Chapter 6

The Network Layer

6.1 IP Addressing

- The Network layer is responsible for logical addressing.
- The IP protocol (of TCP/IP protocol suit) provides the logical address.
- Currently version 4 and version 6 is in use.
- IPv4 uses 32-bit addresses, while IPv6 uses 128-bit addresses.
- Every host and router on the Internet has an IP address.
- An IP address consists of two numbers:
  - The Network number, and
  - The Host number.
- No two machines on the Internet can have the same IP address.
- A single host may have multiple IP addresses.

6.1.1 Classful Addressing

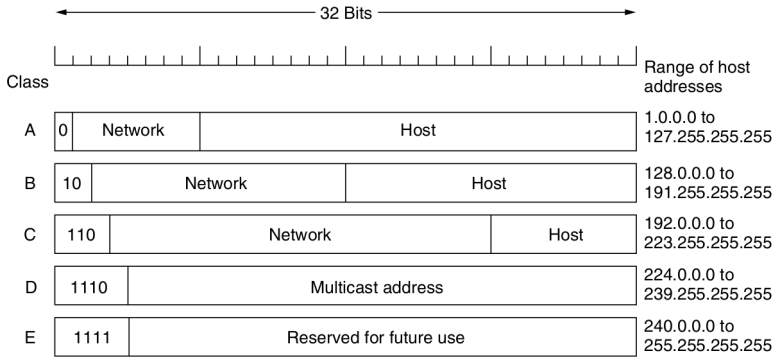

Figure 6.1: IPv4 Address Format & Range (Courtesy: Computer Networks by Tanenbaum, 5th Ed.)

- Consists of five classes. The classes and the IP address range of each class is shown in figure 6.1
- Class A supports 128 networks, and up to 16 million hosts in each network.
- Class B supports 16,384 networks, and up to 64,000 hosts per network.

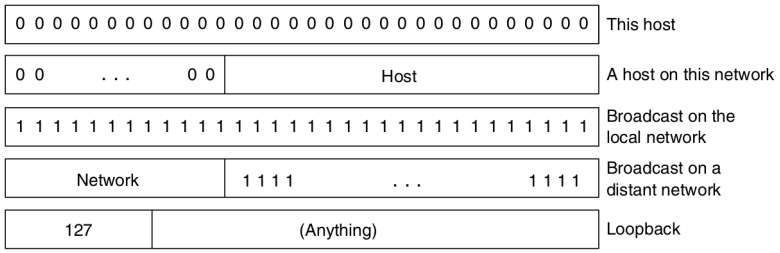

Figure 6.2: Special IP Addresses (Courtesy: Computer Networks by Tanenbaum, 5th Ed.)

10.0.0.0	– 10.255.255.255/8	(16,777,216 hosts)
172.16.0.0	– 172.31.255.255/12	(1,048,576 hosts)
192.168.0.0	– 192.168.255.255/16	(65,536 hosts)

Figure 6.3: Private IP Addresses (Courtesy: Computer Networks by Tanenbaum, 5th Ed.)

- Class C supports 2 million networks, and up to 254 hosts per network.
- To avoid conflicts, network numbers are managed by IANA[1], a department of ICANN[2].
- Network numbers are usually written in *Quad Dotted Decimal Notation*, Each of the 4-bytes are written in decimal (from 0 to 255), separated by a dot character.
- The lowest IP address is 0.0.0.0 (32 zeros in binary) and highest is 255.255.255.255 (32 ones in binary).
- Address with all 0s and all 1s have special meaning. It is shown in figure 6.2:
  - 0.0.0.0 : This network or this host.
  - 255.255.255.255 : Broadcast address.
- Loopback address: 127.x.x.x
- Private IP address ranges are shown in figure 6.3

6.1.2 Classless Inter-Domain Routing (CIDR)

IETF[3] introduced CIDR in 1993 to replace the classful addressing. The goal was to slow-down the growth of routing tables on routers across the internet, and to help slow-down the rapid exhaustion of IPv4 addresses.

- CIDR allocates address space on any address bit boundary, instead of on 8-bit segments.
- CIDR notation is a syntax of specifying IP address and their associated routing prefix. It appends to the address a slash character and the decimal number of leading bits of routing prefix (e.g. 192.168.0.1/16).

6.1.3 Subnets

- Some bits are taken away from the host number to create a subnet number (also called the subnet mask), so that number of distinct additional networks can be created from the available network address.
- Default Subnets:

[1]Internet Assigned Numbers Authority
[2]Internet Corporation for Assigned Names and Numbers
[3]Internet Engineering Task Force

  - For Class A: 255.0.0.0 (/8)
  - For Class B: 255.255.0.0 (/16)
  - For Class C: 255.255.255.0 (/24)

- For example:
  IP Address: 172.16.10.1
  Default Subnet: 255.255.0.0
  So,
  Network Number: 172.16
  Host Number: 10.1

Possible Subnet Numbers:

- $10000000_2 = 128_{10}$
- $11000000_2 = 192_{10}$
- $11100000_2 = 224_{10}$
- $11110000_2 = 240_{10}$
- $11111000_2 = 248_{10}$
- $11111100_2 = 252_{10}$
- $11111110_2 = 254_{10}$
- $11111111_2 = 255_{10}$

6.1.4 Supernets

The basic concept of supernet is similar to subnets. However, in this case, instead of taking away some bits from the host number to extend the number of available networks, some bits from the network number is taken away to extend the number of hosts. This is done mostly on class C addresses.

6.1.5 Subnetting Examples

1. IP Address: 172.16.0.1
   Subnet Mask: 255.255.240.0
   CIDR Notation: 172.16.0.1/20

   In this case:
   Subnet Mask: 20
   Subnet Bits: 4

```
172.16.0.1    : 10101100.00010000.00000000.00000001
255.255.240.0 : 11111111.11111111.11110000.00000000
-------------------------------------------------
Bitwise AND   : 10101100.00010000.00000000.00000000
In Decimal    :      172.      16.       0.       0
```

   So,
   The Subnet ID or the Network ID: 172.16.0.0/20

   To calculate the address range:
   The network number is up to the 1s present in the subnet mask – this part is fixed. So, the first octet value is 172, and the second octet value is 16. The 0s in the subnet mask represents host number. The fourth octet has all 0s in the subnet mask – so the value can range from 0 to 255. Now, we need to calculate the third octet values. Since, we have four 1s followed by four 0s in the

subnet mask in this octet, it can range from 00000000_2 to 00001111_2 (the first four bits can not change, since the subnet bits are 1 here, meaning network number), which is equal to 0 and 15 in decimal.

As such,
Address Range: 172.16.0.1 – 172.16.15.254
Broadcast Address: 172.16.15.255 (the host part has to be all 1s)

2. IP Address: 172.16.10.1
   Subnet Mask: 255.255.252.0
   CIDR Notation: 172.16.10.1/22

   In this case:
   Subnet Mask: 22
   Subnet Bits: 6

```
172.16.10.1   : 10101100.00010000.00001010.00000001
255.255.252.0 : 11111111.11111111.11111100.00000000
-------------------------------------------------
Bitwise AND   : 10101100.00010000.00001000.00000000
In Decimal    :      172.      16.       8.       0
```

   So, The Subnet ID or the Network ID: 172.16.8.0/22

   To calculate the address range:
   The third octet values can range from 00001000_2 to 00001011_2, since only the last two bits are 0 in the subnet mask of this octet.

   As such,
   Address Range: 172.16.8.1 – 172.16.11.254
   Broadcast Address: 172.16.11.255

3. IP Address: 172.16.100.1
   Subnet Mask: 255.255.240.0
   CIDR Notation: 172.16.100.1/20

   In this case:
   Subnet Mask: 20
   Subnet Bits: 4

```
172.16.100.1  : 10101100.00010000.01100100.00000001
255.255.240.0 : 11111111.11111111.11110000.00000000
-------------------------------------------------
Bitwise AND   : 10101100.00010000.01100000.00000000
In Decimal    :      172.      16.      96.       0
```

   So, The Subnet ID or Network ID: 172.16.96.0/20

   To calculate the address range:
   The third octet values can range from 01100000_2 to 01101111_2, since the last four bits are 0 in the subnet mask of this octet.

   As such,
   Address Range: 172.16.96.1 – 172.16.111.254
   Broadcast Address: 172.16.111.255

6.1.6 Supernetting Example

1. IP Address: 192.168.128.1
   Subnet Mask: 255.255.252.0
   CIDR Notation: 192.168.128.1/22

   In this case:
   Subnet Mask: 22
   Subnet Bits: 6

```
192.168.128.1 : 11000000.10101000.10000000.00000001
255.255.252.0 : 11111111.11111111.11111100.00000000
-----------------------------------------------------
Bitwise AND   : 11000000.10101000.10000000.00000000
In Decimal    :      192.     168.     128.       0
```

   So, The Subnet ID or Network ID: 192.168.128.0/22

   To calculate the address range:
   The third octet values can range from 10000000_2 to 10000011_2, since only the last two bits are 0 in the subnet mask of this octet.

   As such,
   Address Range: 192.168.128.1 – 192.168.131.254
   Broadcast Address: 192.168.131.255

6.2 The IP Protocol

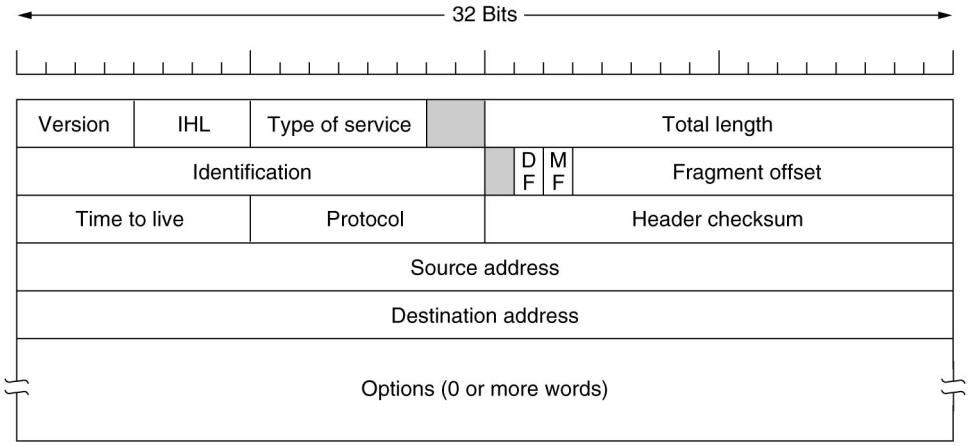

Figure 6.4: IPv4 Header (Original)

- IP datagram consists of a *Header* part and a *Text* (data) part.
- The header consists of 20-byte fixed part and a variable optional part.
- IP datagrams are transmitted in big-endian order: from left to right.
- The original IPv4 header is shown in figure 6.4, while the newer one is shown in figure 6.5
- The various fields of the header has the following meaning:

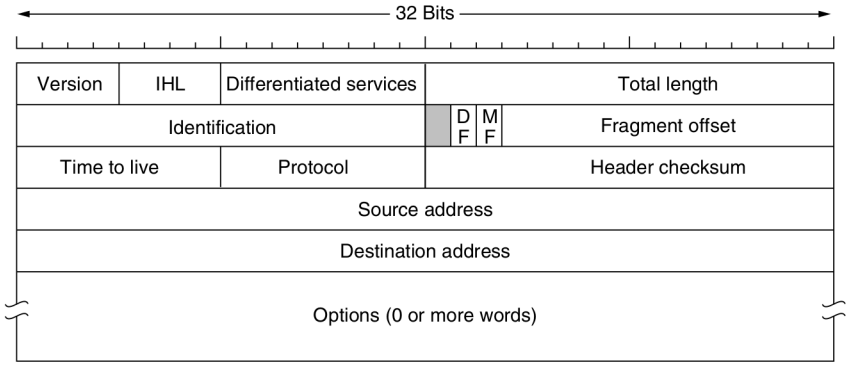

Figure 6.5: The IPv4 Header (Courtesy: Computer Networks by Tanenbaum, 5th Ed.)

- *Version:* IP version used, typically 4 or 6.
- *IHL:* Specifies the header length in 32-bit words. Minimum value if 5 (i.e. 20-bytes, when no option is set), and the maximum value is 15 (i.e. limits the header length to 60-bytes. So, options field can be of maximum 40-bytes long).
- *Type of Service:* Mostly ignored.
- *Total Length:* Size of (header + data). Maximum length is 65,535 bytes.
- *Identification:* For identification of fragments of a datagram (all fragments of a datagram contains the same ID value).
- *DF:* Don't fragment.
- *MF:* More fragments. All fragments of a datagram except the last one have this bit set.
- *Fragment Offset:* Where in the current datagram this fragment belongs to (with 13-bits: maximum 8192 fragments).
- *Time to Live:* Limits packet lifetime (maximum lifetime: 255 sec.). TTL is decremented hop-by-hop.
- *Protocol:* Name of transport process, i.e. the protocol used in the data portion of the IP datagram (eg. TCP, UDP, ICMP etc.).
- *Header Checksum:* Verifies the header (with 1's complement half-word addition).
- *Source & Destination Address:* Network and host number (Section 6.1 on page number 63).
- *Options:* Can contain options like Security, Strict Source Routing, Loose Source Routing, Record Routing, Timestamp.
- *Differentiated services:* Mostly used by emerging technologies requiring real-time data streaming. It was a replacement for the "Type of Service" field, and occupied 8-bits.
- *DSCP:* Differentiated Services Code Point. Same as the "Differentiated services" field, but reduced to 6-bits. The extra 2-bits are assigned to 'ECN'.
- *ECN:* Explicit Congestion Notification. Used to notify end-to-end network congestion without dropping packets. It is used only when both end points support it and agree to use it.
- *Flags:* The unused bit, DF bit, and MF bit (figure 6.4 & figure 6.5) are combined in the new format and called flags. The DF and MF bits have the same use as mentioned above. The first bit (unused bit in previous versions) must be zero.
- For DSCP, ECN, and Flags — refer to Wikipedia web site.

6.3 Routing Algorithms

Routing algorithms are broadly divided into two classes:

- Non-adaptive or Static Routing Algorithms.
- Adaptive or Dynamic Routing Algorithms.

6.3.1 Non-Adaptive or Static Routing Algorithms

a) *Shortest Path Routing:* In this technique, first we need to ascertain the topology of the network and assign some weight to the links connecting the routers. We may choose the weight to represent things like hop count, geographic distance, mean queuing & transmission delay (fastest path), bandwidth, average traffic, communication cost etc.

Once this is done, we can represent the network using a graph — the routers representing the nodes of the graph, and the lines or the links connecting the routers representing the edges of the graph.

Now, we can use a shortest path algorithm (such a Dijkstra's algorithm) to compute the shortest path between any two nodes of the graph. The result can be used as the routing table to forward the packets.

b) *Flooding:*

- Every incoming packet is sent out on every outgoing link, except the one it arrived on.
- To limit the duplicate packets, hop counter is used, which is decremented at each hop. The packet is discarded when the hop counter reaches zero.
- An alternative technique is to keep track of which packets have been flooded, to avoid sending them out a second time.
- A variation of flooding is called *selective flooding*. Instead of sending out packets on every lines, only those lines are chosen, that lies in the same direction of the destination.
- Flooding always chooses the shortest path.

6.3.2 Adaptive or Dynamic Routing Algorithms

a) *Distance Vector Routing:*

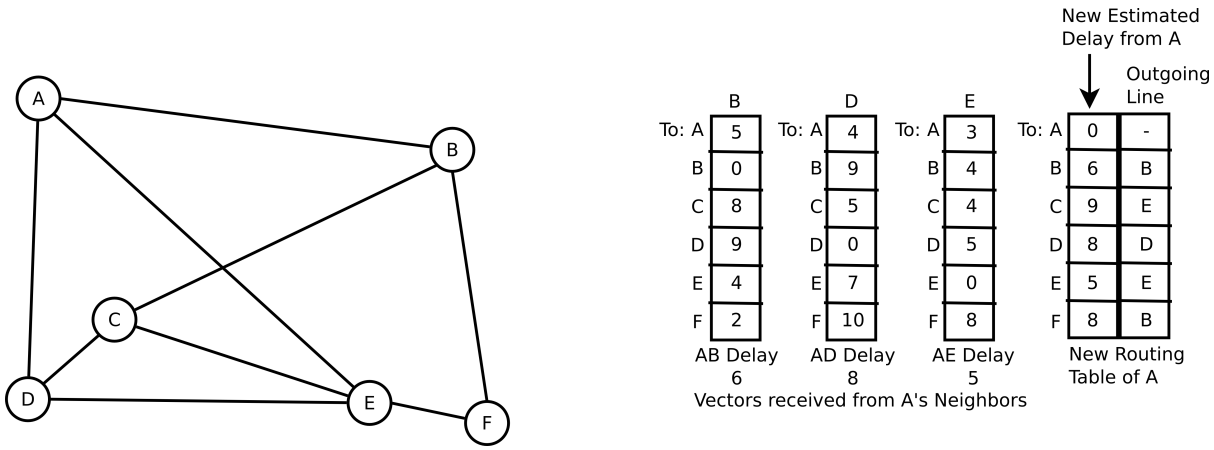

Figure 6.6: Distance Vector Routing (Calculating router A's routing table)

- This routing technique is also known as Distributed Bellman-Ford routing algorithm, Ford-Fulkerson routing algorithm, and RIP.
- Each router maintains a routing table indexed by, and containing one entry for each router in the subnet.
- This entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination.
- The 'distance' metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar.

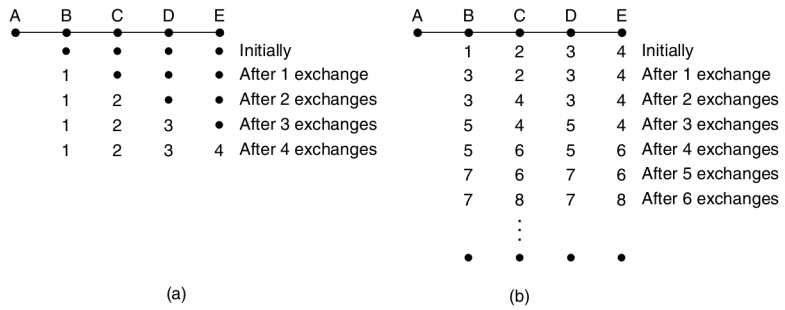

Figure 6.7: Count-to-Infinity Problem (Courtesy: Computer Networks by Tanenbaum, 5$^{\text{th}}$ Ed.)

- For example, say delay is used as a metric and that the router knows the delay to each of it's neighbors. Once every T msec, each router sends to each neighbor a list of it's estimated delays to each destination. It also receives a similar list from each of it's neighbors. Imagine that one of these tables has just come in from neighbor X, with X_i being X's estimate of how long it takes to get to router i. If the router knows that the delay to X is m msec, it also knows that it can reach router i via X in $X_i + m$ msec. By performing this calculation for each neighbor, a router can find out which estimate seems the best and use that estimate and the corresponding line in it's new routing table. Note that the old routing table is not used in the calculation.
- In the topology shown in figure 6.6, let's say A wants to reach C:
  - B says it can reach C in: 8 msec (B $\rightarrow$ C delay)
    A can reach B in: 6 msec (A $\rightarrow$ B delay)
    Total delay: 8 + 6 = 14 msec.
  - D says it can reach C in: 5 msec (D $\rightarrow$ C delay)
    A can reach D in: 8 msec (A $\rightarrow$ D delay)
    Total delay: 5 + 8 = 13 msec.
  - E says it can reach C in: 4 msec (E $\rightarrow$ C delay)
    A can reach E in: 5 msec (A $\rightarrow$ E delay)
    Total delay: 4 + 5 = 9 msec.

  So, A updates it's routing table with E as the preferred next router with 9 msec as delay, to reach destination C.

The Count-to-Infinity Problem

- If a router reports a better metric (e.g. distance) than the existing value for a particular route, it takes a short amount of time to switch to that route for the routers in the subnet. However, if a router or the path to a router fails, updation of the routing tables of routers may take a significant amount of time.
- The number of exchanges required (to update the routing tables of all routers of the subnet, in case a path or router goes down) depends on the numerical value used for infinity.
- So, infinity value should be set to the longest path value + 1.
- If the metric is time delay, there is no well defined upper bound. So, a high value is needed to prevent a path with long delay from being treated as down.
- This problem is shown in figure 6.7
- In the figure, (a) represents the good news, i.e. initially path to router A, or router A itself is down. All the other routers after some time has updated their routing table to indicate a infinity value for A, meaning there is no possible path to A. However, when the path to router A is restored, in a small number of exchanges (time), all the routers in the subnet updates their routing tables.

- The problem (or the bad news) is shown in (b). Initially there was a path to router A, but afterwards the link to router A or router A itself goes down. In this case, it takes a comparatively large amount of time to reflect the infinity value in all the router's routing tables in the subnet.

b) *Link State Routing:*

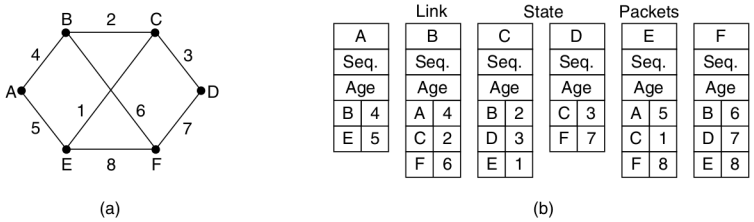

Figure 6.8: Link State Packets (Courtesy: Computer Networks by Tanenbaum, 5th Ed.)

In this technique, each router carries out the following five tasks:

i) Discover it's neighbors and learn their network addresses.

ii) Measure the delay or cost to each of it's neighbors.

iii) Construct a packet telling all it has just learned.

iv) Send this packet to all other routers.

v) Compute the shortest path to every other routers.

Each task is further explained below:

i) When a router boots, it's first job is to discover it's neighbors. It sends a special *HELLO* (ICMP) packet on each of it's point-to-point links. The router at the other end sends a reply telling who it is. All routers must have globally unique names.

ii) The link state routing algorithm requires each router to know, or at least have a reasonable estimate of the delay to each of it's neighbors. One way to calculate the delay is to use special *ECHO* (ICMP) packet, which the router at the other end replies immediately. The round-trip time is known in this case, and the delay can thus be calculated. The line load may also be taken into consideration.

iii) Once the information is collected, each router builds a packet containing all the data. The packet contains the identity of the sender, followed by a sequence number and age, and a list of neighbors. For each neighbor, the delay to that neighbor is given. This is shown in figure 6.8

The link state packets may be built periodically or when some significant event occurs, such as a line or neighbor going down or coming back up again or changing it's properties appreciably.

iv) One way to distribute the packets is through flooding. To minimize the flood packets, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (*source router*, *sequence*) pairs they have seen. When a new packet comes in, it is checked against the list of packets already received. If it is new, it is forwarded on all lines except the one it came in. Duplicate packets are discarded. A packet arriving with a lower sequence number is rejected. If a router crashes or the sequence number field is corrupted, the router will lose track of the correct sequence number. To solve this problem, the age field is used. The age field is decremented once per second. When the age field becomes zero, information from that router is discarded. The age field is also decremented by each router during the initial flooding process.

v) When the routers accumulate a full set of link state packets, it constructs the entire subnet graph. Every link is represented twice — once for each direction. The two values can be averaged or used separately. Now, Dijkstra's algorithm can be used locally to construct the shortest path. The result can be installed in the routing table and the routers can start operating.

Problems in Link State Routing

- If a router claims to have a line it does not have, or forgets a line it does have, the subnet graph will be incorrect.
- If a router fails to forward packet or corrupts them while forwarding them, problem will arise.
- If a router runs out of memory, or does the routing calculation wrong, problem will arise.

6.4 Congestion Control

6.4.1 Reasons for Congestion

- If a large number of packets arrive together in multiple lines of a router, and they all need the same outgoing line, a queue will build up. If there is insufficient memory to hold all of them, packets will be lost.
- Slow processors (in the router) can also cause congestion.
- Low bandwidth lines can also cause congestion.

6.4.2 Congestion Control Techniques

a) *The Warning Bit:* The router upon detection of congestion, can set a warning bit in the acknowledgment sent to the source. The source can then slow down the traffic. The router can keep setting the warning bit in the acknowledgments, until the congestion eases up.

b) *Choke Packets:* The router generates a special choke packet and sends to the source, mentioning the destination. The source reduces the traffic to the specified destination by some percent. More choke packets may be necessary to control the congestion.

c) *Hop-by-Hop Choke Packets:* By the time choke packets reaches the source, a large amount of data may have already been sent to the destination (if the bandwidth is large).

In this method, the intermediate routers upon receipt of the choke packets, also slow-down the traffic.

d) *Load Shedding:* In this technique, when congestion occurs, the routers simply starts dropping packets. However, some policy is required so that important packets are not dropped first. For this, senders must add a priority class to it's packets.

e) *Random Early Detect (RED):* This method starts dropping packets before the congestion problem becomes critical. To determine when to start discarding packets, routers maintain a running average of their queue lengths. When the average queue length on some lines exceeds a threshold, the line is said to be congested and action is taken. The routers picks up random packets to drop from the queue, since the router may not be able to tell which source is causing most of the problem. To inform the source, the router may send a chock packet, which will put more load on the congested network. Another approach is to not send any ACK. TCP reacts to unacknowledged packets by slowing down the traffic in a wired network. However, this technique can not be used in case of wireless networks.

6.5 Quality of Service (QoS)

A stream of packets from a source to the destination is called a *flow.*

- *For Connection-Oriented Service*: All the packets belonging to a flow follow the same route.
- *For Connectionless Service*: They may follow different routes.

6.5.1 Requirements of QoS

The needs of each flow can be characterized by *four* primary parameters:

i) Reliability

ii) Delay

iii) Jitter

iv) Bandwidth

Together, these determine the QoS the flow requires. Different types of applications may have different emphasis on the above parameters. For example, e-mail, file transfer, web access, remote login etc. requires high reliability, whereas audio/video conferencing is less strict in reliability but requires as less delay as possible. Similarly, e-mail, file transfer etc. can tolerate some amount of delay but requires good amount of bandwidth. In case of applications like video playback, jitter is an important parameter to consider.

6.5.2 Techniques of Achieving Good QoS

i) *Over-provisioning*:

- Provide more of the required critical resources, like provide high router capacity, large buffer space, and high bandwidth etc.
- However, it is expensive.

ii) *Buffering*:

- Buffering does not affect the reliability or bandwidth.
- It also increases the delay.
- But it smooths out the jitter.

iii) *Traffic Shaping*:

a) *The Leaky Bucket Algorithm*:

- It is modelled after a bucket which can hold some amount of water. The bucket has a small hole in the bottom. Water is coming in at uneven rate.
- No matter the rate at which water enters the bucket, the outflow (through the hole) is at a constant rate (provided there is water in the bucket, otherwise the outflow is zero).
- If more and more water enters the bucket, and the bucket becomes full, further incoming water will be lost.
- This idea can be used to handle incoming packets, arriving at an uneven rate, but for which we need a constant outgoing rate.
- In computer networks, this can be easily implemented with a finite *queue*.
- We can use the computer clock to regulate the rate of the outgoing packets.
- At every clock tick, one packet can be transmitted (unless the queue is empty), or some bytes can be transmitted (if the packets have variable sizes).

b) *The Token Bucket Algorithm*:

- The leaky bucket algorithm enforces a rigid output pattern at the average rate (i.e. no bursts are allowed).
- The token bucket algorithm is used to somewhat speed up the output, when large bursts arrive.
- In this algorithm, the bucket holds *tokens*, generated by a clock at the rate of one token every ΔT seconds.
- For a packet to be transmitted, it must capture and destroy one token.

- The leaky bucket algorithm does not allow bursts, but the token bucket algorithm allows bursts upto the number of tokens present in the bucket.
- The leaky bucket algorithm discards packets when the bucket fills up, but the token bucket algorithm discards token (i.e. transmission capacity) when the bucket fills up, but never discards packets.
- For the implementation purpose, a *variable* is used to count tokens. The counter is incremented by one every ΔT seconds and decremented by one whenever a packet (or some bytes) is sent.
- When the counter hits zero, no packets may be sent.