Chapter 5. The Network Layer

- The network layer is responsible for packet forwarding including routing through intermediate routers.
- To achieve its goals, the network layer must know about the topology of the communication subnet (i.e., the set of all routers) and choose appropriate paths through it.
- It must also take care to choose routes to avoid overloading some of the communication lines and routers while leaving others idle.
- Finally, when the source and destination are in different networks, new problems occur. It is up to the network layer to deal with them.

5.1 Network Layer Design Issues

Store-and-Forward Packet Switching

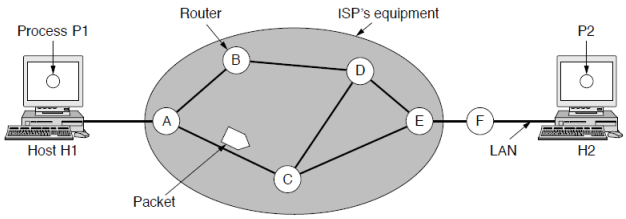


Figure 5-1. The environment of the network layer protocols.

- A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the carrier.
- The packet is stored there until it has fully arrived so the checksum can be verified.
- Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered.
- This mechanism is store-and-forward packet switching.

Services Provided to the Transport Layer

- The network layer provides services to the transport layer at the network layer/transport layer interface.
- The network layer services have been designed with the following goals in mind.

1. The services should be independent of the router technology.

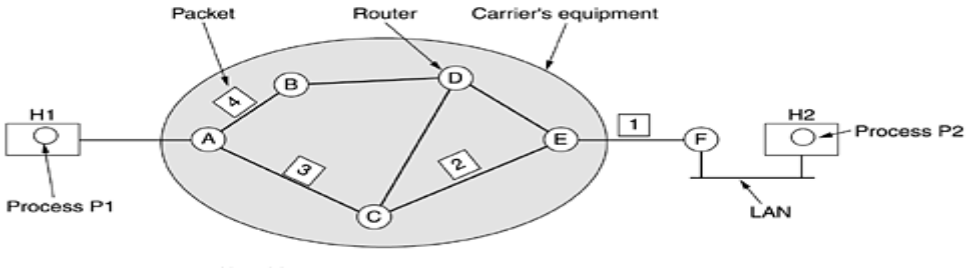
2. The transport layer should be shielded from the number, type, and topology of the routers present.

3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

- The network layer provides **connection-oriented** service or **connectionless** service.

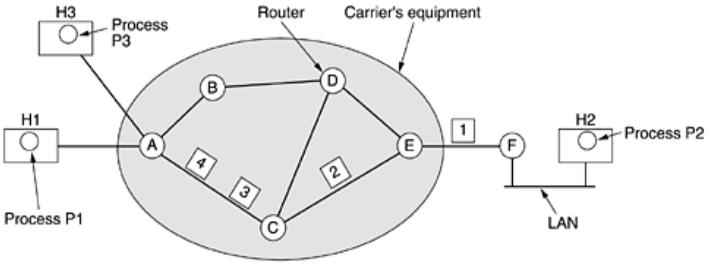
Implementation of Connectionless Service

Routing within a datagram subnet.



- In connectionless service, packets are injected into the subnet individually and routed independently of each other.
- No advance setup is needed. In this context, the packets are frequently called **datagrams** (in analogy with telegrams) and the subnet is called a **datagram subnet.**
- The algorithm that makes the routing decisions is called the **routing algorithm.**

Implementation of Connection-Oriented Service



- In connection-oriented service, a path from the source router to the destination router must be established before any data packets can be sent.
- All packets are routed through same path.
- This connection is called a **VC (virtual circuit),** in analogy with the physical circuits set up by the telephone system, and the subnet is called a **virtual-circuit subnet.**

5.2 Routing Algorithms in a single network

- The main function of the network layer is routing packets from the source machine to the destination machine.
- In most subnets, packets will require multiple hops to make the journey.
- The **routing algorithm** is that part of the network layer software responsible for deciding in which output line a packet should be transmitted.
- If the subnet uses **datagrams** internally, this decision must be made for every arriving data packet.
- If the subnet uses **virtual circuits** internally, routing decisions are made only when a new virtual circuit is being set up.
- A router having two processes inside it.

- One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing tables. This process is **forwarding**.
- The other process is responsible for filling in and updating the routing tables.

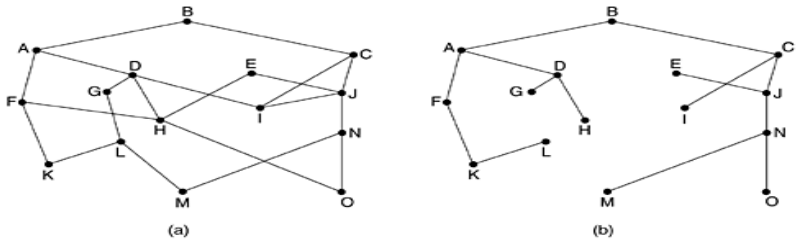
- Certain properties are desirable in a routing algorithm: **correctness, simplicity, robustness, stability, fairness, and optimality.**
- Routing algorithms can be grouped into two major classes: **nonadaptive and adaptive**.
- **Nonadaptive algorithms** do not base their routing decisions on measurements or estimates of the current traffic and topology.
- The choice of the route is computed in advance. This procedure is sometimes called **static routing**.
- **Adaptive algorithms** change their routing decisions based on measurements or estimates of the current traffic and topology.
- **Adaptive algorithms**, change their routing decisions to reflect changes in the topology.

5.2.1 The Optimality Principle

- **Optimality principle:** statement about optimal routes without regard to network topology or traffic
- It states that if router *J* is on the optimal path from router *I* to router *K*, then the optimal path from *J* to *K* also falls along the same route.
- As a direct consequence of the optimality principle, the set of optimal routes from all sources to a given destination form a **tree** rooted at the destination.
- Such a tree is called a **sink tree**, where the distance metric is the **number of hops**.
- The goal of all routing algorithms is to discover and use the sink trees for all routers.

Figure 5-6. (a) A subnet. (b) A sink tree for router B.

- Since a sink tree is indeed a tree, it **does not contain any loops**, so each packet will be delivered within a finite and bounded number of hops.

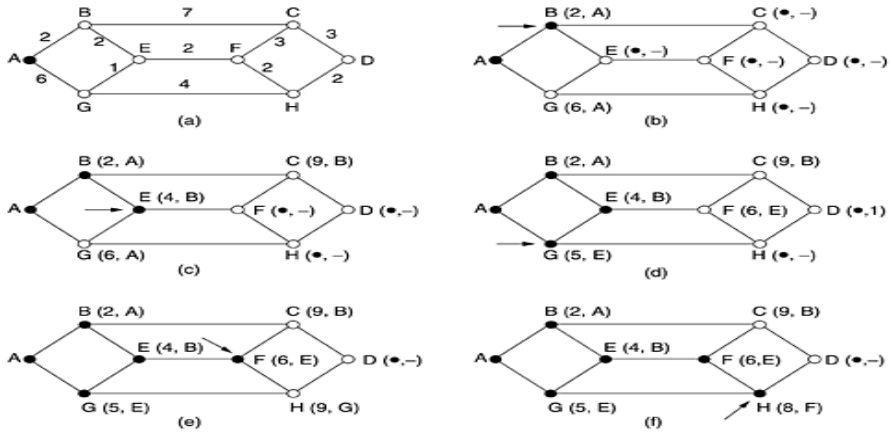


5.2.2 Shortest Path Routing

- The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line (often called a link).
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- **One way of measuring shortest path length is the number of hops.**

- Using this metric, the paths *ABC* and *ABE* in Fig. 5-7 are equally long.
- **Another metric is the geographic distance in kilometers.**
- In this case *ABC* is clearly much longer than *ABE*.
- However, many other metrics besides hops and physical distance are also possible.
- For example, each arc could be labeled with the **mean queueing and transmission delay** for some standard test packet as determined by hourly test runs.
- **Another metric is the fastest path** rather than the path with the fewest hops or arcs or kilometers.

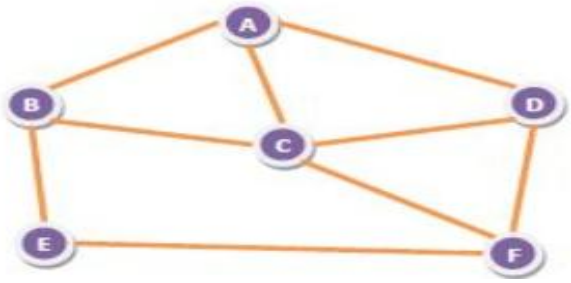
Figure 5-7. The first five steps used in computing the shortest path from A to D. The arrows indicate the working node.



- In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors.
- By changing the weighting function, the algorithm would then compute the "shortest" path measured according to any one of a number of criteria or to a combination of criteria.
- Several algorithms for computing the shortest path between two nodes of a graph are known. This one is due to **Dijkstra** (1959).
- **Dijkstra algorithm- shortest path algorithm**
- Each node is labeled (in parentheses) with its distance from the source node along the best known path. Initially, no paths are known, so all nodes are labeled with infinity.
- As the algorithm proceeds and paths are found, the labels may change, reflecting better paths. A label may be either tentative or permanent.
- Initially, all labels are tentative. When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

5.2.3 Flooding

- Another static algorithm is **flooding**, in which every incoming packet is sent out on every outgoing line except the one it arrived on.



- Using flooding technique −
  - An incoming packet to A, will be sent to B, C and D.
  - B will send the packet to C and E.
  - C will send the packet to B, D and F.
  - D will send the packet to C and F.
  - E will send the packet to F.
  - F will send the packet to C and E.

Limitations of Flooding

- **Flooding generates vast numbers of duplicate packets**, in fact, an infinite number unless some measures are taken to damp the process.
- **It is wasteful if a single destination needs the packet**, since it delivers the data packet to all nodes irrespective of the destination.
- **The network may be clogged with unwanted and duplicate data packets**. This may hamper delivery of other data packets.
- **One such measure is to have a hop counter** contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero.
- Ideally, the hop counter should be initialized to the length of the path from source to destination.
- If the sender does not know how long the path is, it can initialize the counter to the worst case, namely, the full diameter of the subnet.
- **An alternative technique is to keep track of which packets have been flooded**, to avoid sending them out a second time.
- In this method the source router put a sequence number in each packet it receives from its hosts.
- Each router then needs a list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded.
- A variation of flooding that is slightly more practical is **selective flooding**.

- **In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.**

Use of Flooding

- **In military applications**, where large numbers of routers may be blown at any instant, the tremendous robustness of flooding is highly desirable.
- **In distributed database applications**, it is sometimes necessary to update all the databases concurrently, in which case flooding can be useful.
- **In wireless networks**, all messages transmitted by a station can be received by all other stations within its radio range, which is, in fact, flooding, and some algorithms utilize this property.
- **In comparison of routing algorithms**, use as a metric to compare routing algorithms. Flooding always chooses the shortest path because it chooses every possible path in parallel.

The Network Layer in the Internet

5.7.1 The IP Version 4 Protocol

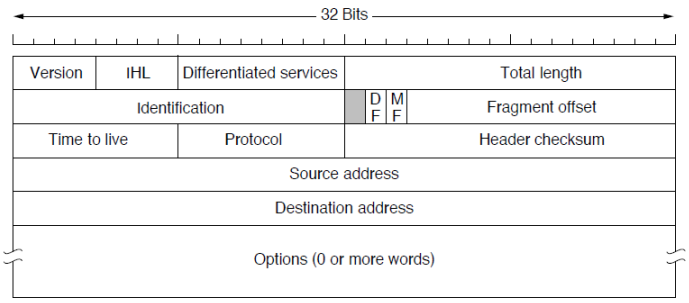


Figure 5-47. The IPv4 (Internet Protocol version 4) header.

- It is a connectionless and datagram protocol service.
- A field in the header, *IHL*, is provided to tell how long the header is, in 32-bit words.
- *Differentiated services- Type of service*
- The *Total length* includes everything in the datagram—both header and data.
- The *Identification* field is needed to allow the destination host to determine which packet a newly arrived fragment belongs to. All the fragments of a packet contain the same *Identification* value.
- *DF* stands for Don't Fragment. *MF* stands for More Fragments.
- The *Fragment offset* tells where in the current packet this fragment belongs.
- The *TTL (Time to live)* field is a counter used to limit packet lifetimes.
- The *Protocol* field tells it which transport process to give the packet to. TCP is one possibility, but so are UDP and some others.
- The *Source address* and *Destination address* indicate the IP address of the source and destination network interfaces.

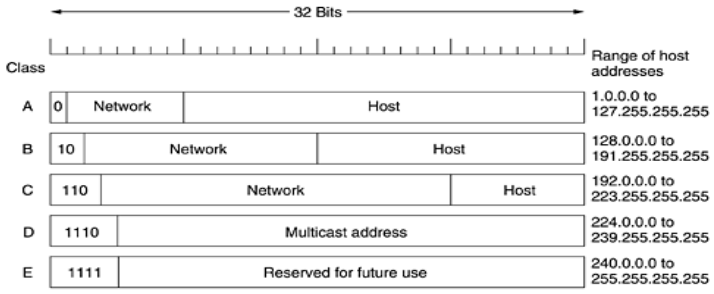
IP Address

- Every host and router on the Internet has an IP address, which encodes its network number and host number.
- The combination is unique: in principle, no two machines on the Internet have the same IP address.
- In the TCP/IP protocol, the unique identifier for a computer is called its IP address.
- There are two standards for IP addresses: **IP Version 4 (IPv4)** and **IP Version 6 (IPv6).**

Addresses - IPv4

- All IP addresses are 32 bits long and are used in the *Source address* and *Destination address* fields of IP packets.
- It is important to note that an IP address does not actually refer to a host. It really refers to a network interface, so if a host is on two networks, it must have two IP addresses.
- However, in practice, most hosts are on one network and thus have one IP address.
- The **32** bits of an IPv4 address are broken into **4 octets**, or 8 bit fields (0-255 value in decimal notation).

IP address formats



Classes of IPv4 address

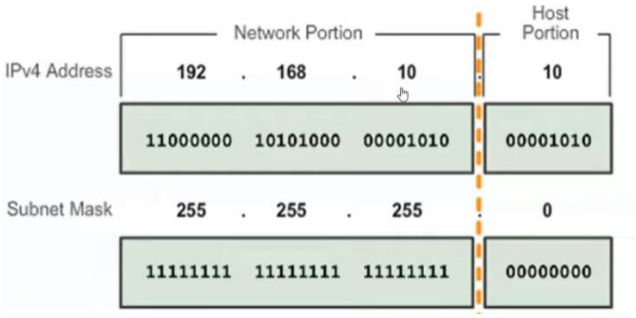
Address Class	1st Octet range in decimal	1st Octet bits (Blue Dots do not change)	Network (N) and Host (H) Portion	Default mask (Decimal)	Number of possible networks and hosts per network
A	0–127	00000000 - 01111111	N.H.H.H	255.0.0.0	128 Nets (2^7) 16,777,214 hosts (2^{24}–2)
B	128–191	10000000 - 10111111	N.N.H.H	255.255.0.0	16,384 Nets (2^{14}) 65,534 hosts (2^{16}–2)
C	192–223	11000000 - 11011111	N.N.N.H	255.255.255.0	2,09,150 Nets (2^{21}) 254 hosts (2^8–2)
D	224–239	11100000 - 11101111	NA (Multicast)	-	-
E	240–255	11110000 - 11111111	NA (Experimental)	-	-

nesoacademy.org

- For networks of different size, The first one (for large networks) to three (for small networks) octets can be used to identify the **network**, while the rest of the octets can be used to identify the **node** on the network.
- The class A formats allow for up to 128 networks with 16 million hosts each,
- The class B formats allow 16,384 networks with up to 64K hosts, and
- The class C formats allow 2 million networks (e.g., LANs) with up to 256 hosts each (although a few of these are special).

What is a subnet mask?

A subnet mask is a 32-bit number created by setting the host bits to all 0s and setting network bits to all 1s. In this way, the subnet mask is separated the IP address into the host address and network address. The broadcast address is always assigned to the "255" address, and a network address is always assigned to the "0" address. Since the subnet mask is reserved for a special purpose, it cannot be assigned to the host.



Addresses – Ipv6

- IPv6-use 16-byte addresses.
- IPv6 addresses are customarily written in eight blocks of four hexadecimal digits separated by colons, such as FEDC: BA98:7654:3210: FEDC: BA98:7654:3210.
- Leading zeros do not need to be written. A double colon, at most one of which may appear in any address, indicates multiple zero blocks.
- **For example,** FEDC:0000:0000:0000:00DC:0000:7076:0010
  could be written more compactly as FEDC::DC: 0:7076:10.

Q. 4

Change the following IP addresses from dotted-decimal notation to binary notation.

a. 114.34.2.8
b. 129.14.6.8
c. 208.34.54.12
d. 238.34.2.1

Solution:

a. 01110010 00100010 00000010 00001000
b. 10000001 00001110 00000110 00001000
c. 11010000 00100010 00110110 00001100
d. 11101110 00100010 00000010 00000001

Q. 5

Find the class of the following IP addresses.

a. 208.34.54.12
b. 238.34.2.1
c. 114.34.2.8
d. 129.14.6.8

Solution:

a. Class C
b. Class D
c. Class A
d. Class B

Q. 6 **Convert the IP address whose hexadecimal representation is C22F1582 to dotted decimal notation.**

Solution:

The address is 194.47.21.130