

Week 6

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

Service provide by the network layer

1. Logical Addressing
2. Routing

The Internet protocol

The physical and data link layers of a network operate locally. These two layers are jointly responsible for data delivery on the network from one node to the next, when we want to transmit the data through various links or to solve the problem of delivery through several links, the network layer (or the internetwork layer, as it is sometimes called) was designed. The network layer is responsible for host-to-host delivery and for routing the packets through the routers or switches.

Internet as a Datagram Network

The Internet has chosen the datagram approach to switching in the network layer.

It uses the universal addresses defined in the network layer to route packets from the source to the destination.

Internet as a Connectionless Network

Delivery of a packet can be accomplished by using either a connection-oriented or a Connectionless network service. In a connection-oriented service, the source first makes a connection with the destination before sending a packet. When the connection is established, a sequence of packets from the same source to the same destination can be sent one after another. In this case, there is a relationship between packets. In connectionless service, the network layer protocol treats each packet independently, with each packet having no relationship to any other packet. The packets in a message may not travel the same path to their destination. This type of service is used in the datagram approach to packet switching. The Internet has chosen this type of service at the network layer. The reason for this decision is that the Internet is made of so many heterogeneous networks that it is almost impossible to create a connection from the source to the destination without knowing the nature of the networks in advance

IPv4

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP

Protocols. IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. The term *best-effort* means that IPv4 provides

no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. If reliability is important, IPv4 must be paired with a reliable protocol such as TCP. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would-be recipient to discover the loss and rectify the problem. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage. IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram. This means that each datagram is handled independently, and each datagram can follow a different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order. Also, some could be lost or corrupted during transmission. Again, IPv4 relies on a higher-level protocol to take care of all these problems.

IPv4 Datagram Format

Packets in the IPv4 layer are called datagrams. Figure shows the IPv4 datagram format.

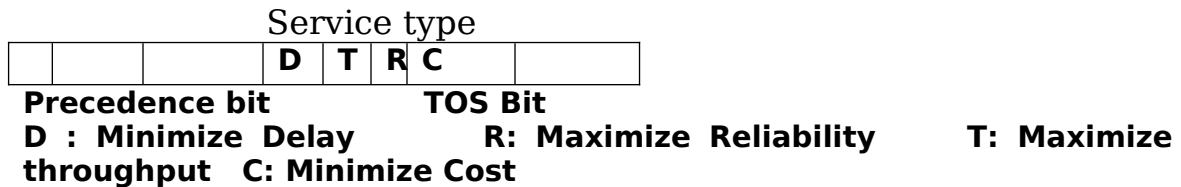
Header 20 to 60 bytes			Data 20 - 65536 bytes	
VER 4 bits	HLEN 4 Bits	Services 8 bits	Total Length 16 bit	
Identification 16 bits			Flag 3 Bits	Fragmentation Offset 13 bits
Time to live 8 bits	Protocol 8bits		Header Checksum 16 bits	
Source IP Address 32 Bits				
Destination IP Address 32 bits				
Optional				

A datagram is a variable-length packet consisting of two parts: header and data.

The header is 20 to 60 bytes in length and contains information essential to routing and delivery. It is customary in *TCP/IP* to show the header in 4-byte sections. A brief description of each field is in order.

1. **Version (VER).** This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. However, version 6 (or IPv6) may totally replace version 4 in the future. This field tells the IPv4 software running in the processing machine that the datagram has the format of version 4.

2. **Header length (HLEN).** This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 ($5 \times 4 = 20$). When the option field is at its maximum size, the value of this field is 15 ($15 \times 4 = 60$).
3. **Services.** IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services.



Service Type In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are

Called type of service (TOS) bits, and the last bit is not used.

Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first.

A TOS bit is a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. The bit patterns and their interpretations are given in Table. With only 1 bit set at a time, we can have five different types of services.

TOS Bits	Description
0000	Normal Default
0001	Minimize cost
0010	Maximize Reliability
0100	Maximize Throughput
1000	Minimize Delay

4. **Total length.** This is a 16-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

Length of data = total length - header length. Since the field length is 16 bits, the total length of the IPv4 datagram is limited to 65,535 ($2^{16} - 1$) bytes, of which 20 to 60 bytes are the header and the rest is data from the upper layer.

5. **Time to live.** A datagram has a limited lifetime in its travel through an internet. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was

discarded when the value became zero. However, for this scheme, all the machines must have synchronized clocks and must know how long it takes for a datagram to go from one machine to another. Today, this field is used mostly to control the maximum number of hops (routers) visited by the datagram.

6. **Protocol.** This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered.
7. **Source address.** This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
8. **Destination address.** This 32-bit field defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
9. **Fragmentation** we must divide the datagram to make it possible to pass through these networks. This is called fragmentation. The source usually does not fragment the IPv4 packet. The transport layer will instead segment the data into a size that can be accommodated by IPv4 and the data link layer in use. When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but with some changed. A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU. In other words, a datagram can be fragmented several times before it reaches the final destination. In IPv4, a datagram can be fragmented by the source host or any router in the path although there is a tendency to limit fragmentation only at the source. The reassembly of the datagram, however, is done only by the destination host because each fragment becomes an independent datagram. Whereas the fragmented datagram can travel through different routes, and we can never control or guarantee which route a fragmented datagram may take, all the fragments belonging to the same datagram should finally arrive at the destination host. So it is logical to do the reassembly at the final destination.
10. **Identification.** This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host. To guarantee uniqueness, the IPv4 protocol uses a counter to label the datagrams. The counter is initialized to a positive number. When the IPv4 protocol sends a datagram, it copies the current value of the counter to the identification field and increments the counter by 1. As long as the counter is kept in the main memory, uniqueness is guaranteed. When a datagram is fragmented, the value in the identification field is copied to all fragments. In other words, all fragments have the same identification number, the same as the

original datagram. The identification number helps the destination in reassembling the datagram.

11. **Flags.** This is a 3-bit field. The first bit is reserved. The second bit is called the *do not fragment* bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary. The third bit is called the *more fragment* bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.
12. **Fragmentation offset.** This 13-bit field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes. Figure 20.11 shows a datagram with a data size of 4000 bytes fragmented into three fragments. The bytes in the original datagram are numbered 0 to 3999. The first fragment carries bytes 0 to 1399. The offset for this datagram is $0/8=0$ the second fragment carries bytes 1400 to 2799; the offset value for this fragment is $1400/8 = 175$. Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is $2800/8 = 350$.
13. **Checksum** The checksum in the IPv4 packet covers only the header, not the data. There are two good reasons for this. First, all higher-level protocols that encapsulate data in the IPv4 datagram have a checksum field that covers the whole packet. Therefore, the checksum for the IPv4 datagram does not have to check the encapsulated data. Second, the header of the IPv4 packet changes with each visited router, but the data do not. So the checksum includes only the part that has changed. If the data were included, each router must recalculate checksum for the whole packet, which means an increase in processing time.
14. **Option** The header of the IPv4 datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long and was discussed in the previous section. The variable part comprises the options that can be a maximum of 40 bytes. Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging. Although options are not a required part of the IPv4 header, option processing is required of the IPv4 software. This means that all implementations must be able to handle options if they are present in the header.

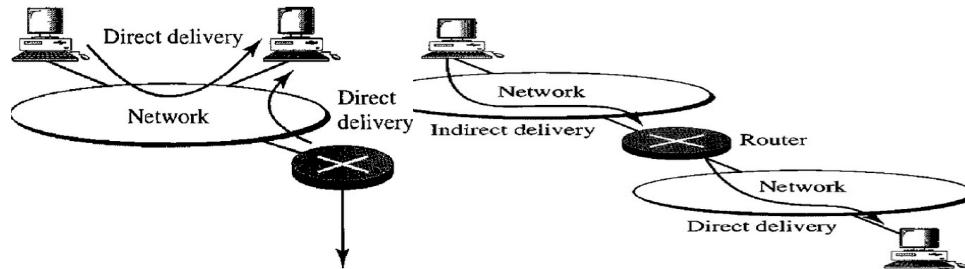
Delivery The network layer supervises the handling of the packets by the underlying physical networks. We define this handling as the delivery of a packet.

Direct Delivery

In a direct delivery, the final destination of the packet is a host connected to the same physical network as the deliverer. Direct delivery occurs when the source and destination of the packet are located on the same physical network or when the delivery is between the last router and the destination host.

Indirect Delivery

If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. In an indirect delivery, the packet goes from router to router until it reaches the one connected to the same physical network as its final destination.



Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination. However, this simple solution is impossible today in an internetwork such as the Internet because the number of entries needed in the routing table would make table lookups inefficient.

Routing Table A host or a router has a routing table with an entry for each destination, or a combination of destinations, to route IP packets. The routing table can be either static or dynamic.

Static Routing Table

A **static routing table** contains information entered manually. The administrator enters the route for each destination into the table. When a table is created, it cannot update automatically when there is a change in the Internet. The table must be manually altered by the administrator.

A static routing table can be used in a small internet that does not change very often, or in an experimental internet for troubleshooting.

Dynamic Routing Table

A dynamic routing table is updated periodically by using one of the dynamic routing protocols such as RIP, OSPF, or BGP. Whenever there is a change in the Internet, such as a shutdown of a router or breaking of a link, the dynamic routing protocols update all the tables in the routers (and eventually in the host) automatically. The routers in a big internet such as the Internet need to be updated dynamically for efficient delivery of the IP packets.

Format a routing table for classless addressing has a minimum of four columns. However, some of today's routers have even more columns. We should be aware that the number of columns is vendor-dependent, and not all columns can be found in all routers.

Mask	Network Address	Next Hop Address	Interface	Flag	Reference Count	Use

1. **Mask.** This field defines the mask applied for the entry.
2. **Network address.** This field defines the network address to which the packet is finally delivered. In the case of host-specific routing, this field defines the address of the destination host.
3. **Next-hop address.** This field defines the address of the next-hop router to which the packet is delivered.
4. **Interface.** This field shows the name of the interface or the network.
5. **Flags.** This field defines up to five flags. Flags are on/off switches that signify either presence or absence. The five flags are U (up), G (gateway), H (host-specific), D (added by redirection), and M (modified by redirection).

a. U (up). The U flag indicates the router is up and running. If this flag is not present, it means that the router is down. The packet cannot be forwarded and is discarded.

b. G (gateway). The G flag means that the destination is in another network. The packet is delivered to the next-hop router for delivery (indirect delivery). When this flag is missing, it means the destination is in this network (direct delivery).

c. H (host-specific). The H flag indicates that the entry in the network address field is a host-specific address. When it is missing, it means that the address is only the network address of the destination.

d. D (added by redirection). The D flag indicates that routing information for this destination has been added to the host routing table by a redirection message from ICMP.

e. M (modified by redirection). The M flag indicates that the routing information for this destination has been modified by a redirection message from ICMP.

6. Reference count. This field gives the number of users of this route at the moment. For example, if five people at the same time are connecting to the same host from this router, the value of this column is 5.

7. Use. This field shows the number of packets transmitted through this router for the corresponding destination.

Intra- and Interdomain Routing

Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems. An autonomous system (AS) is

a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as intradomain routing. Routing between autonomous systems is referred to as interdomain routing. Each autonomous system can choose one or more intradomain routing protocols to handle routing inside the autonomous system. However, only one interdomain routing protocol handles routing between autonomous systems. Intra-domain protocol:-RIP, OSPF, Inter-domain protocol: - BGP

