation. However, an organlsation that was given the class B address probably did not have 64,000 computers, so most of the host addresses available to the organisation were never assigned.

The class-structured address design was critically strained by the rapid growth of the Internet. At one point it appeared that all class B addresses might be rapidly ,exhausted. To prevent this, a new way of looking at IP addresses without a class structure was developed.

Check Your Progress 1

1) Internet Protocol is a _____ layer protocol.
   a) Transport
   b) Application
   c) Network
   d) Data Link

.....

2) What is the size of "type of service" field?
   a) 4 bit
   b) 5 bit
   c) 7 bit
   d) 8 bit

.....

3) If DF flag is set to 1, it means the datagram:
   a) Large and need fragmentation
   b) Small & need reassembling
   c) Cannot be fragmented
   d) Is accepted and message is sent back to sending device

.....

2.4.2 Subnet Masks and CIDR Networks (Classless IP Addresses)

IP addresses are actually 32-bit binary numbers. Each 32-bit IP address consists of two subaddresses, one identifying the network and the other identifying the host to the network, with an imaginary boundary separating the two. The location of the boundary between the network and host portions of an IP address is determined through the use of a subnet mask. A subnet mask is another 32-bit binary number, which acts like a filter when it is applied to the 32-bit IP address. By comparing a subnet mask with an IP address, systems can determine which portion of the IP address relates to the network, and which portion relates to the host. Anywhere the subnet mask has a bit set to "1", the underlying bit in the IP address is part of the network address. Anywhere the subnet mask is set to "0", the related bit in the IP address is part of the host address. For example, assume that the IP address 11000000101010000000000010001010 0 has a subnet mask of 11111111111111111111111100000000. In this example, the first 24 bits of the 32-bit IP addresses are used to identify the network, while the last 8 bits are used to identify the host on that network.

The size of a network (i.e., the number of host addresses available for use on it) is a function of the number of bits used to identify the host portion of the address. If a subnet mask shows that 8 bits are used for the host portion of the address block, a maximum of 256 possible host addresses are available for that specific network. Similarly, if a subnet mask shows that 16 bits are used for the host portion of the address block, a maximum of 65,536 possible host addresses are available for use on that network.

If a network administrator needs to split a single network into multiple virtual networks, the bit-pattern in use with the subnet mask can be changed to allow as many networks as necessary. For example, assume that we want to split the 24-bit 192.168.10.0 network (which allows for 8 bits of host addressing, or a maximum of 256 host addresses) into two smaller networks. All we have to do in this situation is to change the subnet mask of the devices on the network so that they use 25 bits for the network instead of 24 bits, resulting in two distinct networks with 128 possible host addresses on each network. In this case, the first network would have a range of network addresses between 192.168.10.0 -192.168.10.127, while the second network would have a range of addresses between 192.168.10.128 -192.168.10.255.

Networks can also be enlarged through the use of a technique known as "**supernetting**," which works by extending the host portion of a subnet mask to the left, into the network portion of the address. Using this technique, a pair of networks with 24-bit subnet masks can be turned into a single large network with a 23-bit subnet mask. However, this works only if you have two neighbouring 24-bit network blocks, with the lower network having an even value (when the network portion of the address is shrunk, the trailing bit from the original network portion of the subnet mask will fall into the host portion of the new subnet mask, so the new network mask will consume both networks). For example, it is possible to combine the 24-bit 192.168.10.0 and 192.168.11.0 networks together since the loss of the trailing bit from each network (00001010 vs. 00001011) produces the same 23-bit subnet mask (0000101x), resulting in a consolidated 192.168.10.0 network. However, it is not possible to combine the 24-bit 192.168.11.0 and 192.168.12.0 networks, since the binary values in the seventh bit position (00001011 vs. 00001100) do not match when the trailing bit is removed.

Classless Inter-Domain Routing

In the modern networking environment defined by RFC 1519 [Classless Inter-Domain Routing (CIDR)], the subnet mask of a network is typically annotated in written form as a "slash prefix" that trails the network number. In the subnetting example in the previous paragraph, the original 24-bit network would be written as 192.168.10.0/24, while the two new networks would be written as 192.168.10.0/25 and 192.168.10.128/25. Likewise, when the 192.168.10.0/24 and 192.168.11.0/24 networks were joined together as a single supernet, the resulting network would be written as 192.168.10.0/23. Note that the slash prefix annotation is generally used for human benefit; infrastructure devices still use the 32-bit binary subnet mask internally to identify networks and their routes. All networks must reserve host addresses that are made up entirely of either ones or zeros, to be used by the networks themselves. This is so that each subnet will have a network-specific address (the all-zeroes address) and a broadcast address (the all-ones address). For example, a /24 network allows for 8 bits of host addresses, but only 254 of the 256 possible addresses are available for use. Similarly, /25 networks have a maximum of 7 bits for host addresses, with 126 of the 128 possible addresses available (the all-ones and all-zeroes addresses from each subnet must be set aside for the subnets themselves). All the systems on the same subnet must use the same subnet mask in order to communicate with each other directly. If they use different subnet masks they will think they are on different networks, and will not be able to communicate with each other without going through a router first. Hosts on different networks can use different subnet maks, although the routers will have to be aware of the subnet masks in use on each of the segments.

Subnet masks are used only by systems that need to communicate with the network directly; For example, external systems do not need to be aware of the subnet masks in use on your internal networks, since those systems will route data to your networks by way of your parent network's address block. As such, remote routers need to know only about your provider's subnet mask. For example, if you have a small network that uses only a /28 prefix that is a subset of your ISP's 120 network, remote routers need to know only about your upstream provider's 120 network, while your upstream provider needs to know your subnet mask in order to get the data to your local /28 network. The rapid depletion of the class B addresses showed that three primary address classes were not enough: class A was much too large and class C was much too small. Even a class B address was too large for many networks but was used because it was better than the alternatives.

The obvious solution to the class B address crisis was to force organisations to use multiple class C addresses. There were millions of these addresses available and they were in no immediate danger of depletion. As is often the case, the obvious solution is not as simple as it may seem. Each class C address requires its own entry within the routing table. Assigning thousands or millions of class C addresses would cause the routing table to grow so rapidly that the routers would soon be overwhelmed. The solution require a new way of assigning addresses and a new way of looking at addresses.

Originally network addresses were assigned in more or less sequential order as they were requested. This worked fine when the network was small and centralised. However, it did not take network topology into account. Thus only random chance would determine if the same intermediate routers would be used to reach network 195.4.12.0 and network 195.4.13.0, which makes it difficult to reduce the size of the routing table. Addresses can only be aggregated if they are contiguous numbers and are reachable through the same route. For example, if addresses are contiguous for one service provider, a single route can be created for that aggregation because that service provider will have a limited number of routes to the Internet. But if one network address is in France and the next contiguous address is in Australia, creating a consolidated route for these addresses does not work.

Today, large, contiguous blocks of addresses are assigned to large network service providers in a manner that better reflects the topology of the network. The service providers then allocate chunks of these address blocks to the organisations to which they provide network services. This alleviates the short-term shortage of class B addresses and, because the assignment of addressees reflects the topology of the network, it permits route aggregation. Under this new scheme, we know that network 195.4.12.0 and network 195.4.13.0 are reachable through the same intermediate routers. In fact, both of these addresses are in the range of the addresses assigned to Europe, 194.0.0.0 to 195.255.255.255. Assigning addresses that reflect the topology of the network enables route aggregation, but does not implement it. As long as network 195.4.1 2.0 and network 195.4.13.0 are interpreted as separate class C addresses, they will require separate entries in the routing table. A new, flexible way of defining addresses is needed.

Evaluating addresses according to the class rules discussed above limits the length of network numbers to 8, 16, or 24 bits - 1, 2, or 3 bytes. The IP address, however, is not really byte-oriented. It is 32 contiguous bits. A more flexible way fo interpret the network and host portions of an address is with a bit mask. An address bit mask works in this way: if a bit is on in the mask, that equivalent bit in the address is interpreted as a network bit; if a bit in the mask is off, the bit belongs to the host part of the address. For example, if address 195.4.12.0 is interpreted as a class C address, the first 24 bits are the network numbers and the last 8 bits are the host addresses. The network mask that represents this is 255.255.255.0, 24 bits on and 8 bits off. The bit mask that is derived from the traditional class structure is called the default mask or the natural mask.

However, with bit masks we are no longer limited by the address class structure. A mask of 255.255.0.0 can be applied to network address 195.4.0.0. This mask includes

all addresses from 195.4.0.0 to 195.4.255.255 in a single network number. In effect, it creates a network number as large as a class B network in the class C address space. Using bit masks to create networks larger than the natural mask is called supernetting, and the use of a mask instead of the address class to determine the destination network is called Classless Inter-Domain Routing (CIDR).

Specifying both the address and the mask is cumbersome when writing out addresses. A shorthand notation has been developed for writing CIDR addresses. Instead of writing network 172.16.26.32 with a mask of 255.255.255.224, we can write 172.16.26.32127. The format of this notation is addresslprefix-length, where prefix-length is the number of bits in the network portion of the address. Without this notation, the address 172.16.26.32 could easily be interpreted as a host address. RFC 1878 list all 32 possible prefix values. But little documentation is needed because the CIDR prefix is much easier to understand and remember than are address classes. I know that 10.104.0.19 is a class A address, but writing it as 10.104.0.19/8 shows me that this address has 8 bits for the network number and therefore 24 bits for the host number. I don't have to remember anything about the class A address structure.

2.4.3 Internet-Legal Versus Private Addressing

Although the pool of IP addresses is somewhat limited, most companies have no problems obtaining them. However, many organisations have already installed TCP/IP products on their internal networks without obtaining "legal" addresses from the proper sources. Sometimes these addresses come from example books or are simply picked at random (several firms use networks numbered 1.2.3.0, for example). Unfortunately, since they are not legal, these addresses will not be usable when these organisations attempt to connect to the Internet. These firms will eventually have to reassign Internet-legal IP addresses to all the devices on their networks, or invest in address-translation gateways that rewrite outbound IP packets so they appear to be coming from an Internet-accessible host.

Even if an address-translation gateway is installed on the network, these firms will never be able to communicate with any site that is a registered owner of the IP addresses in use on the local network. For example, if you choose to use the 36.0.0.018 address block on your internal network, your users will never be able to access the computers at Stanford University, the registered owner of that address block. Any attempt to connect to a host at 36.x.x.x will be interpreted by the local routers as a request for a local system, so those packets will never leave your local network.

Not all firms have the luxury of using Internet-legal addresses on their hosts, for any number of reasons. For example, there may be legacy applications that use hardcode addresses, or there may be too many systems across the organisation for a clean upgrade to be successful. If you are unable to use Internet-legal addresses, you should at least be aware that there are groups of "private" Internet addresses that can be used on internal networks by anyone. These address pools were set-aside in RFC 1918, and therefore cannot be "assigned" to any organization. The Internet's backbone routers are configured explicitly not to route packets with these addresses, so they are completely useless outside an organization's internal network. The address blocks available are listed in Table 3.

Table 3: Private Addresses Provided in RFC 1918

Class	Range of Addresses
A	Any addresses in 10.x.x.x
B	Addresses in the range of 172.16.x.x-172.31.x.x
C	Addresses in the range of 192.168.0.x-192.168.255.x

Since these addresses cannot be routed across the Internet, you must use an .address-translation gateway or a proxy server in conjunction with them. Otherwise, you will not be able to communicate with any hosts on the Internet.

An important note here is that since nobody can use these addresses on the Internet, it is safe to'assume that anybody who is using these addresses is also utilising an

address-translation gateway of some sort. Therefore, while you will never see these addresses used as destinations on the Internet, if your organisation establishes a private connection to a partner organisation that is using the same block of addresses that you are using, your firms will not be able to communicate. The packets destined for your partner's network will appear to be local to your network, and will never be forwarded to the remote network.

There are many other problems that arise from using these addresses, making their general usage difficult for normal operations. For example, many application-layer protocols embed addressing information directly into the protocol stream, and in order for these protocols to work properly, the address-translation gateway has to be aware of their mechanics. In the preceding scenario, the gateway has to rewrite the private addresses (which are stored as application data inside the application protocol), rewrite the UDP/TCP and IP checksums, and possibly rewrite TCP sequence numbers as well. This is difficult to do even with simple and open protocols such as FTP, and extremely difficult with proprietary, encrypted, or dynamic applications (these are problems for many database protocols, network games, and voice-over-IP services, in particular). These gateways almost never work for all the applications in use at a specific location.

It is always best to use formally-assigned, Internet-legal addresses whenever possible, even if the hosts on your network do not necessarily require direct Internet access. In those cases in which your hosts are going through a firewall or application proxy of some sort, the use of Internet-legal addresses causes the least amount of maintenance trouble over time. If for some reason this is not possible, use one of the private address pools described in Table 3. Do not use random, self-assigned addresses if you can possibly avoid it, as this will only cause connectivity problems for you and your users.

Check Your Progress 2

1) If the first bit of an IP address is zero, then it belongs to
   a) Class A
   b) Class B
   c) Class C
   d) Class D

.....

2) The Internet.backbone renters are configured explicitly not to route packets with _____ address.
   a) Legal
   b) Private
   c) Public
   d) Virtual

.....

3) _____ Works by extending the host portion of a subnet mask to the left, into the network portion of the address.
   a) Subnetting
   b) Supernetting
   c) Classless addressing
   d) Hyper testing

.....

2.5 IP ROUTING

An important function of the IP layer is *IP* routing. It provides the basic mechanism for routers to interconnect different physical networks. This means that an internet host can function as a normal host and a router simultaneously.

A basic router of this type is referred to as a router with partial routing information, because the router only has information about four kinds of destination:

- Hosts which are directly attached to one of the physical networks to which the router is attached.
- Hosts or networks for which the router has been given explicit definitions.
- Hosts or networks for which the router has received an ICMP redirect message.
- A default destination for everything else.

The last two items allow a basic router to begin with a very limited amount of information and to increase its information because a more sophisticated router will issue an ICMP redirect message if it receives a datagram and it knows of a better router on the same network for the sender to use. This process is repeated each time a basic router of this type is restarted.

Additional protocols are needed to implement a full-function router that can exchange information with other routers in remote network. Such routers are essential except in small networks, and the protocols they use are discussed in Routing Protocol.

2.5.1 Routing Protocol

The Routing protocols evolved in different phases. There are some references are as below:

- RFC 1074 - The NSFNET Backbone SPF Based Interior Gateway Protocol.
- RFC 1092 - EGP and Policy Based Routing in the New NSFNET Backbone.
- RFC 1093 - The NSFNET Routing Architecture.
- RFC 1104 - Models of Policy Based Routing.
- RFC 1133 - Routing between the NSFNET and the DDN.
- RFC 1222 - Advancing the NSFNET Routing Architecture

These protocols use two important groups of routing algorithms.

2.5.2 Routing Algorithms

In this section, we discuss the Distance-Vector and Link-State, Shortest Path First (SPF) Routing Algorithms.

Distance-Vector

The term Vector-Distance refers to a class of algorithms that gateways use to update routing information. Each router begins with a set of routes for those networks or subnets to which it is directly attached, and possibly some additional routes to other networks or hosts if the network topology is such that the routing protocol will be unable to produce the desired routing correctly. This list is kept in a routing table, where each entry identifies a destination network or host and gives the "distance" to that network. The distance is called a *metric* and is typically measured in "hops".

Periodically, each router sends a copy of its routing table to any other router it can reach directly. When a report arrives at router B from router A, B examines the set of destinations it receives and the distance to each. B will update its routing table if:

- A knows a shorter way to reach a destination.
- A lists a destination that B does not have in its table.
- A's distance to a destination, already routed through A from B, has changed.

This kind of algorithm is easy to implement, but it has a number of disadvantages:

- When routes change rapidly, that is, a new connection appears or an old one fails, the routing topology may not stabilize to match the changed network topology because information propagates slowly from one router to another and while it is propagating, some routers will have incorrect routing information.
- Another disadvantage is that each router has to send a copy of its entire routing table to every neighbour at regular intervals. Of course, one can use longer

intervals to reduce the network load but that introduces problems related to how well the network responds to changes in topology.

- Vector-distance algorithms using hop counts as a metric does not take account of the link speed or reliability. Such an algorithm will use a path with hop count 2 that crosses two slow-speed lines, instead of using a path with hop count 3 that crosses three token-rings and may be substantially faster.

The most difficult task in a distance vector algorithm is to prevent instability. Different solutions are available:

Counting to infinity

Let us choose a value of 16 to represent infinity. Suppose a network becomes inaccessible, all the immediately neighbouring routers time out and set the metric to that network to 16. We can consider that all the neighbouring routers have a piece of hardware that connects them to the vanished network, with a cost of 16. Since that is the only connection to the vanished network, all the other routers in the system will converge to new routes that go through one of those routers with a direct but unavailable connection. Once convergence has happened, all the routers will have metrics of 16 for the vanished network. Since 16 indicates infinity, all routers then regard the network as unreachable.

Link-State, Shortest Path First

The growth in networking over the past few years has pushed the currently available Interior Gateway Protocols, which use distance-vector algorithms, past their limits.

The primary alternative to vector-distance schemes is a class of protocols known as Link State, Shortest Path First.

The important features of these routing protocols are:

- A set of physical networks is divided into a number of areas.
- All routers within an area have an identical database.
- Each router's database describes the complete topology (which routers are connected to which networks) of the routing domain. The topology of an area is represented with a database called a Link State Database describing all of the links that each of the routers in the area has.
- Each router uses its database to derive the set of optimum paths to all destinations from which it builds its routing table. The algorithm used to determine the optimum paths is called a Shortest Path First (SPF) algorithm.

In general, a link state protocol works as follows each router periodically sends out a description of its connections (the state of its links) to its neighbours (routers are neighbours if they are connected to the same network). This description, called a Link State Advertisement (LSA), includes the configured cost of the connection. The LSA is flooded throughout the router's domain. Each router in the domain maintains an identical synchronised copy of a database composed of this link state information. This database describes both the topology of the router's domain and routes to networks outside of the domain such as routes to networks in other autonomous systems. Each router runs an algorithm on its topological database resulting in a shortest-path tree. This shortest-path tree contains the shortest path to every router and network the gateway can reach. From the shortest-path tree, the cost to the destination and the next hop to forward a datagram is used to build the router's routing table. Link-state protocols, in comparison with vector-distance protocols, send out updates when there is news, and may send out regular updates as a way of ensuring neighbour routers that a connection is still active. More importantly, the information exchanged is the state of a router's links, not the contents of the routing table. This means that link-state algorithms use fewer network resources than their vector-distance counterparts, particularly when the routing is complex or the autonomous system is large. They are, however, compute-intensive. In return, users get faster response to network events, faster route convergence, and access to more advanced network services.

Check Your Progress 3

1) Which of the following device provides the basic mechanism for interconnect different physical networks.

a) Gateways
b) Bridges
c) Routers
d) Hubs

.....

2) _____ routing algorithm all routers within an area have an identical database.

a) Distance-vector
b) Link state
c) Longest-path
d) Shortest path

.....

3) _____ Routing algorithms use features network resources than the distance-vector routing.

a) Link-state
b) Policy based
c) Connection based
d) Hops based

.....

2.6 SUMMARY

In this unit you have studied the architecture of the Internet protocol, you must have learned different fields of IP header and their role in network communication. You learned about the different IP classes and their use in network design. You also studied about the practical issues of subnettting, supernetting and the Classless IP Addresses. Further in this unit we have given you comparisons between Internet-legal addressing and private addressing. In the end of this unit we have explained the procedure of IP routing with the help of Distance-Vector and Link-State routing algorithms. The next unit of this course covers about the transport layer and its role in TCP/IP.

2.7 SOLUTIONS/ANSWERS

Check Your Progress 1

1) C 2) D 3) C

Check Your Progress 2

1) A 2) B 3) B

Check Your Progress 3

1) C 2) B 3) A

2.8 FURTHER READINGS

1) Achyut S. Godbole, Web Technoligies TATA McGrawHill, 2003.

2) Berhouz Forouzan, *TCP/IP* Protocol Suite, 3rd edition, TATA McGraw Hill, 2006.

3) http://www.tcpipguide.com.

4) http://www.cisco.com.