UNIT – 4

The Network Layer Design Issues

Network layer is majorly focused on getting packets from the source to the destination, routing error handling and congestion control.

Various functions in transport layer.

- **Addressing:**

  Maintains the address at the frame header of both source and destination and performs addressing to detect various devices in network.

  - **Packeting:**

    This is performed by Internet Protocol. The network layer converts the packets from its upper layer.

  - **Routing:**

    It is the most important functionality. The network layer chooses the most relevant and best path for the data transmission from source to destination.

  - **Inter-networking:**

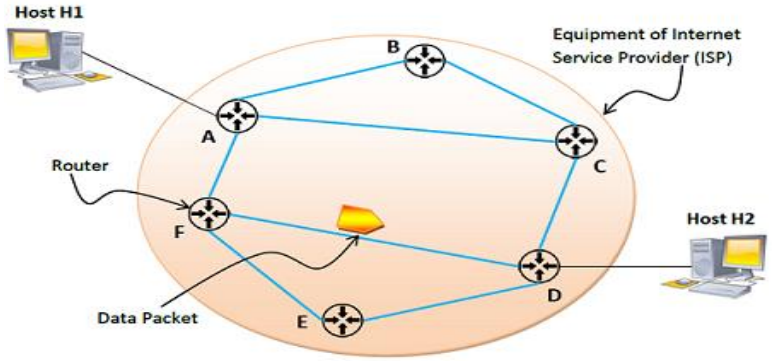
    It works to deliver a logical connection across multiple devices.

Network layer design issues:

The network layer comes with some design issues they are described as follows:

1. Store and Forward packet switching:

The host sends the packet to the nearest router. This packet is stored there until it has fully arrived once the link is fully processed by verifying the checksum then it is forwarded to the next router till it reaches the destination. This mechanism is called "Store and Forward packet switching."



In the above diagram, we can see that the Internet Service Provider (ISP) has six routers (A to F) connected by transmission lines shown in blue lines. There are two hosts, host H1 is connected to router A, while host H2 is connected to router D. Suppose that H1 wants to send a data packet to H2. H1 sends the packet to router A. The packet is stored in router A until it has arrived fully. Router A verifies the checksum using CRC (cyclic redundancy check) code. If there is a CRC error, the packet is discarded, otherwise it is transmitted to the next hop, here router F. The same process is followed by router F which then transmits the packet to router D. Finally router D delivers the packet to host H2.

Advantages and Disadvantages

Store and forward packet switching ensures high quality data packet transmission. Since erroneous packets are discarded at each router, bad packets or invalid packets in the network are mostly eliminated.

However, error − free packet transmission is achieved by compromising on the overall speed of transmission. Switch latency (Switch latency is the amount of time that an frame spends traversing an router i.e. waiting time) is introduced due to waiting for entire packet to arrive as well as computation of CRC. Though the latency at each router may seem small enough, the cumulative latency at all routers make it inappropriate for time − critical online applications.

2. Services provided to Transport Layer:

Through the network/transport layer interface, the network layer transfers it's services to the transport layer. These services are described below.

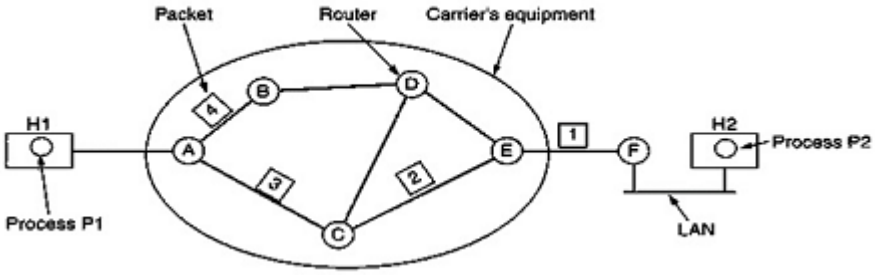
Connectionless: The routing and insertion of packets into subnet is done individually. No added setup is required.

Connection-Oriented: Subnet must offer reliable service and all the packets must be transmitted over a single route.

3. Implementation of Connectionless Service:

When connectionless service is offered, packets are frequently called Datagrams (just like telegrams) because individual packets are injected to the subnet and are routed individually.

No advance setup is required. Subnets are called Datagram subnets. When Connection oriented service is provided, then before any packet is sent a path from source router to destination router is established. This connection is called Virtual Circuit and the subnet is called Virtual Circuit subnet. The implementation of connectionless service is diagrammatically represented as follows



Example for Datagram Network

Let us discuss how datagram network works in stepwise manner −

Step 1: Suppose there is a process P1 on host H1 and is having a message to deliver to P2 on host H2. P1 hands the message to the transport layer along with instructions to be delivered to P2 on H2.

Step 2: Transport Layer code is running on H1 and within the operating system. It prepends a transport header to the message and the end result is given to the network layer.

Step 3: Let us assume for this example a packet which is four times heavier than the maximum size of the packet, then the packet is broken to four different packets and each of the packet is sent to the router A using point to point protocol and from this point career takes over.

Step 4: Each router will have an internal table saying where packets to be sent. Every table entry is a pair consisting of a destination and outgoing line to use for that destination. Only directly connected lines can be used.

Step 5: For example A has only two outgoing lines to B and C, therefore every incoming packet must be sent to one of these routers, even if the ultimate destination will be some other router.

Step 6: As the packets arrived at A, packet 1,2,3 and 4 were stored in brief. Then every packet is moved to C as per A's table. Packet 1 is forwarded to E and then moved to F. When packet 1 is moved to F, then it will be encapsulated in a data link layer and sent to H2 over to LAN. Packet 2 and 3 will also follow the same route.

Step 7: When packet 4 reaches A, then it was sent to router B, even if the destination was F. For some purpose A decided to send packet 4 through a different route. It was because of the traffic jam

in ACE path and the routing table was updated. Routing Algorithm decides routes, makes routing decisions and manages routing tables.

4. Implementation of Connection Oriented service:

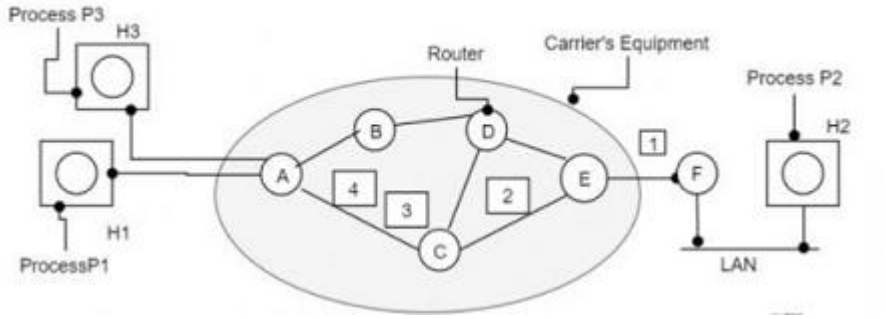
To use a connection-oriented service, first we establishes a connection, use it and then release it. In connection-oriented services, the data packets are delivered to the receiver in the same order in which they have been sent by the sender.

It can be done in either two ways :

Circuit Switched Connection: A dedicated physical path or a circuit is established between the communicating nodes and then data stream is transferred.

Virtual Circuit Switched Connection: The data stream is transferred over a packet switched network, in such a way that it seems to the user that there is a dedicated path from the sender to the receiver. A virtual path is established here. While, other connections may also be using the same path.

Example



Consider the scenario as mentioned in the above figure.

Step 1: Host H1 has established connection 1 with host H2, which is remembered as the first entry in every routing table.

Step 2: The first line of A's infers when packet is having connection identifier 1 is coming from host H1 and has to be sent to router W and given connection identifier as 1.

Step 3: Similarly, the first entry at W routes the packet to Y, also with connection identifier 1.

Step 4: If H3 also wants to establish a connection to H2 then it chooses connection identifier 1 and tells the subnet to establish the virtual circuit. This will be appearing in the second row in the table.

Step 5: Note that we have a conflict here because although we can easily distinguish connection 1 packets from H1 from connection 1 packet from H3, W cannot do this.

Step 6: For this reason, we assign a different connection identifier to the outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets. In some contexts, this is called label switching.

Comparison of Virtual Circuit and Datagram Networks

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing Algorithm

The main function of NL (Network Layer) is routing packets from the source machine to the destination machine. The routing algorithm is that part of the NL software responsible for deciding which output line an incoming packet should be transmitted on.

A routing algorithm is a procedure that lays down the route or path to transfer data packets from source to the destination. They help in directing Internet traffic efficiently. After a data packet leaves

its source, it can choose among the many different paths to reach its destination. Routing algorithm mathematically computes the best path, i.e. “least – cost path” that the packet can be routed through.

The Optimality Principle

Introduction:

A general statement is made about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle (Bellman,1975).

Statement of the optimality principle:

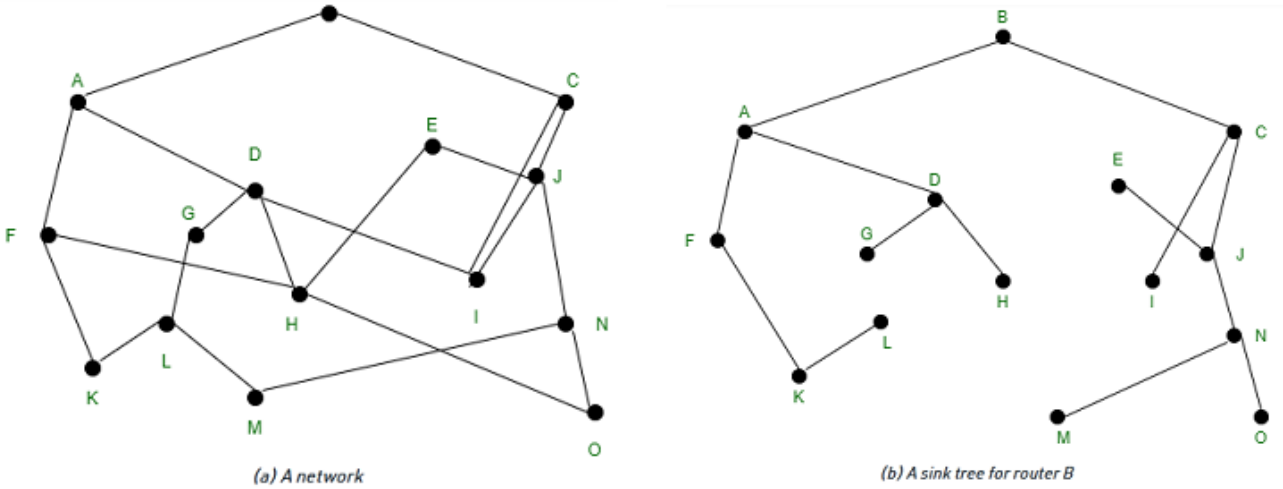
It states that if the router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route. Call the route from I to J r1 and the rest of the route r2. it could be concatenated with r1 to improve the route from I to K, contradicting our statement that r1r2 is optimal only if a route better than r2 existed from J to K.

Sink Tree for routers:

We can see that the set of optimal routes from all sources to a given destination from a tree rooted at the destination as a directed consequence of the optimality principle. This tree is called a sink tree and is illustrated in fig(1).

Description of figure:

In the given figure the distance metric is the number of hops. Therefore, the goal of all routing algorithms is to discover and use the sink trees for all routers.



(a) A network

(b) A sink tree for router B

The sink tree is not unique also other trees with the same path lengths may exist. If we allow all of the possible paths to be chosen, the tree becomes a more general structure called a DAG (Directed Acyclic Graph). DAGs have no loops. We will use sink trees as a convenient shorthand for both cases. we will take technical assumption for both cases that the paths do not interfere with each other so, for example, a traffic jam on one path will not cause another path to divert.

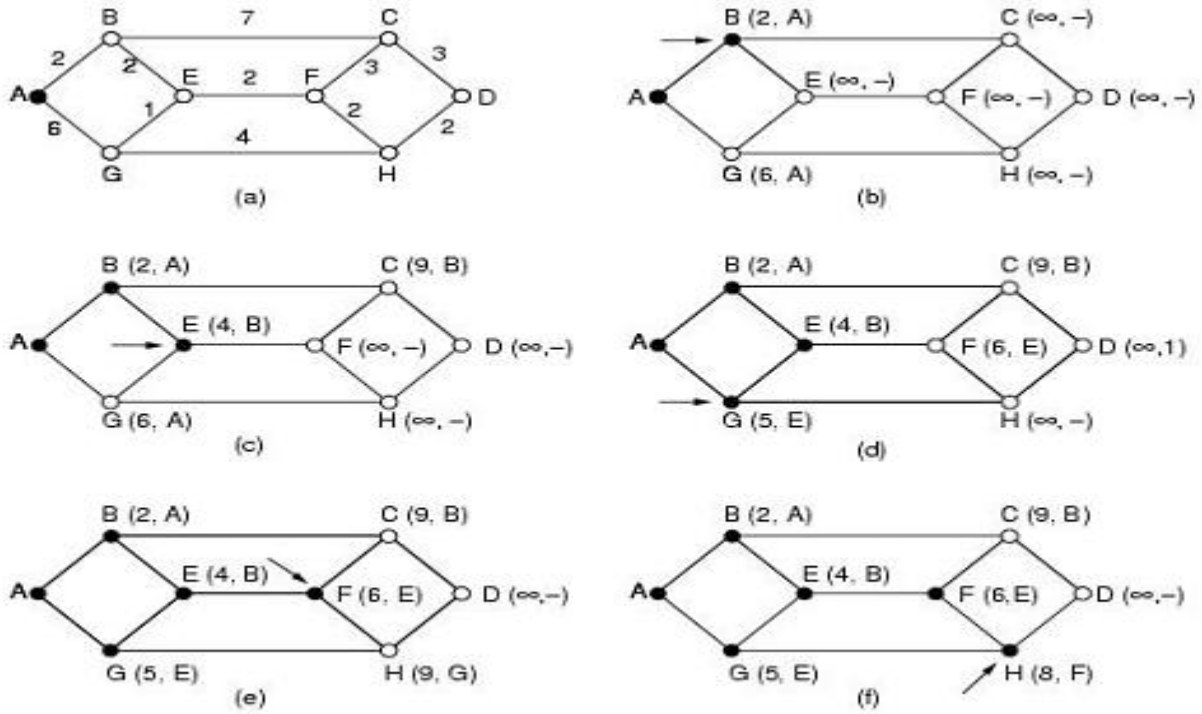
Conclusion:

The sink tree does not contain any loops, so each packet will be delivered within a finite and bounded number of hops. In practice, life is not quite easy. Links and routers can go on and come back up during operation, so different routers may have different ideas about the current topology. Also, we have found the issue of whether each router has to individually acquire the information on which to base its sink tree computation or whether this information is collected by some other means. Sink tree and the optimality principle provide a benchmark against which other routing algorithms can be measured.

Shortest Path Algorithm

Let us begin our study of routing algorithms with a simple technique for computing optimal paths given a complete picture of the network. These paths are the ones that we want a distributed routing algorithm to find, even though not all routers may know all of the details of the network. The idea is to build a graph of the network, with each node of the graph representing a router and each edge of the graph representing a communication line, or link. To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.

The concept of a shortest path deserves some explanation. One way of measuring path length is the number of hops. Using this metric, the paths ABC and ABE in Fig. are equally long. Another metric is the geographic distance in kilometers, in which case ABC is clearly much longer than ABE (assuming the figure is drawn to scale).



The first six steps used in computing the shortest path from A to D. The arrows indicate the working node.

However, many other metrics besides hops and physical distance are also possible. For example, each edge could be labeled with the mean delay of a standard test packet, as measured by hourly runs. With this graph labeling, the shortest path is the fastest path rather than the path with the fewest edges or kilometers. In the general case, the labels on the edges could be computed as a function of the distance, bandwidth, average traffic, communication cost, 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) and finds the shortest paths between a source and all destinations in the network. Each node is labeled (in parentheses) with its distance from the source node along the best known path. The distances must be non-negative, as they will be if they are based on real quantities like bandwidth and delay. 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. To illustrate how the labeling algorithm works, look at the weighted, undirected graph of Fig., where the weights represent, for example, distance. We want to find the shortest path from A to D. We start out by marking node A as permanent, indicated by a filled-in circle. Then we examine, in turn, each of the nodes adjacent to A (the working node), relabeling each one with the distance to A. Whenever a node is relabeled, we also label it with the node from which the probe was made so that we can reconstruct the final path later. If the network had more than one shortest path from A to D and we wanted to find all of them, we would need to remember all of the probe nodes that could reach a node with the same distance. Having examined each of the nodes adjacent to A, we examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label permanent, as shown in Fig. (b). this one becomes the new working node. We now start at B and examine all nodes adjacent to it. If the sum of the label on B and the distance from B to the node being considered is less than the label on that node, we have a shorter path, so the node is relabeled. After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively labeled node with the smallest value. This node is made permanent and becomes the working node for the next round. Figure shows the first six steps of the algorithm. To see why the algorithm works, look at Fig. (c). At this point we have just made E permanent. Suppose that there were a shorter path than ABE, say AXYZE (for some X and Y). There are two possibilities: either node Z has already been made permanent, or it has not been. If it has, then E has already been probed (on the round following the one when Z was made permanent), so the AXYZE path has not escaped our attention and thus cannot be a shorter path. Now consider the case where Z is still tentatively labeled. If the label at Z is greater than or equal to that at E, then AXYZE cannot be a shorter path than ABE. If the label is less than that of E, then Z and not E will become permanent first, allowing E to be probed from Z. This algorithm is given in Fig. The global variables n and dist describe the graph and are initialized before shortest path is called. The only difference between the program and the algorithm described above is that in Fig., we compute the shortest path starting at the terminal node, t, rather than at the source node, s. Since the shortest paths from t to s in an undirected graph are the same as the shortest paths from s to t, it does not matter at which end we begin. The reason for searching backward is that each node is labeled with its predecessor rather than its successor. When the final path is copied into the output

variable, path, the path is thus reversed. The two reversal effects cancel, and the answer is produced in the correct order.

Flooding

When a routing algorithm is implemented, each router must make decisions based on local knowledge, not the complete picture of the network. A simple local technique is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on. Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process. One such measure is to have a hop counter contained in the header of each packet that 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 network. Flooding with a hop count can produce an exponential number of duplicate packets as the hop count grows and routers duplicate packets they have seen before. A better technique for damming the flood is to have routers keep track of which packets have been flooded, to avoid sending them out a second time. One way to achieve this goal is to have the source router put a sequence number in each packet it receives from its hosts. Each router then needs a list persource 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.

```
#define MAX NODES 1024 /* maximum number of nodes */
#define INFINITY 1000000000 /* a number larger than every maximum path */
int n, dist[MAX NODES][MAX NODES]; /* dist[i][j] is the distance from i to j */
void shortest path(int s, int t, int path[])
{
struct state { /* the path being worked on */
int predecessor; /* previous node */
int length; /* length from source to this node */
enum {permanent, tentative} label; /* label state */
}
state[MAX NODES];
```

```
int i, k, min;
struct state *p;
for (p = &state[0]; p < &state[n]; p++) { /* initialize state */
p->predecessor = −1;
p->length = INFINITY;
p->label = tentative;
}
state[t].length = 0; state[t].label = permanent;
k = t; /* k is the initial working node */
do { /* Is there a better path from k? */
for (i = 0; i < n; i++) /* this graph has n nodes */
if (dist[k][i] != 0 && state[i].label == tentative) {
if (state[k].length + dist[k][i] < state[i].length) {
state[i].predecessor = k;
state[i].length = state[k].length + dist[k][i];
}
}
/* Find the tentatively labeled node with the smallest label. */
k = 0; min = INFINITY;
for (i = 0; i < n; i++)
if (state[i].label == tentative && state[i].length < min) {
min = state[i].length; k = i;
}
state[k].label = permanent;
} while (k != s);
/* Copy the path into the output array. */
i = 0; k = s;
do {path[i++] = k; k = state[k].predecessor; } while (k >= 0);
}
```

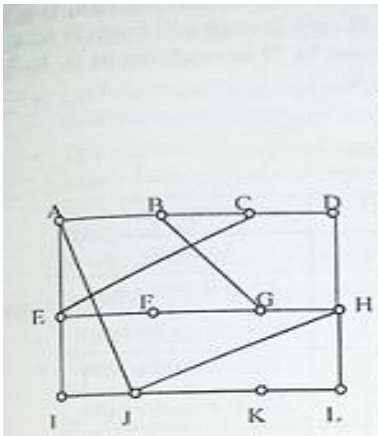
Distance Vector Routing

Computer networks generally use dynamic routing algorithms that are more complex than flooding, but more efficient because they find shortest paths for the current topology. Two dynamic algorithms in particular, distance vector routing and link state routing, are the most popular. In this section, we will look at the former algorithm. In the following section, we will study the latter algorithm. A distance vector routing algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there. These tables are updated by exchanging information with the neighbors. Eventually, every router knows the best link to reach each destination. The distance vector routing algorithm is sometimes called by other names, most commonly the distributed Bellman-Ford routing algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962). It was the original ARPANET routing algorithm and was also used in the Internet under the name RIP. In distance vector routing, each router maintains a routing table indexed by, and containing one entry for each router in the network. This entry has two parts: the preferred outgoing line to use for that destination and an estimate of the distance to that destination. The distance might be measured as the number of hops or using another metric, as we discussed for computing shortest paths. The router is assumed to know the ‘‘distance’’ to each of its neighbors. If the metric is hops, the distance is just one hop. If the metric is propagation delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can. As an example, assume that delay is used as a metric and that the router knows the delay to each of its neighbors. Once every T m sec, each router sends to each neighbor a list of its estimated delays to each destination. It also receives a similar list from each neighbor. Imagine that one of these tables has just come in from neighbor X, with Xi 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 m sec, it also knows that it can reach router i via X in Xi 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 link in its new routing table. Note that the old routing table is not used in the calculation. This updating process is illustrated in Fig. 5-9. Part (a) shows a network. The first four columns of part (b) show the delay vectors received from the neighbors of router J. A claims to have a 12-msec delay to B, a 25-msec delay to C, a 40-msec delay to D, etc.

Suppose that J has measured or estimated its delay to its neighbors, A, I, H, and K, as 8, 10, 12, and 6 m sec, respectively.

(a) A network. (b) Input from A, I, H, K, and the new routing table for J.

Consider how J computes its new route to router G. It knows that it can get to A in 8 m sec, and furthermore A claims to be able to get to G in 18 m sec, so J knows it can count on a delay of 26 m sec to G if it forwards packets bound for G to A. Similarly, it computes the delay to G via I, H, and K as 41 (31 + 10), 18 (6 + 12), and 37 (31 + 6) m sec, respectively. The best of these values is 18, so it makes an entry in its routing table that the delay to G is 18 m sec and that the route to use is via H. The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.



(a) Subnet

New estimated delay from J

TO	A	I	H	K		LINE
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K

(b) Input from routers A, I, H and K and new routing table for J.

JA Delay is 8	JI Delay is 10	JH Delay is 12	JK Delay is 6	New routing table for J

Vector received from J's four neighbors

Link State Routing

Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.

The three keys to understand the Link State Routing algorithm:

- **Knowledge about the neighborhood:** Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other routers.
- **Flooding:** Each router sends the information to every other router on the internetwork except its neighbors. This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbors. Finally, each and every router receives a copy of the same information.
- **Information sharing:** A router sends the information to every other router only when the change occurs in the information.

Link State Routing has two phases:

- **Reliable Flooding**
  - Initial state: Each node knows the cost of its neighbors.
  - Final state: Each node knows the entire graph.
- **Route Calculation**

Each node uses Dijkstra's algorithm on the graph to calculate the optimal routes to all nodes.

- The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network.
- The Dijkstra's algorithm is an iterative, and it has the property that after kth iteration of the algorithm, the least cost paths are well known for k destination nodes.

Let's describe some notations:

- **c(i , j):** Link cost from node i to node j. If i and j nodes are not directly linked, then c(i , j) = ∞.
- **D(v):** It defines the cost of the path from source code to destination v that has the least cost currently.
- **P(v):** It defines the previous node (neighbor of v) along with current least cost path from source to v.
- **N:** It is the total number of nodes available in the network.

Algorithm

Initialization

N = {A} // A is a root node.

for all nodes v

if v adjacent to A

then D(v) = c(A,v)

else D(v) = infinity

loop

find w not in N such that D(w) is a minimum.

Add w to N

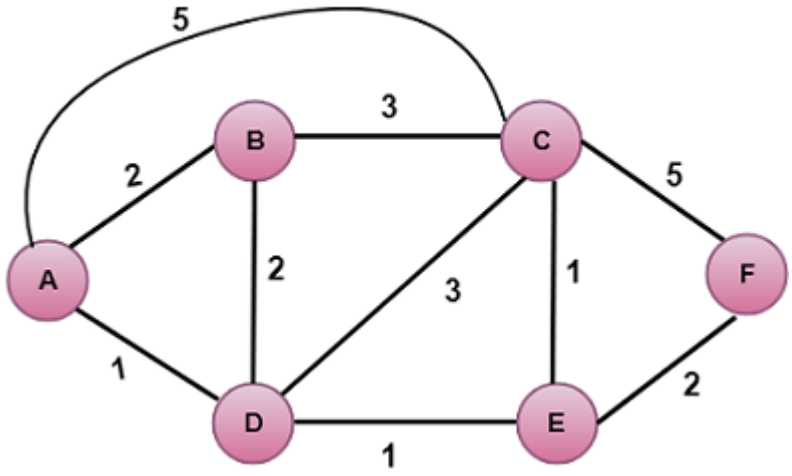
Update D(v) for all v adjacent to w and not in N:

D(v) = min(D(v) , D(w) + c(w,v))

Until all nodes in N

In the above algorithm, an initialization step is followed by the loop. The number of times the loop is executed is equal to the total number of nodes available in the network.

Example:



In the above figure, source vertex is A.

Step 1:

The first step is an initialization step. The currently known least cost path from A to its directly attached neighbors, B, C, D are 2,5,1 respectively. The cost from A to B is set to 2, from A to D is set to 1 and from A to C is set to 5. The cost from A to E and F are set to infinity as they are not directly linked to A.

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞

Step 2:

In the above table, we observe that vertex D contains the least cost path in step 1. Therefore, it is added in N. Now, we need to determine a least-cost path through D vertex.

a) Calculating shortest path from A to B

v = B, w = D

D(B) = min(D(B) , D(D) + c(D,B))

= min(2, 1+2)>

= min(2, 3)

The minimum value is 2. Therefore, the currently shortest path from A to B is 2.

b) Calculating shortest path from A to C

v = C, w = D

D(B) = min(D(C) , D(D) + c(D,C))

= min(5, 1+3)

= min(5, 4)

The minimum value is 4. Therefore, the currently shortest path from A to C is 4.</p>

c) Calculating shortest path from A to E

v = E, w = D

D(B) = min(D(E) , D(D) + c(D,E))

= min(∞, 1+1)

= min(∞, 2)

The minimum value is 2. Therefore, the currently shortest path from A to E is 2.

Note: The vertex D has no direct link to vertex E. Therefore, the value of D(F) is infinity.

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞

Step 3:

In the above table, we observe that both E and B have the least cost path in step 2. Let's consider the E vertex. Now, we determine the least cost path of remaining vertices through E.

a) Calculating the shortest path from A to B.

v = B, w = E

D(B) = min(D(B) , D(E) + c(E,B))

= min(2 , 2+ ∞)

= min(2, ∞)

The minimum value is 2. Therefore, the currently shortest path from A to B is 2.

b) Calculating the shortest path from A to C.

v = C, w = E

D(B) = min(D(C) , D(E) + c(E,C))

= min(4 , 2+1)

= min(4,3)

The minimum value is 3. Therefore, the currently shortest path from A to C is 3.

c) Calculating the shortest path from A to F.

v = F, w = E

D(B) = min(D(F) , D(E) + c(E,F))

= min(∞ , 2+2)

= min(∞ ,4)

The minimum value is 4. Therefore, the currently shortest path from A to F is 4.

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞
3	ADE	2,A	3,E			4,E

Step 4:

In the above table, we observe that B vertex has the least cost path in step 3. Therefore, it is added in N. Now, we determine the least cost path of remaining vertices through B.

a) Calculating the shortest path from A to C.

v = C, w = B

D(B) = min(D(C) , D(B) + c(B,C))

= min(3 , 2+3)

= min(3,5)

The minimum value is 3. Therefore, the currently shortest path from A to C is 3.

b) Calculating the shortest path from A to F.

v = F, w = B

D(B) = min(D(F) , D(B) + c(B,F))

= min(4, ∞)

= min(4, ∞)

The minimum value is 4. Therefore, the currently shortest path from A to F is 4.

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞
3	ADE	2,A	3,E			4,E
4	ADEB		3,E			4,E

Step 5:

In the above table, we observe that C vertex has the least cost path in step 4. Therefore, it is added in N. Now, we determine the least cost path of remaining vertices through C.

a) Calculating the shortest path from A to F.

v = F, w = C

D(B) = min(D(F) , D(C) + c(C,F))

= min(4, 3+5)

= min(4,8)

The minimum value is 4. Therefore, the currently shortest path from A to F is 4.

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞
3	ADE	2,A	3,E			4,E
4	ADEB		3,E			4,E
5	ADEBC					4,E

Final Table:

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞
3	ADE	2,A	3,E			4,E
4	ADEB		3,E			4,E
5	ADEBC					4,E
6	ADEBCF					

Disadvantage:

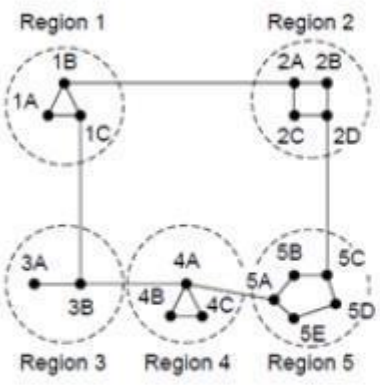
Heavy traffic is created in Line state routing due to Flooding. Flooding can cause an infinite looping, this problem can be solved by using Time-to-leave field

Hierarchical Routing

As networks grow in size, the router routing tables grow proportionally. Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them. At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network. When hierarchical routing is used, the routers are divided into what we will call regions. Each router knows all the details about how to route packets to destinations within its own region but knows nothing about the internal structure of other regions. When different networks are interconnected, it is natural to regard each one as a separate region to free the routers in one network from having to know the topological structure of the other ones. For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on, until we run out of names for aggregations. As an example of a multilevel hierarchy, consider how a packet might be routed from Berkeley,

California, to Malindi, Kenya. The Berkeley router would know the detailed topology within California but would send all out-of-state traffic to the Los Angeles router. The Los Angeles router would be able to route traffic directly to other domestic routers but would send all foreign traffic to New York. The New York router would be programmed to direct all traffic to the router in the destination country responsible for handling foreign traffic, say, in Nairobi. Finally, the packet would work its way down the tree in Kenya until it got to Malindi. Figure gives a quantitative example of routing in a two-level hierarchy with five regions. The full routing table for router 1A has 17 entries, as shown in Fig. (b). When routing is done hierarchically, as in Fig. 5-14(c), there are entries for all the local routers, as before, but all other regions are condensed into a single router, so all traffic for region 2 goes via the 1B-2A line, but the rest of the remote traffic goes via the 1C-3B line. Hierarchical routing has reduced the table from 17 to 7 entries. As the ratio of the number of regions to the number of routers per region grows, the savings in table space increase. Unfortunately, these gains in space are not free. There is a penalty to be paid: increased path length. For example, the best route from 1A to 5C is via region 2,but with hierarchical routing all traffic to region 5 goes via region 3, because that is better for most destinations in region 5.When a single network becomes very large, an interesting question is ‘‘how many levels should the hierarchy have?’’ For example, consider a network with 720 routers. If there is no hierarchy, each router needs 720 routing table entries. If the network is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of 53 entries. If a three-level hierarchy is chosen, with 8 clusters each containing 9 regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own cluster, and 7 entries for distant clusters, for a total of 25 entries.Kamoun and Kleinrock (1979) discovered that the optimal number of levels for an N router network is ln N, requiring a total of e ln N entries per router. They have also shown that the increase in effective mean path length caused by hierarchical routing is sufficiently small that it is usually acceptable.

Example:



Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Explanation

Step 1 − For example, the best path from 1A to 5C is via region 2, but hierarchical routing of all traffic to region 5 goes via region 3 as it is better for most of the other destinations of region 5.

Step 2 − Consider a subnet of 720 routers. If no hierarchy is used, each router will have 720 entries in its routing table.

Step 3 − Now if the subnet is partitioned into 24 regions of 30 routers each, then each router will require 30 local entries and 23 remote entries for a total of 53 entries.

Example

If the same subnet of 720 routers is partitioned into 8 clusters, each containing 9 regions and each region containing 10 routers. Then what will be the total number of table entries in each router.

Solution

10 local entries + 8 remote regions + 7 clusters = 25 entries.

Congestion Control

Congestion causes choking of the communication channel. When too many packets are displayed in a part of the subnet, the subnet's performance degrades. Hence, the network's communication channel is called congested if packets are traversing the path experience primarily over the path propagation delay.

It is known as heavily congested when the packets never reach the destination, denoting the delay method infinity. When the input traffic rate exceeds the output lines capacity, the subnet's input part gets choked and generates congestion. It also happens when the routers are too slow to execute queuing buffers, refreshing tables, etc. The loss of capacity of the routers' buffer is also one of the many factors for congestion. However, enhancing the memory of the router may be helpful up to a certain point.

Principles of Congestion Control

As per control theory, the computer network, which is also a system, is divided into two groups. They are open-loop and closed-loop solutions.

The open-loop solutions

It provides an excellent design to ensure that the problem does not occur in the first place. The designing tools include deciding to accept new traffic, discarding packets and scheduling the packets at various network points. The open-loop solution's decisions are independent of the current state of the network.

Closed-loop solutions

It makes the decision based on the concept of a feedback loop. The feedback Notes loop enables the closed-loop system to monitor the procedure to detect when and where congestion occurs. After that, it passes the information to the places where they can take actions.

The system's monitoring depends on the percentage of all packets discarded for drawback of buffer area, the average queue lengths, the multiple packets that time out and are retransmitted, the normal packet delay and the standard deviation of packet delay.

Secondly, hosts or routers share packets periodically to directly know about congestion so that the traffic around congested areas can be routed to alternate destination routes.

The congestion can be controlled as given below:

- It can raise the bandwidth in the network. It can be increased if an additional line temporarily increases the bandwidth between specific points.
- It is used to split traffic to follow multiple routes.
- It can increase the resources. For example, use spare routers.
- It is used to decrease the load by denying service to some users or degrading service to some or all users.
- It is used to estimate users' schedule and demands in a more predictable way.

Congestion prevention policies

The open loop systems are designed to minimize congestion in the first place, rather than letting it happen and reacting after the fact. They try to achieve their goal by using appropriate policies at various levels. In Fig.4 there are different data link, network, and transport policies that can affect congestion (Jain, 1990).

The retransmission policy is concerned with how fast a sender times out and what it transmits upon timeout. A jumpy sender that times out quickly and retransmits all outstanding packets using go back n will put a heavier load on the system than will a leisurely sender that uses selective repeat. Closely related to this is the buffering policy. If receivers routinely discard all out-of-order packets, these packets will have to be transmitted again later, creating extra load.

Acknowledgement policy also affects congestion. If each packet is acknowledged immediately, the acknowledgement packets generate extra traffic. However, if acknowledgements are saved up to piggyback onto reverse traffic, extra timeouts and retransmissions may result. A tight flow control scheme (e.g., a small window) reduces the data rate and thus helps fight congestion.

At the network layer, the choice between using virtual circuits and using datagrams affects congestion since many congestion control algorithms work only with virtual-circuit subnets. Packet queueing and service policy relates to whether routers have one queue per input line, one queue per output line, or both. It also relates to the order in which packets are processed (e.g., round robin or priority based). Discard policy is the rule telling which packet is dropped when there is no space. A good policy can help alleviate congestion and a bad one can make it worse.

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Fig. 4 Policies that affect congestion.

A good routing algorithm can help avoid congestion by spreading the traffic over all the lines whereas a bad one can send too much traffic over already congested lines. Finally, packet lifetime management deals with how long a packet may live before being discarded. If it is too long, lost packets may clog up the works for a long time, but if it is too short, packets may sometimes time out before reaching their destination, thus inducing retransmissions.

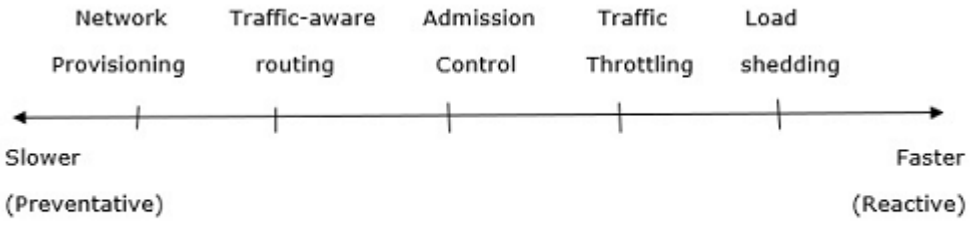
In the transport layer, the same issues occur as in the data link layer, but in addition, determining the timeout interval is harder because the transit time across the network is less predictable than the transit time over a wire between two routers. If the timeout interval is too short, extra packets will be sent unnecessarily. If it is too long, congestion will be reduced but the response time will suffer whenever a packet is lost.

Approaches to congestion control

The presence of congestion means the load is greater than the resources available over a network to handle. Generally we will get an idea to reduce the congestion by trying to increase the resources or decrease the load, but it is not that much of a good idea.

Approaches to Congestion Control

There are some approaches for congestion control over a network which are usually applied on different time scales to either prevent congestion or react to it once it has occurred.



Time scale of approaches to congestion control

Let us understand these approaches step wise as mentioned below −

Step 1: The basic way to avoid congestion is to build a network that is well matched to the traffic that it carries. If more traffic is directed but a low-bandwidth link is available, definitely congestion occurs.

Step 2: Sometimes resources can be added dynamically like routers and links when there is serious congestion. This is called provisioning, and which happens on a timescale of months, driven by long-term trends.

Step 3: To utilize most existing network capacity, routers can be tailored to traffic patterns making them active during daytime when network users are using more and sleep in different time zones.

Step 4: Some of local radio stations have helicopters flying around their cities to report on road congestion to make it possible for their mobile listeners to route their packets (cars) around hotspots. This is called traffic aware routing.

Step 5: Sometimes it is not possible to increase capacity. The only way to reduce the congestion is to decrease the load. In a virtual circuit network, new connections can be refused if they would cause the network to become congested. This is called admission control.

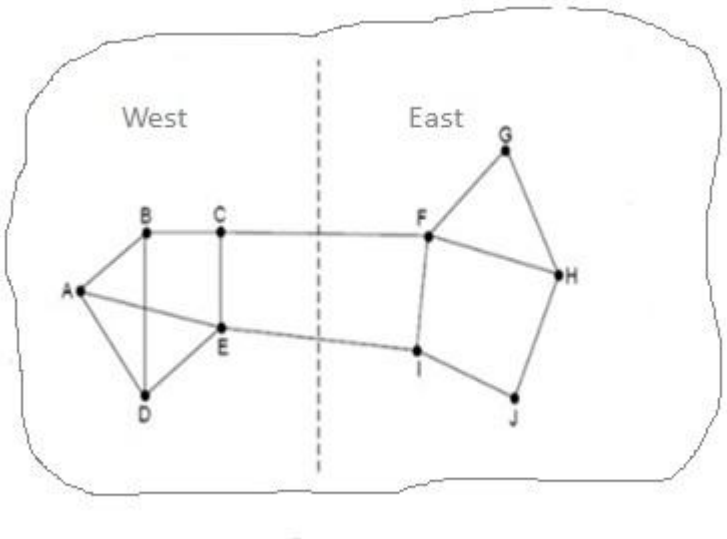
Step 6: Routers can monitor the average load, queueing delay, or packet loss. In all these cases, the rising number indicates growing congestion. The network is forced to discard packets that it cannot deliver. The general name for this is Load shedding. The better technique for choosing which packets to discard can help to prevent congestion collapse.

Traffic aware routing

Traffic awareness is one of the approaches for congestion control over the network. The basic way to avoid congestion is to build a network that is well matched to the traffic that it carries. If more traffic is directed but a low-bandwidth link is available, congestion occurs.

The main goal of traffic aware routing is to identify the best routes by considering the load, set the link weight to be a function of fixed link bandwidth and propagation delay and the variable measured load or average queuing delay.

Least-weight paths will then favor paths that are more lightly loaded, remaining all are equal.
The traffic aware routing is diagrammatically represented as follows



Explanation

Step 1: Consider a network which is divided into two parts, East and West both are connected by links CF and EI.

Step 2: Suppose most of the traffic in between East and West is using link CF, and as a result CF link is heavily loaded with long delays. Including queueing delay in the weight which is used for shortest path calculation will make EI more attractive.

Step 3: After installing the new routing tables, most of East-West traffic will now go over the EI link. As a result in the next update CF link will appear to be the shortest path.

Step 4: As a result the routing tables may oscillate widely, leading to erratic routing and many potential problems.

Step 5: If we consider only bandwidth and propagation delay by ignoring the load, this problem does not occur. Attempts to include load but change the weights within routing scheme to shift traffic across routes arrow range only to slow down routing oscillations.

Step 6: Two techniques can contribute for successful solution, which are as follows −

- Multipath routing
- The routing scheme to shift traffic across routes.

Features

The features of traffic aware routing are as follows:

- It is one of the congestion control techniques.
- To utilise most existing network capacity, routers can be tailored to traffic patterns making them active during daytime when network users are using more and sleep in different time zones.
- Roots can be changed to shift traffic away because of heavily used paths.
- Network Traffic can be split across multiple paths.

Admission control approach

The presence of congestion means the load is greater than the resources available over a network to handle. Generally, we will get an idea to reduce the congestion by trying to increase the resources or decrease the load, but it is not that much of a good idea.

There are some approaches for congestion control over a network which are usually applied on different time scales to either prevent congestion or react to it once it has occurred.

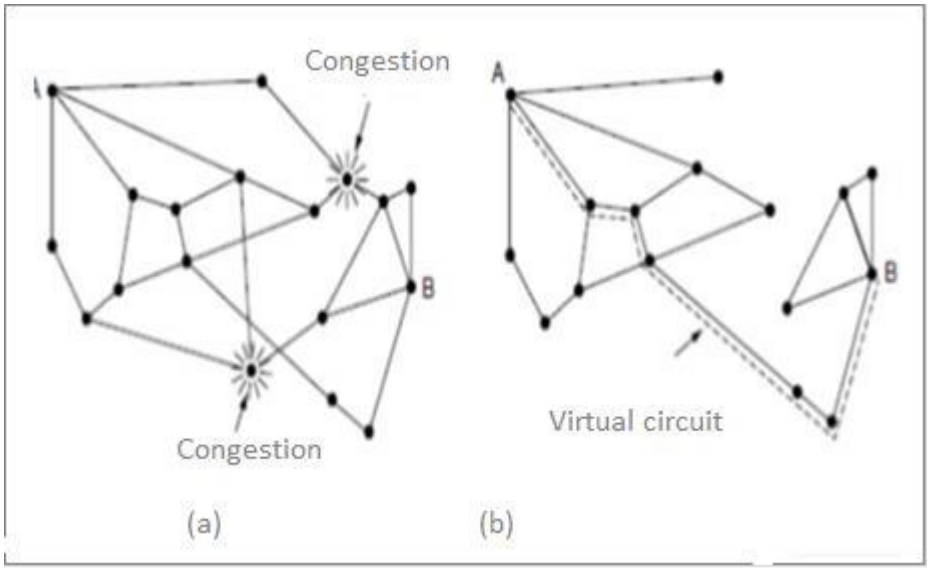
Admission Control

It is one of techniques that is widely used in virtual-circuit networks to keep congestion at bay. The idea is do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.

Admission control can also be combined with traffic aware routing by considering routes around traffic hotspots as part of the setup procedure.

Example

Take two networks (a) A congestion network and (b) The portion of the network that is not congested. A virtual circuit A to B is also shown below



Explanation

Step 1: Suppose a host attached to router A wants to set up a connection to a host attached to router B. Normally this connection passes through one of the congested routers.

Step 2: To avoid this situation, we can redraw the network as shown in figure (b), removing the congested routers and all of their lines.

Step 3: The dashed line indicates a possible route for the virtual circuit that avoids the congested routers.

Traffic Throttling

Traffic throttling is one of the approaches for congestion control. In the internet and other computer networks, senders trying to adjust the transmission need to send as much traffic as the network can readily deliver. In this setting the network aim is to operate just before the onset of congestion.

There are some approaches to throttling traffic that can be used in both datagram and virtual-circuit networks.

Each approach has to solve two problems:

Firs

Routers have to determine when congestion is approaching ideally before it has arrived. Each router can continuously monitor the resources it is using.

There are three possibilities, which are as follows:

- Utilisation of output links.
- Buffering of queued packets inside the router.
- Numbers of packets are lost due to insufficient buffering.

Second

Average of utilization does not directly account for burstiness of most traffic and queueing delay inside routers directly captures any congestion experienced by packets.

To manage the good estimation of queueing delay d, a sample of queue length s, can be made periodically and d updated according to,

$$d_{new}=\alpha d_{old}+(1-\alpha)s$$

Where the constant α determines how fast the router forgets recent history. This is called EWMA (Exponentially Weighted Moving Average)

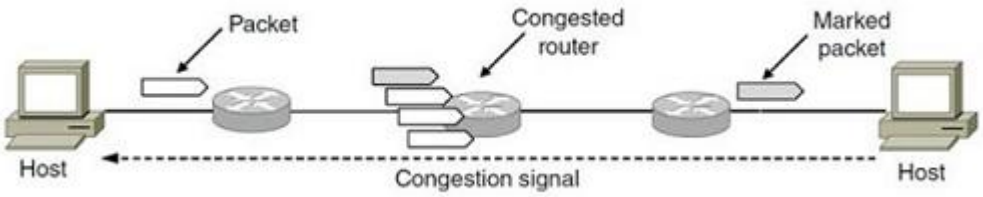
It smoothest out fluctuations and is equivalent to allow-pass filter. Whenever d moves above the threshold, the router notes the onset of congestion.

Routers must deliver timely feedback to the senders that are causing the congestion. Routers must also identify the appropriate senders. It must then warn carefully, without sending many more packets into an already congested network.

There are many feedback mechanisms one of them is as follows:

Explicit Congestion Notification (ECN)

The Explicit Congestion Notification (ECN) is diagrammatically represented as follows:



Explanation of ECN

Step 1: Instead of generating additional packets to warn of congestion, a router can tag any packet it forwards by setting a bit in the packet header to signal that it is experiencing congestion.

Step 2: When the network delivers the packet, the destination can note that there is congestion and inform the sender when it sends a reply packet.

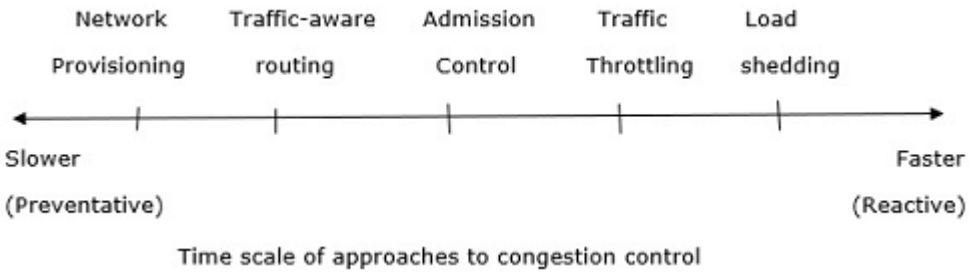
Step 3: The sender can then throttle its transmissions as before.

Step 4: This design is called explicit congestion notification and is mostly used on the Internet.

Load shedding

The presence of congestion means the load is greater than the resources available over a network to handle. Generally we will get an idea to reduce the congestion by trying to increase the resources or decrease the load, but it is not that much of a good idea.

There are some approaches for congestion control over a network which are usually applied on different time scales to either prevent congestion or react to it once it has occurred.



Time scale of approaches to congestion control

Now let us see one of the techniques for congestion control which is termed as load shedding:

Load Shedding

It is one of the approaches to congestion control. Router contains a buffer to store packets and route it to destination. When the buffer is full, it simply discards some packets. It chooses the packet to be discarded based on the strategy implemented in the data link layer. This is called load shedding.

Load shedding will use dropping the old packets than new to avoid congestion. Dropping packets that are part of the difference is preferable because a future packet depends on full frame.

To implement an intelligent discard policy, applications must mark their packets to indicate to the network how important they are. When packets have to be discarded, routers can first drop packets from the least important class, then the next most important class, and so on.

Advantages

The advantages of load shedding are given below:

- It can be used in detection of congestion.
- It can recover from congestion.
- It reduces the network traffic flow.
- Synchronised flow of packets across a network.
- Removes the packets before congestion occurs.

Disadvantages

The disadvantages of load shedding are given below:

- Packets get lost because of discarding by the router.
- If buffer size is less it results in more packets to get discarded.

- Cannot ensure congestion avoidance.
- Overhead for the router to always keep on checking whether the buffer is full.

Traffic Control Algorithm

In the network layer, before the network can make Quality of service guarantees, it must know what traffic is being guaranteed. One of the main causes of congestion is that traffic is often bursty.

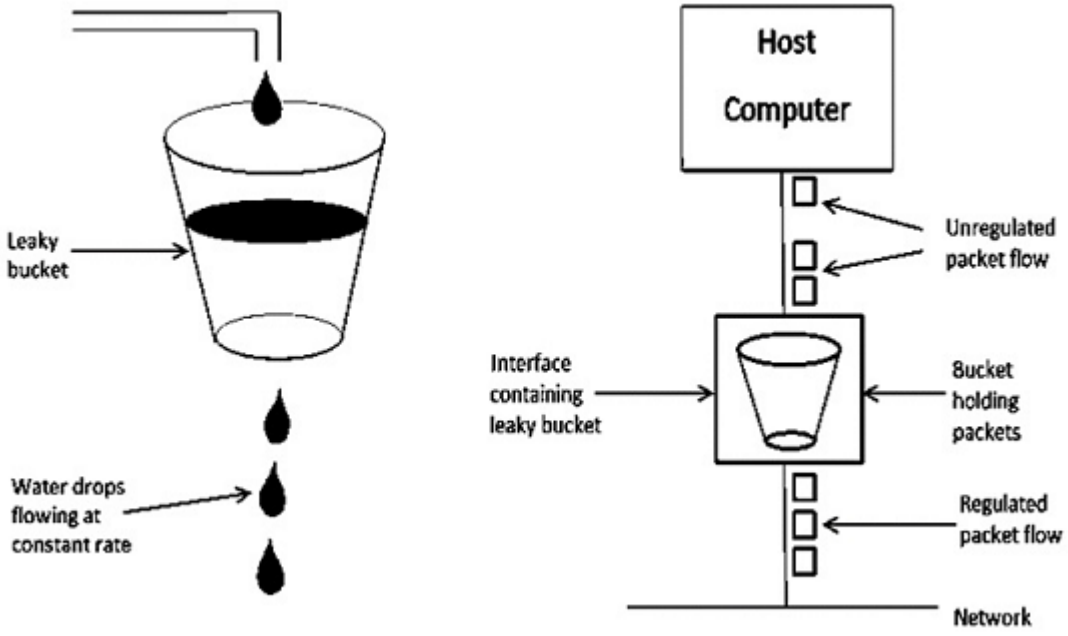
To understand this concept first we have to know little about traffic shaping. Traffic Shaping is a mechanism to control the amount and the rate of traffic sent to the network. Approach of congestion management is called Traffic shaping. Traffic shaping helps to regulate the rate of data transmission and reduces congestion.

There are 2 types of traffic shaping algorithms:

- Leaky Bucket
- Token Bucket

Leaky Bucket Algorithm

Let see the working condition of Leaky Bucket Algorithm:



Leaky Bucket Algorithm mainly controls the total amount and the rate of the traffic sent to the network.

Step 1: Let us imagine a bucket with a small hole at the bottom where the rate at which water is poured into the bucket is not constant and can vary but it leaks from the bucket at a constant rate.

Step 2: So (up to water is present in the bucket), the rate at which the water leaks does not depend on the rate at which the water is input to the bucket.

Step 3: If the bucket is full, additional water that enters into the bucket that spills over the sides and is lost.

Step 4: Thus the same concept applied to packets in the network. Consider that data is coming from the source at variable speeds. Suppose that a source sends data at 10 Mbps for 4 seconds. Then there is no data for 3 seconds. The source again transmits data at a rate of 8 Mbps for 2 seconds. Thus, in a time span of 8 seconds, 68 Mb data has been transmitted.

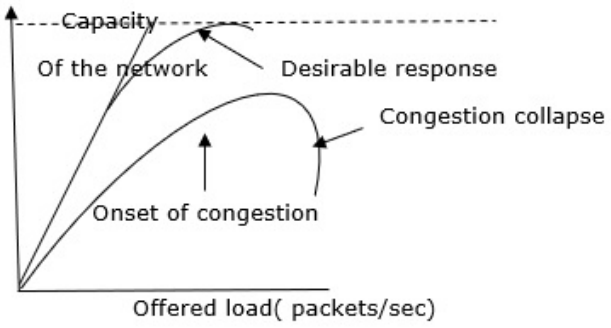
That's why if a leaky bucket algorithm is used, the data flow would be 8 Mbps for 9 seconds. Thus, the constant flow is maintained.

Token Bucket algorithm

Token bucket algorithm is one of the techniques for congestion control algorithms. When too many packets are present in the network it causes packet delay and loss of packet which degrades the performance of the system. This situation is called congestion.

The network layer and transport layer share the responsibility for handling congestions. One of the most effective ways to control congestion is trying to reduce the load that transport layer is placing on the network. To maintain this network and transport layers have to work together.

The Token Bucket Algorithm is diagrammatically represented as follows:



With too much traffic, performance drops sharply.

Token Bucket Algorithm

The leaky bucket algorithm enforces output patterns at the average rate, no matter how busy the traffic is. So, to deal with the more traffic, we need a flexible algorithm so that the data is not lost. One such approach is the token bucket algorithm.

Let us understand this algorithm step wise as given below:

Step 1: In regular intervals tokens are thrown into the bucket f.

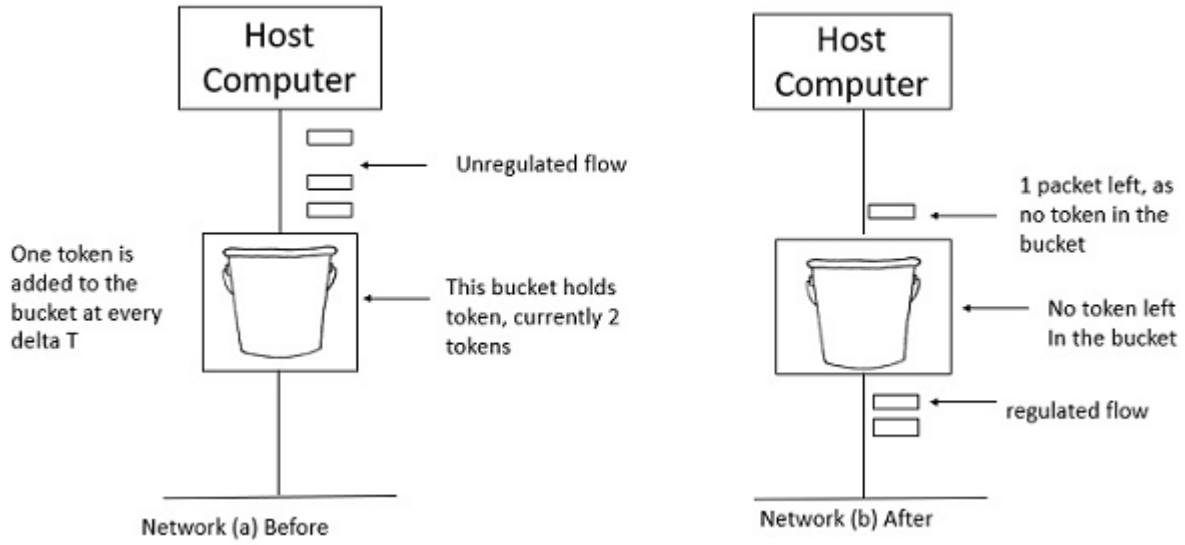
Step 2: The bucket has a maximum capacity f.

Step 3: If the packet is ready, then a token is removed from the bucket, and the packet is sent.

Step 4: Suppose, if there is no token in the bucket, the packet cannot be sent.

Example

Let us understand the Token Bucket Algorithm with an example:

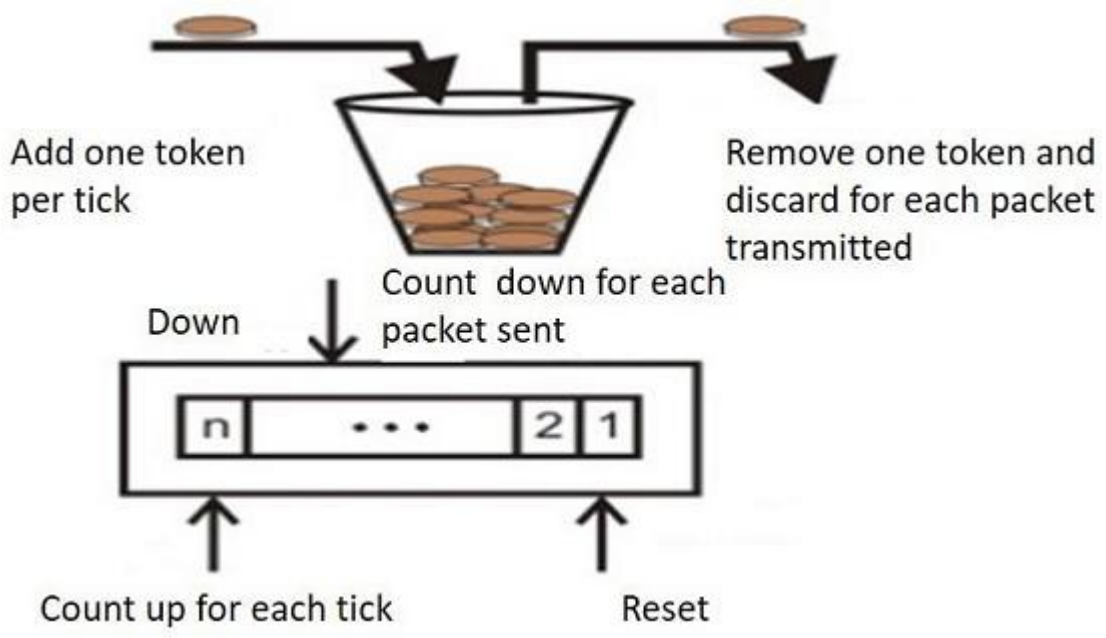


In figure (a) the bucket holds two tokens, and three packets are waiting to be sent out of the interface.

In Figure (b) two packets have been sent out by consuming two tokens, and 1 packet is still left.

When compared to Leaky bucket the token bucket algorithm is less restrictive that means it allows more traffic. The limit of busyness is restricted by the number of tokens available in the bucket at a particular instant of time.

The implementation of the token bucket algorithm is easy − a variable is used to count the tokens. For every t seconds the counter is incremented and then it is decremented whenever a packet is sent. When the counter reaches zero, no further packet is sent out.
This is shown in below given diagram:



Internetworking

Internetworking is combined of 2 words, inter and networking which implies an association between totally different nodes or segments. This connection area unit is established through intercessor devices akin to routers or gateway. The first term for associate degree internetwork was catenet. This interconnection is often among or between public, private, commercial, industrial, or governmental networks. Thus, associate degree internetwork could be an assortment of individual networks, connected by intermediate networking devices, that function as one giant network. Internetworking

refers to the trade, products, and procedures that meet the challenge of making and administering internet works.

To enable communication, every individual network node or phase is designed with a similar protocol or communication logic, that is Transfer Control Protocol (TCP) or Internet Protocol (IP). Once a network communicates with another network having constant communication procedures, it's called Internetworking. Internetworking was designed to resolve the matter of delivering a packet of information through many links.

There is a minute difference between extending the network and Internetworking. Merely exploitation of either a switch or a hub to attach 2 local area networks is an extension of LAN whereas connecting them via the router is an associate degree example of Internetworking. Internetworking is enforced in Layer three (Network Layer) of the OSI-ISO model. The foremost notable example of internetworking is the Internet.

Types of Internet working:

There is chiefly 3 units of Internetworking:

- Extranet
- Intranet
- Internet

Intranets and extranets might or might not have connections to the net. If there is a connection to the net, the computer network or extranet area unit is usually shielded from being accessed from the net if it is not authorized. The net isn't thought-about to be a section of the computer network or extranet, though it should function as a portal for access to parts of the associate degree extranet.

Extranet: It's a network of the internetwork that's restricted in scope to one organization or entity however that additionally has restricted connections to the networks of one or a lot of different sometimes, however not essential. It's the very lowest level of Internetworking, usually enforced in an exceedingly personal area. Associate degree extranet may additionally be classified as a Man, WAN, or different form of network however it cannot encompass one local area network i.e. it should have a minimum of one reference to associate degree external network.

Intranet: This associate degree computer network could be a set of interconnected networks, which exploits the Internet Protocol and uses IP-based tools akin to web browsers and FTP tools, that are underneath the management of one body entity. That body entity closes the computer network to the remainder of the planet and permits solely specific users. Most typically, this network is the internal network of a corporation or different enterprise. An outsized computer network can usually have its own internet server to supply users with browsable data.

Internet: A selected Internetworking, consisting of a worldwide interconnection of governmental, academic, public, and personal networks based mostly upon the Advanced analysis comes Agency Network (ARPANET) developed by ARPA of the U.S. Department of Defense additionally home to the World Wide Web (WWW) and cited as the 'Internet' to differentiate from all different generic Internetworks. Participants within the web, or their service suppliers, use IP Addresses obtained from address registries that manage assignments.

How Networks Can Be Connected

Networks can be interconnected by different devices. The following section will explain the network connection in different way.

Network connection in Different layer:

- In the physical layer, networks can be connected by repeaters or hubs, which just move the bits from one network to an identical network.
  - These are mostly analog devices and do not understand anything about digital protocols (they just regenerate signals).
- In the data link layer we find bridges and switches, which operates.
  - They can accept frames, examine the MAC addresses, and forward the frames to a different network while doing minor protocol translation in the process..
- In the network layer, we have routers that can connect two networks.
  - If two networks have dissimilar network layers, the router may be able to translate between the packet formats, although packet translation is now increasingly rare.
  - A router that can handle multiple protocols is called a multiprotocol router.

- In the transport layer we find transport gateways, which can interface between two transport connections.
  - For example, a transport gateway could allow packets to flow between a TCP network and an SNA network, which has a different transport protocol, by essentially gluing a TCP connection to an SNA connection.
- In the application layer, application gateways translate message semantics.
  - As an example, gateways between Internet e-mail (RFC 822) and X.400 e-mail must parse the e-mail messages and change various header fields.

How interworking in Network layer differs from switching in data link layer?

- In Fig. 3.3(a), consider the source machine, S, wants to send a packet to the destination machine, D.
- These machines are on different Ethernets, connected by a switch.
- S encapsulates the packet in a frame and sends it on its way.
- The frame arrives at the switch, which then determines that the frame has to go to LAN 2 by looking at its MAC address.
- The switch just removes the frame from LAN 1 and deposits it on LAN 2.

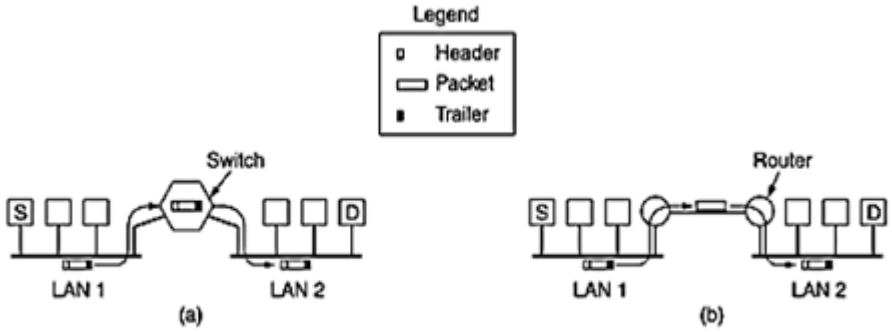


Figure3.3 (a) Two Ethernets connected by a switch. (b) Two Ethernets connected by routers.

- Now let us consider the same situation but with the two Ethernets connected by a pair of routers instead of a switch.
- The routers are connected by a point-to-point line, possibly a leased line thousands of kilometers long.
- Now the frame is picked up by the router and the packet removed from the frame's data field.

- The router examines the address in the packet (e.g., an IP address) and looks up this address in its routing table.
- Based on this address, it decides to send the packet to the remote router, potentially encapsulated in a different kind of frame, depending on the line protocol.
- At the far end, the packet is put into the data field of an Ethernet frame and deposited onto LAN 2.
- An essential difference between the switched (or bridged) case and the routed case is this.
- With a switch (or bridge), the entire frame is transported on the basis of its MAC address.
- With a router, the packet is extracted from the frame and the address in the packet is used for deciding where to send it. Switches do not have to understand the network layer protocol being used to switch packets. Routers do.

Tunnelling

Tunnelling is a protocol for transferring data securely from one network to another. Using a method known as encapsulation, Tunnelling allows private network communications to be sent across a public network, such as the Internet. Encapsulation enables data packets to appear general to a public network when they are private data packets, allowing them to pass unnoticed.

Note: Port forwarding is another name for Tunnelling.

When data is tunnelled, it is split into smaller parts called packets, as it travels through the tunnel. The packets are encrypted via the tunnel, and another process known as encapsulation takes place. For transmission, private network data and protocol details are encased in public network transmission units. The units have the appearance of public data, allowing them to be sent via the Internet. Encapsulation enables packets to reach their intended destination. De-capsulation and decryption take place at the final destination.

Tunnelling is possible thanks to a variety of procedures, including:

- Point-to-Point Tunnelling Protocol (PPTP)
- Layer Two Tunnelling Protocol (L2TP)

PPTP (Point-to-Point Tunnelling Protocol)

PPTP protects confidential information even when transmitted via public networks. An Internet service provider can provide authorized users with access to a private network called a virtual private network. Because it was built in a tunnelled environment, this is a "virtual" private network.

Layer Two Tunnelling Protocol (L2TP)

This tunnelling protocol combines PPTP with Layer 2 Forwarding.

Tunnelling is a technique for communicating over a public network while going through a private network. This is especially beneficial in a corporate situation, and it also includes security measures like encryption.

The IP packet in this scenario does not have to deal with the WAN, and neither do the hosts A and B. IP, and WAN packets will be understood by the multiprotocol routers M1 and M2. As a result, the WAN can be compared to a large tunnel connecting multiprotocol routers M1 and M2, and the process is known as Tunnelling.

Tunnelling makes use of a layered protocol paradigm like the OSI or TCP/IP protocol suite. In other words, when data travels from host A to host B, it traverses all levels of the specified protocol (OSI, TCP/IP, and so on), and data conversion (encapsulation) to suit different interfaces of the particular layer is referred to as Tunnelling.

Applications of Tunnelling

Several protocols use a public network, such as the Internet, to transfer private network data by establishing a VPN (Virtual Private Network), making data transmissions more secure, especially when using unencrypted data.

IPsec (GPRS tunnelling protocol), SSH (Secure Socket Tunnelling Protocol), PPTP (Point-to-Point Tunnelling Protocol), and others are standard protocols, each designed for a specific tunnelling task or purpose.

Some examples of how tunnelling protocols are used are as follows:

- Although a foreign protocol is not supported to run over a specific network, a tunnelling protocol can run IP-v6 over IP-v4.
- When the corporate network does not include the user's physical network address, it is also used to deliver unfeasible fundamental network services, such as a corporate network address) to a remote user.
- Tunnelling allows users to get around a firewall by using an unblocked protocol such as HTTP and the technique of "wrapping" to piggyback/ slip past the firewall rules.
- Another option is to use the HTTP CONNECT tunnel's command/ technique. The HTTP proxy establishes a TCP connection to a specific server when the client issues an HTTP CONNECT command to the proxy server. This security flaw is exploited to use the HTTP proxy to transmit data between the client connection and the designated port. Usually, HTTP proxies enable connections like 443 but deny other proxy servers' access to the CONNECT command.

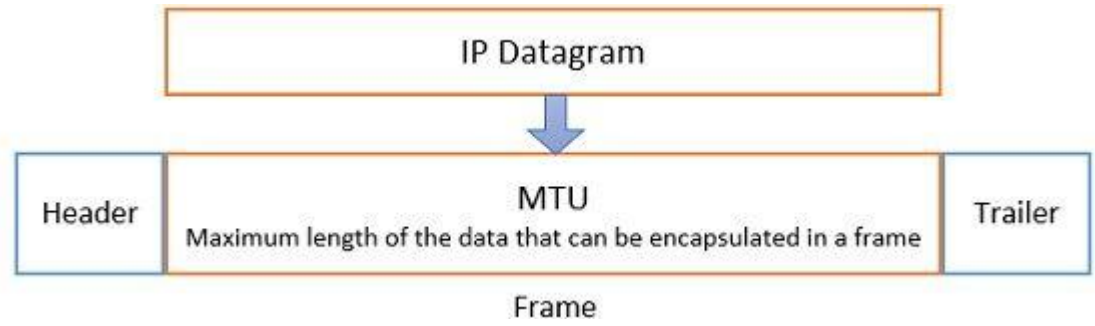
Fragmentation

Fragmentation is a technique implemented on a datagram at the network layer if the size of the datagram is larger than the size of the datagram that the corresponding network can forward. This technique involves the division of the large size datagram into smaller fragments. In this section, we will study all about the fragmentation of datagram in the network layer.

Fragmentation in Networking

In our previous content, we have discussed the functions of the network layer where we have seen that the network layer receives the data to be transmitted from its upper layer along with some other pieces of information such as the length of data, ID of the protocol used, type of service logical address of the destination, and the logical address of the destination computer.

The network layer encapsulates this data by adding the header to this data and forms a datagram. Now the datagram is formed and further, we need to know why this datagram is fragmented.



Why Fragmentation is Needed?

The datagram created by the network layer at the source computer has to pass through several networks before it reaches the destination computer. Usually, the source computer prefers to send the datagrams of large size.

This is because if the datagram is split into small fragments, then for each fragmented datagram unit the header is repeated. This repetition of the header for each fragmented datagram wastes the network bandwidth.

But over this event, every network has a certain limit over the maximum size of the packet it can forward. The source computer is even not aware of the path the packet will take to reach the destination. So, it is unable to decide how small each fragmented datagram must be.

The limit over the size of a packet has several reasons behind it. We have discussed those reason in the list below:

- Hardware and operating system used puts a limit on the size of data.
- Protocols at each network allow different packet sizes.
- Compliance with the national and international standards.
- Large packets occupy the network for a long period as compared to small packets.
- Reduce the error induced during retransmission.

These are the reasons behind the fragmentation of a large size datagram in a small fragmented datagram. Before elaborating on fragmentation let us discuss the datagram on which fragmentation is implemented.

Internet protocol

Internet Protocols are a set of rules that governs the communication and exchange of data over the internet. Both the sender and receiver should follow the same protocols in order to communicate the data. In order to understand it better, let's take an example of a language. Any language has its own set of vocabulary and grammar which we need to know if we want to communicate in that language. Similarly, over the internet whenever we access a website or exchange some data with another device then these processes are governed by a set of rules called the internet protocols.

Working of internet protocol: The internet and many other data networks work by organizing data into small pieces called packets. Each large data sent between two network devices is divided into smaller packets by the underlying hardware and software. Each network protocol defines the rules for how its data packets must be organized in specific ways according to the protocols the network supports.

Why do we need protocols?

It may be that the sender and receiver of data are parts of different networks, located in different parts of the world having different data transfer rates. So, we need protocols to manage the flow control of data, access control of the link being shared in the communication channel. Suppose there is a sender X who has a data transmission rate of 10 Mbps. And, there is a receiver Y who has a data receiving rate of 5Mbps. Since the rate of receiving the data is slow so some data will be lost during transmission. In order to avoid this, the receiver Y needs to inform sender X about the speed mismatch so that the sender X can adjust its transmission rate. Similarly, the access control decides the node which will access the link shared in the communication channel at a particular instant of time. If not the transmitted data will collide if many computers send data simultaneously through the same link resulting in the corruption or loss of data.

Types of internet protocol

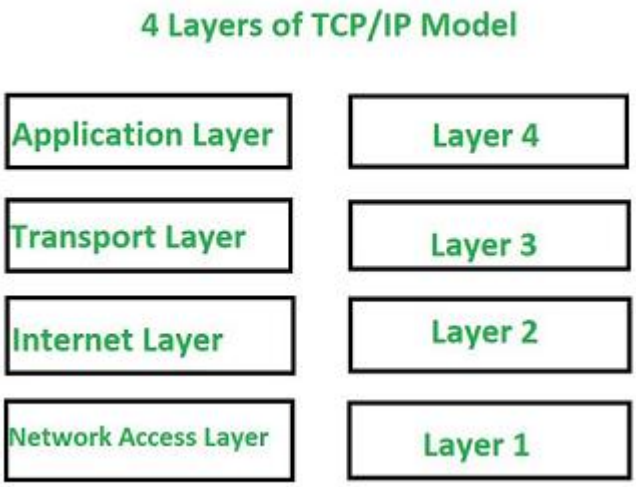
The Internet Protocols are of different types having different uses:-

1. TCP/IP(Transmission Control Protocol/ Internet Protocol): These are a set of standard rules that allows different types of computers to communicate with each other. The IP protocol ensures that each computer that is connected to the Internet is having a specific serial number called the IP

address. TCP specifies how data is exchanged over the internet and how it should be broken into IP packets. It also makes sure that the packets have information about the source of the message data, the destination of the message data, the sequence in which the message data should be re-assembled, and checks if the message has been sent correctly to the specific destination. The TCP is also known as a connection-oriented protocol.

The functionality of TCP/IP is divided into 4 layers with each one having specific protocols:

1. **Application Layer:** The application layer makes sure that the data from the sending end is received in a format that is acceptable and supported at the receiving end.
2. **Transport Layer:** The transport layer is responsible for the smooth transmission of data from one end to the other. It is also responsible for reliable connectivity, error recovery, and flow control of the data.
3. **Internet Layer:** This Internet Layer moves packets from source to destination by connecting independent networks.
4. **Network Access Layer:** The Network Access Layer sees how a computer connects to a network.



2. SMTP(Simple Mail Transfer Protocol): These protocols are important for sending and distributing outgoing emails. This protocol uses the header of the mail to get the email id of the receiver and enters the mail into the queue of outgoing mails. And as soon as, it delivers the mail to the receiving email id, it removes the email from the outgoing list. The message or the electronic

mail may consider of text, video, image etc. It helps in setting up of some communication server rules.

3. PPP(Point to Point Protocol): It is a communication protocol that is used to create a direct connection between two communicating devices. This protocol defines the rules using which two devices will authenticate with each other and exchange information with each other. For example, A user connects his PC to the server of an Internet Service Provider also uses PPP. Similarly, for connecting two routers for direct communication it uses PPP.

4. FTP (File Transfer Protocol): This protocol is used for transferring files from one system to the other. This works on a client-server model. When a machine requests for file transfer from another machine, the FTO sets up a connection between the two and authenticates each other using their ID and Password. And, the desired file transfer takes place between the machines.

5. SFTP(Secure File Transfer Protocol): SFTP which is also known as SSH FTP refers to File Transfer Protocol (FTP) over Secure Shell (SSH) as it encrypts both commands and data while in transmission. SFTP acts as an extension to SSH and encrypts files and data then sends them over a secure shell data stream. This protocol is used to remotely connect to other systems while executing commands from the command line.

6. HTTP(Hyper Text Transfer Protocol): This protocol is used to transfer hypertexts over the internet and it is defined by the www(world wide web) for information transfer. This protocol defines how the information needs to be formatted and transmitted. And, it also defines the various actions the web browsers should take in response to the calls made to access a particular web page. Whenever a user opens their web browser, the user will indirectly use HTTP as this is the protocol that is being used to share text, images, and other multimedia files on the World Wide Web.

Note: *Hypertext refers to the special format of the text that can contain links to other texts*.

7. HTTPS(HyperText Transfer Protocol Secure): HTTPS is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network with the

SSL/TLS protocol for encryption and authentication. So, generally, a website has an HTTP protocol but if the website is such that it receives some sensitive information such as credit card details, debit card details, OTP, etc then it requires an SSL certificate installed to make the website more secure. So, before entering any sensitive information on a website, we should check if the link is HTTPS or not. If it is not HTTPS then it may not be secure enough to enter sensitive information.

8. TELNET(Terminal Network): TELNET is a standard TCP/IP protocol used for virtual terminal service given by ISO. This enables one local machine to connect with another. The computer which is being connected is called a remote computer and which is connecting is called the local computer. TELNET operation lets us display anything being performed on the remote computer in the local computer. This operates on the client/server principle. The local computer uses the telnet client program whereas the remote computer uses the telnet server program.

9. POP3(Post Office Protocol 3): POP3 stands for Post Office Protocol version 3. It has two Message Access Agents (MAAs) where one is client MAA (Message Access Agent) and another is server MAA(Message Access Agent) for accessing the messages from the mailbox. This protocol helps us to retrieve and manage emails from the mailbox on the receiver mail server to the receiver's computer. This is implied between the receiver and receiver mail server. It can also be called as one way client server protocol. The POP3 WORKS ON THE 2 PORTS I.E. PORT 110 AND PORT 995.

IP Version 4 protocol

IP stands for Internet Protocol and v4 stands for Version Four (IPv4). IPv4 was the primary version brought into action for production within the ARPANET in 1983.

IP version four addresses are 32-bit integers which will be expressed in decimal notation.

Example- 192.0.2.126 could be an IPv4 address.

Parts of IPv4

- **Network part:**

The network part indicates the distinctive variety that's appointed to the network. The network part conjointly identifies the category of the network that's assigned.

- **Host Part:**

The host part uniquely identifies the machine on your network. This part of the IPv4 address is assigned to every host.

For each host on the network, the network part is the same, however, the host half must vary.

- **Subnet number:**

This is the nonobligatory part of IPv4. Local networks that have massive numbers of hosts are divided into subnets and subnet numbers are appointed to that.

Characteristics of IPv4

- IPv4 could be a 32-Bit IP Address.
- IPv4 could be a numeric address, and its bits are separated by a dot.
- The number of header fields is twelve and the length of the header field is twenty.
- It has Unicast, broadcast, and multicast style of addresses.
- IPv4 supports VLSM (Virtual Length Subnet Mask).
- IPv4 uses the Post Address Resolution Protocol to map to the MAC address.
- RIP may be a routing protocol supported by the routed daemon.
- Networks ought to be designed either manually or with DHCP.
- Packet fragmentation permits from routers and causing host.

Advantages of IPv4

- IPv4 security permits encryption to keep up privacy and security.
- IPV4 network allocation is significant and presently has quite 85000 practical routers.
- It becomes easy to attach multiple devices across an outsized network while not NAT.
- This is a model of communication so provides quality service also as economical knowledge transfer.
- IPV4 addresses are redefined and permit flawless encoding.

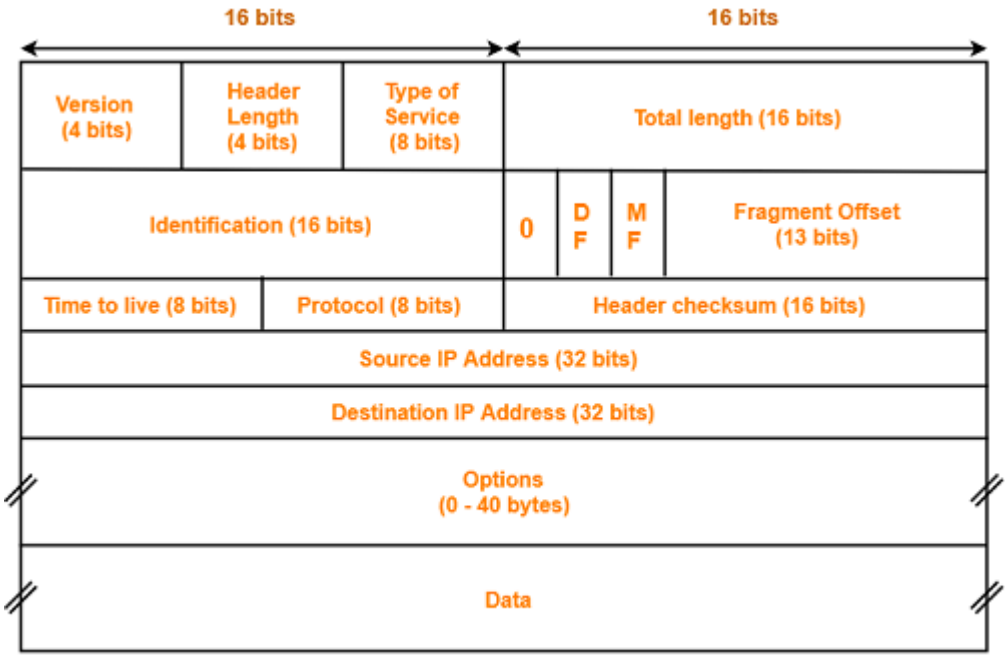
- Routing is a lot of scalable and economical as a result of addressing is collective more effectively.
- Data communication across the network becomes a lot of specific in multicast organizations.
  - Limits net growth for existing users and hinders the use of the net for brand new users.
  - Internet Routing is inefficient in IPv4.
  - IPv4 has high System Management prices and it's labor-intensive, complex, slow & frequent to errors.
  - Security features are nonobligatory.
  - Difficult to feature support for future desires as a result of adding it on is extremely high overhead since it hinders the flexibility to attach everything over IP.

Limitations of IPv4

- IP relies on network layer addresses to identify end-points on network, and each network has a unique IP address.
- The world's supply of unique IP addresses is dwindling, and they might eventually run out theoretically.
- If there are multiple host, we need IP addresses of next class.
- Complex host and routing configuration, non-hierarchical addressing, difficult to re-numbering addresses, large routing tables, non-trivial implementations in providing security, QoS (Quality of Service), mobility and multi-homing, multicasting etc. are the big limitation of IPv4 so that's why IPv6 came into the picture.

IPv4 Header-

The following diagram represents the IPv4 header-



IPv4 Header

1. Version-

- Version is a 4 bit field that indicates the IP version used.
- The most popularly used IP versions are version-4 (IPv4) and version-6 (IPv6).
- Only IPv4 uses the above header.
- So, this field always contains the decimal value 4.

NOTES

It is important to note-

- Datagrams belonging to different versions have different structures.
- So, they are parsed differently.
- IPv4 datagrams are parsed by version-4 parsers.
- IPv6 datagrams are parsed by version-6 parsers.

2. Header Length:

- Header length is a 4 bit field that contains the length of the IP header.
- It helps in knowing from where the actual data begins.

Minimum And Maximum Header Length:

The length of IP header always lies in the range- [20 bytes , 60 bytes]

- The initial 5 rows of the IP header are always used.
- So, minimum length of IP header = 5 x 4 bytes = 20 bytes.
- The size of the 6th row representing the Options field vary.
- The size of Options field can go up to 40 bytes.
- So, maximum length of IP header = 20 bytes + 40 bytes = 60 bytes.

Concept of Scaling Factor

- Header length is a 4 bit field.
- So, the range of decimal values that can be represented is [0, 15].
- But the range of header length is [20, 60].
- So, to represent the header length, we use a scaling factor of 4.

In general,

Header length = Header length field value x 4 bytes

Examples:

- If header length field contains decimal value 5 (represented as 0101), then-

  Header length = 5 x 4 = 20 bytes
- If header length field contains decimal value 10 (represented as 1010), then-

  Header length = 10 x 4 = 40 bytes
- If header length field contains decimal value 15 (represented as 1111), then-

  Header length = 15 x 4 = 60 bytes

NOTES

It is important to note-

- Header length and Header length field value are two different things.

- The range of header length field value is always [5, 15].
- The range of header length is always [20, 60].

While solving questions-

- If the given value lies in the range [5, 15] then it must be the header length field value.
- This is because the range of header length is always [20, 60].

3. Type Of Service:

- Type of service is a 8 bit field that is used for Quality of Service (QoS).
- The datagram is marked for giving a certain treatment using this field.

4. Total Length:

- Total length is a 16 bit field that contains the total length of the datagram (in bytes).

Total length = Header length + Payload length

- Minimum total length of datagram = 20 bytes (20 bytes header + 0 bytes data)
- Maximum total length of datagram = Maximum value of 16 bit word = 65535 bytes

5. Identification:

- Identification is a 16 bit field.
- It is used for the identification of the fragments of an original IP datagram.

When an IP datagram is fragmented,

- Each fragmented datagram is assigned the same identification number.
- This number is useful during the re assembly of fragmented datagrams.
- It helps to identify to which IP datagram, the fragmented datagram belongs to.

6. DF Bit:

- DF bit stands for Do Not Fragment bit.
- Its value may be 0 or 1.

When DF bit is set to 0,

- It grants the permission to the intermediate devices to fragment the datagram if required.

When DF bit is set to 1,

- It indicates the intermediate devices not to fragment the IP datagram at any cost.
- If network requires the datagram to be fragmented to travel further but settings does not allow its fragmentation, then it is discarded.
- An error message is sent to the sender saying that the datagram has been discarded due to its settings.

7. MF Bit:

- MF bit stands for More Fragments bit.
- Its value may be 0 or 1.

When MF bit is set to 0,

- It indicates to the receiver that the current datagram is either the last fragment in the set or that it is the only fragment.

When MF bit is set to 1,

- It indicates to the receiver that the current datagram is a fragment of some larger datagram.
- More fragments are following.
- MF bit is set to 1 on all the fragments except the last one.

8. Fragment Offset:

- Fragment Offset is a 13 bit field.
- It indicates the position of a fragmented datagram in the original unfragmented IP datagram.
- The first fragmented datagram has a fragment offset of zero.

Fragment offset for a given fragmented datagram = Number of data bytes ahead of it in the original unfragmented datagram

Concept Of Scaling Factor

- We use a scaling factor of 8 for the fragment offset.
- Fragment offset field value = Fragment Offset / 8

Need Of Scaling Factor For Fragment Offset

- In IPv4 header, the total length field comprises of 16 bits.
- Total length = Header length + Payload length.
- Minimum header length = 20 bytes.
- So, maximum amount of data that can be sent in the payload field = 216 – 20 bytes.
- In worst case, a datagram containing 216 – 20 bytes of data might be fragmented in such a way that the last fragmented datagram contains only 1 byte of data.
- Then, fragment offset for the last fragmented datagram will be (216 – 20) – 1 = 216 -21 ≅ 216

(if no scaling factor is used)

- Now, this fragment offset value of 216 can not be represented.
- This is because the fragment offset field consists of only 13 bits.
- Using 13 bits, a maximum number of 213 can be represented.
- So, to represent 216 we use the concept of scaling factor.
- Scaling factor = 216 / 213 = 23 = 8.

9. Time To Live

- Time to live (TTL) is a 8 bit field.
- It indicates the maximum number of hops a datagram can take to reach the destination.
- The main purpose of TTL is to prevent the IP datagrams from looping around forever in a routing loop.

The value of TTL is decremented by 1 when-

- Datagram takes a hop to any intermediate device having network layer.
- Datagram takes a hop to the destination.

If the value of TTL becomes zero before reaching the destination, then datagram is discarded.

NOTES

It is important to note

- Both intermediate devices having network layer and destination decrements the TTL value by 1.
- If the value of TTL is found to be zero at any intermediate device, then the datagram is discarded.
- So, at any intermediate device, the value of TTL must be greater than zero to proceed further.
- If the value of TTL becomes zero at the destination, then the datagram is accepted.
- So, at the destination, the value of TTL may be greater than or equal to zero.

10. Protocol:

- Protocol is a 8 bit field.
- It tells the network layer at the destination host to which protocol the IP datagram belongs to.
- In other words, it tells the next level protocol to the network layer at the destination side.
- Protocol number of ICMP is 1, IGMP is 2, TCP is 6 and UDP is 17.

11. Header Checksum:

- Header checksum is a 16 bit field.
- It contains the checksum value of the entire header.
- The checksum value is used for error checking of the header.

At each hop,

- The header checksum is compared with the value contained in this field.
- If header checksum is found to be mismatched, then the datagram is discarded.
- Router updates the checksum field whenever it modifies the datagram header.

The fields that may be modified are:

1. TTL
2. Options

3. Datagram Length
4. Header Length
5. Fragment Offset

NOTE

It is important to note-

- Computation of header checksum includes IP header only.
- Errors in the data field are handled by the encapsulated protocol.

12. Source IP Address:

- Source IP Address is a 32 bit field.
- It contains the logical address of the sender of the datagram.

13. Destination IP Address:

- Destination IP Address is a 32 bit field.
- It contains the logical address of the receiver of the datagram.

14. Options:

- Options is a field whose size vary from 0 bytes to 40 bytes.
- This field is used for several purposes such as-
  1. Record route
  2. Source routing
  3. Padding

1. Record Route

- A record route option is used to record the IP Address of the routers through which the datagram passes on its way.
- When record route option is set in the options field, IP Address of the router gets recorded in the Options field.

NOTE

The maximum number of IPv4 router addresses that can be recorded in the Record Route option field of an IPv4 header is 9.

Explanation:

- In IPv4, size of IP Addresses = 32 bits = 4 bytes.
- Maximum size of Options field = 40 bytes.
- So, it seems maximum number of IP Addresses that can be recorded = 40 / 4 = 10.
- But some space is required to indicate the type of option being used.
- Also, some space is to be left between the IP Addresses.
- So, the space of 4 bytes is left for this purpose.
- Therefore, the maximum number of IP addresses that can be recorded = 9.

2. Source Routing:

- A source routing option is used to specify the route that the datagram must take to reach the destination.
- This option is generally used to check whether a certain path is working fine or not.
- Source routing may be loose or strict.

3. Padding:

- Addition of dummy data to fill up unused space in the transmission unit and make it conform to the standard size is called as padding.
- Options field is used for padding.

Example:

- When header length is not a multiple of 4, extra zeroes are padded in the Options field.
- By doing so, header length becomes a multiple of 4.
- If header length = 30 bytes, 2 bytes of dummy data is added to the header.
- This makes header length = 32 bytes.
- Then, the value 32 / 4 = 8 is put in the header length field.

- In worst case, 3 bytes of dummy data might have to be padded to make the header length a multiple of 4.

Address Format IPv4

The address format of IPv4 is represented into 4-octets (32-bit), which is divided into three different classes, namely class A, class B, and class C.

The above diagram shows the address format of IPv4. An IPv4 is a 32-bit decimal address. It contains four octets or fields separated by 'dot,' and each field is 8-bit in size. The number that each field contains should be in the range of 0-255.

Class A

Class A address uses only first higher order octet (byte) to identify the network prefix, and remaining three octets (bytes) are used to define the individual host addresses. The class A address ranges between 0.0.0.0 to 127.255.255.255. The first bit of the first octet is always set to 0 (zero), and next 7 bits determine network address, and the remaining 24 bits determine host address. So the first octet ranges from 0 to 127 (00000000 to 01111111).

Class B

Class B addresses use the initial two octets (two bytes) to identify the network prefix, and the remaining two octets (two bytes) define host addresses. The class B addresses are range between 128.0.0.0 to 191.255.255.255. The first two bits of the first higher octet is always set to 10 (one and zero bit), and next 14 bits determines the network address and remaining 16 bits determines the host address. So the first octet ranges from 128 to 191 (10000000 to 10111111).

Class C

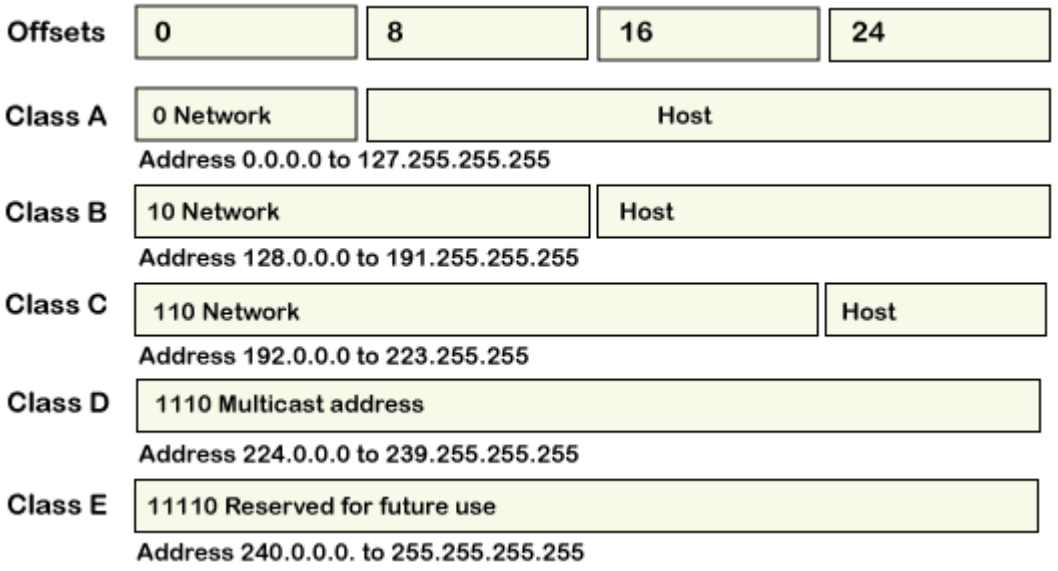
Class C addresses use the first three octets (three bytes) to identify the network prefix, and the remaining last octet (one byte) defines the host address. The class C address ranges between 192.0.0.0 to 223.255.255.255. The first three bit of the first octet is always set to 110, and next 21 bits specify network address and remaining 8 bits specify the host address. Its first octet ranges from 192 to 223 (11000000 to 11011111).

Class D

Class D IP address is reserved for multicast addresses. Its first four bits of the first octet are always set to 1110, and the remaining bits determine the host address in any IP address. The first higher octet bits are always set to 1110, and the remaining bits specify the host address. The class D address ranges between 224.0.0.0 to 239.255.255.255. In multicasting, data is not assigned to any particular host machine, so it is not require to find the host address from the IP address, and also, there is no subnet mask present in class D.

Class E

Class E IP address is reserved for experimental purposes and future use. It does not contain any subnet mask in it. The first higher octet bits are always set to 1111, and next remaining bits specify the host address. Class E address ranges between 240.0.0.0 to 255.255.255.255.



In every IP address class, all host-number bits are specified by a power of 2 that indicates the total numbers of the host's address that can create for a particular network address. Class A address can contain the maximum number of 224 (16,777,216) host numbers. Class B addresses contain the maximum number of 216 (65, 536) host numbers. And class C contains a maximum number of 28 (256) host numbers.

Subnet address of IP address, understand with an example:

Suppose a class A address is 11.65.27.1, where 11 is a network prefix (address), and 65.27.1 specifies a particular host address on the network. Consider that a network admin wants to use 23 to 6 bits to identify the subnet and the remaining 5 to 0 bits to identify the host address. It can be represented in the Subnet mask with all 1 bits from 31 to 6 and the remaining (5 to 0) with 0 bits.

Subnet Mask (binary): 11111111 11111111 11111111 11000000

IP address (binary): 00001011 01000001 00011011 00000001

Now, the subnet can be calculated by applying AND operation (1+1=1, 1+0=0, 0+1=0, 0+0=0) between complete IP address and Subnet mask. The result is:

00001011 01000001 00011011 00000000 = 11.65.27.0 subnet address

IP Address (Decimal):	11	65	27	1
IP Address (Binary):	00001011	01000001	00011011	00000001
Subnet Mask (Binary):	11111111	11111111	11111111	11000000
Subnet Address (Binary):	00001011	01000001	00011011	00000000
Subnet Address (Decimal):	11	65	27	0

CIDR:

Classless Inter-Domain Routing. which is also known as Classless addressing. In the Classful addressing the no of Hosts within a network always remains the same depending upon the class of the Network.

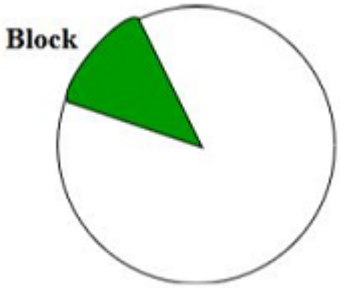
Class A network contains 224 Hosts,

Class B network contains 216 Hosts,

Class C network contains 28 Hosts

Now, let's suppose an Organization requires 214 hosts, then it must have to purchase a Class B network. In this case, 49152 Hosts will be wasted. This is the major drawback of Classful Addressing.

In order to reduce the wastage of IP addresses a new concept of Classless Inter-Domain Routing is introduced. Now a days IANA is using this technique to provide the IP addresses. Whenever any user asks for IP addresses, IANA is going to assign that many IP addresses to the User.



Representation: It is as also a 32-bit address, which includes a special number which represents the number of bits that are present in the Block Id.

$$a\,.\,b\,.\,c\,.\,d\,/\,n$$

Where, n is number of bits that are present in Block Id / Network Id.

Example:

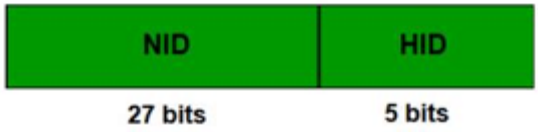
20.10.50.100/20

Rules for forming CIDR Blocks:

1. All IP addresses must be contiguous.
2. Block size must be the power of 2 (2n).
   If the size of the block is the power of 2, then it will be easy to divide the Network. Finding out the Block Id is very easy if the block size is of the power of 2.

Example:

If the Block size is 25 then, Host Id will contain 5 bits and Network will contain 32 – 5 = 27 bits.



3. First IP address of the Block must be evenly divisible by the size of the block. in simple words, the least significant part should always start with zeroes in Host Id. Since all the least significant bits of Host Id is zero, then we can use it as Block Id part.

Example:

Check whether 100.1.2.32 to 100.1.2.47 is a valid IP address block or not?

1. All the IP addresses are contiguous.
2. Total number of IP addresses in the Block = 16 = 24.
3. 1st IP address: 100.1.2.00100000

   Since, Host Id will contains last 4 bits and all the least significant 4 bits are zero. Hence, first IP address is evenly divisible by the size of the block.

All the three rules are followed by this Block. Hence, it is a valid IP address block.

Network Address Translation (NAT)

To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of a private IP address to a public IP address is required. Network Address Translation (NAT) is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts. Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on a router or firewall.

Network Address Translation (NAT) working:

Generally, the border router is configured for NAT i.e the router which has one interface in the local (inside) network and one interface in the global (outside) network. When a packet traverse outside the local (inside) network, then NAT converts that local (private) IP address to a global (public) IP address. When a packet enters the local network, the global (public) IP address is converted to a local (private) IP address.

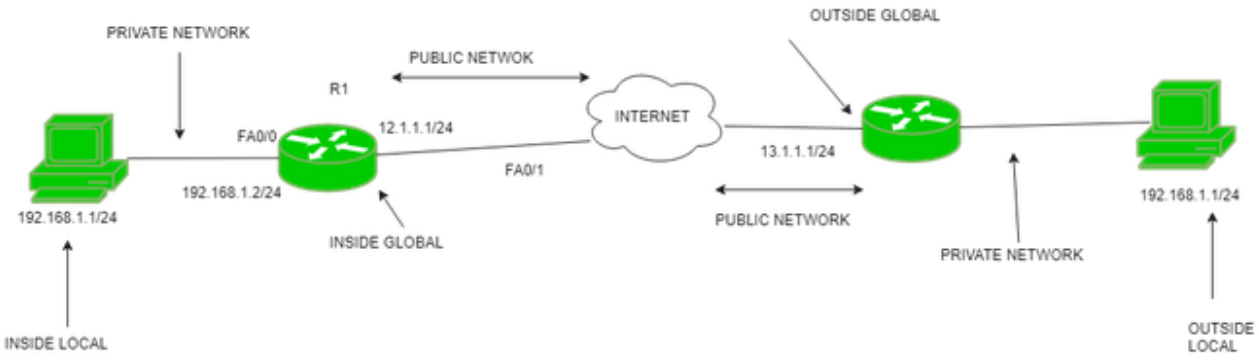
If NAT runs out of addresses, i.e., no address is left in the pool configured then the packets will be dropped and an Internet Control Message Protocol (ICMP) host unreachable packet to the destination is sent.

Why mask port numbers?

Suppose, in a network, two hosts A and B are connected. Now, both of them request for the same destination, on the same port number, say 1000, on the host side, at the same time. If NAT does only translation of IP addresses, then when their packets will arrive at the NAT, both of their IP addresses would be masked by the public IP address of the network and sent to the destination. Destination will send replies to the public IP address of the router. Thus, on receiving a reply, it will be unclear to NAT as to which reply belongs to which host (because source port numbers for both A and B are the same). Hence, to avoid such a problem, NAT masks the source port number as well and makes an entry in the NAT table.

NAT inside and outside addresses:

Inside refers to the addresses which must be translated. Outside refers to the addresses which are not in control of an organization. These are the network Addresses in which the translation of the addresses will be done.



Inside local address: An IP address that is assigned to a host on the Inside (local) network. The address is probably not an IP address assigned by the service provider i.e., these are private IP addresses. This is the inside host seen from the inside network.

Inside global address: IP address that represents one or more inside local IP addresses to the outside world. This is the inside host as seen from the outside network.

Outside local address: This is the actual IP address of the destination host in the local network after translation.

Outside global address: This is the outside host as seen from the outside network. It is the IP address of the outside destination host before translation.

Network Address Translation (NAT) Types:

There are 3 ways to configure NAT:

Static NAT: In this, a single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e one-to-one mapping between local and global addresses. This is generally used for Web hosting. These are not used in organizations as there are many devices that will need Internet access and to provide Internet access, a public IP address is needed.

Suppose, if there are 3000 devices that need access to the Internet, the organization has to buy 3000 public addresses that will be very costly.

Dynamic NAT: In this type of NAT, an unregistered IP address is translated into a registered (Public) IP address from a pool of public IP addresses. If the IP address of the pool is not free, then the packet will be dropped as only a fixed number of private IP addresses can be translated to public addresses.

Suppose, if there is a pool of 2 public IP addresses then only 2 private IP addresses can be translated at a given time. If 3rd private IP address wants to access the Internet then the packet will be dropped therefore many private IP addresses are mapped to a pool of public IP addresses. NAT is used when the number of users who want to access the Internet is fixed. This is also very costly as the organization has to buy many global IP addresses to make a pool.

Port Address Translation (PAT): This is also known as NAT overload. In this, many local (private) IP addresses can be translated to a single registered IP address. Port numbers are used to distinguish the traffic i.e., which traffic belongs to which IP address. This is most frequently used as it is cost-effective as thousands of users can be connected to the Internet by using only one real global (public) IP address.

Advantages of NAT:

- NAT conserves legally registered IP addresses.
- It provides privacy as the device's IP address, sending and receiving the traffic, will be hidden.
- Eliminates address renumbering when a network evolves.

Disadvantage of NAT:

- Translation results in switching path delays.
- Certain applications will not function while NAT is enabled.
- Complicates tunneling protocols such as IPsec.
- Also, the router being a network layer device, should not tamper with port numbers(transport layer) but it has to do so because of NAT.

Subnet (subnetwork)

What is a subnet?

A subnet, or subnetwork, is a segmented piece of a larger network. More specifically, subnets are a logical partition of an IP network into multiple, smaller network segments. The Internet Protocol (IP) is the method for sending data from one computer to another over the internet. Each computer, or host, on the internet has at least one IP address as a unique identifier.

Organizations will use a subnet to subdivide large networks into smaller, more efficient subnetworks. One goal of a subnet is to split a large network into a grouping of smaller, interconnected networks to help minimize traffic. This way, traffic doesn't have to flow through unnecessary routs, increasing network speeds.

Subnetting, the segmentation of a network address space, improves address allocation efficiency. It is described in the formal document, Request for Comments 950, and is tightly linked to IP addresses, subnet masks and Classless Inter-Domain Routing (CIDR) notation.

How do subnets work?

Each subnet allows its connected devices to communicate with each other, while routers are used to communicate between subnets. The size of a subnet depends on the connectivity requirements and the network technology employed. A point-to-point subnet allows two devices to connect, while a data center subnet might be designed to connect many more devices.

Each organization is responsible for determining the number and size of the subnets it creates, within the limits of the address space available for its use. Additionally, the details of subnet segmentation within an organization remain local to that organization.

An IP address is divided into two fields: a Network Prefix (also called the Network ID) and a Host ID. What separates the Network Prefix and the Host ID depends on whether the address is a Class A, B or C address. Figure 1 shows an IPv4 Class B address, 172.16.37.5. Its Network Prefix is 172.16.0.0, and the Host ID is 37.5.

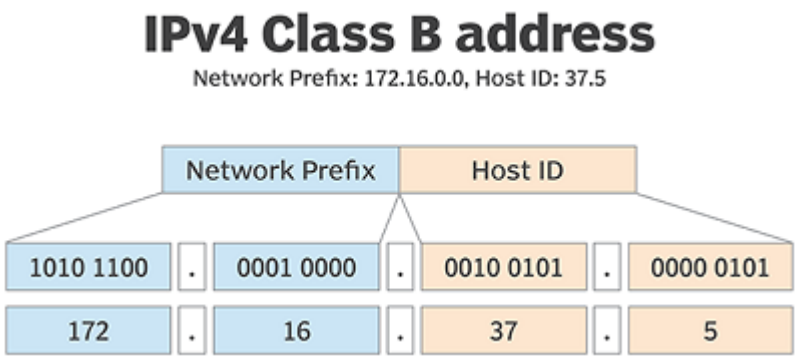


Figure 1. Class B IP address

The subnet mechanism uses a portion of the Host ID field to identify individual subnets. Figure 2, for example, shows the third group of the 172.16.0.0 network being used as a Subnet ID. A subnet mask is used to identify the part of the address that should be used as the Subnet ID. The subnet mask is applied to the full network address using a binary AND operation. AND operations operate, assuming an output is "true" only when both inputs are "true." Otherwise, the output is "false." Only when two bits are both 1. This results in the Subnet ID.

Figure 2 shows the AND of the IP address, as well as the mask producing the Subnet ID. Any remaining address bits identify the Host ID. The subnet in Figure 2 is identified as 172.16.2.0, and the Host ID is 5. In practice, network staff will typically refer to a subnet by just the Subnet ID. It would be common to hear someone say, "Subnet 2 is having a problem today," or, "There is a problem with the dot-two subnet."

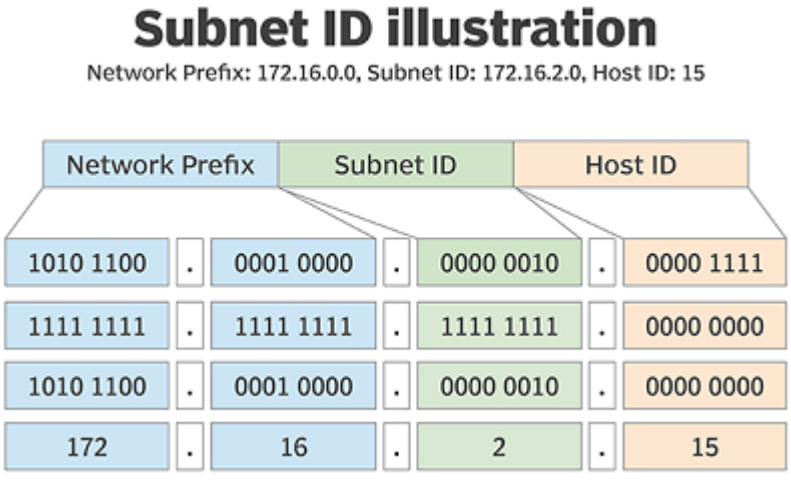


Figure 2. Subnet ID

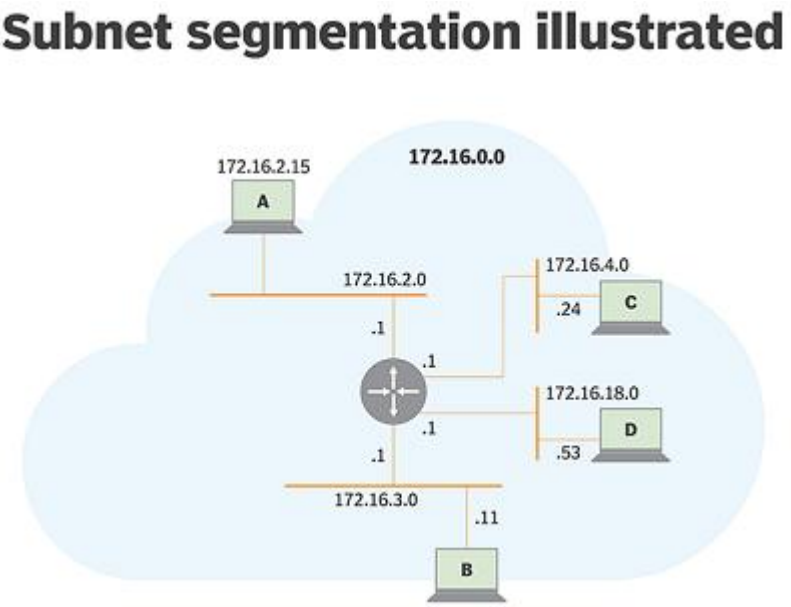
The Subnet ID is used by routers to determine the best route between subnetworks. Figure 3 shows the 172.16.0.0 network, with the third grouping as the Subnet ID. Four of the 256 possible subnets are shown connected to one router. Each subnet is identified either by its Subnet ID or the subnet address with the Host ID set to .0. The router interfaces are assigned the Host ID of .1 -- e.g., 172.16.2.1.

When the router receives a packet addressed to a host on a different subnet than the sender -- host A to host C, for example -- it knows the subnet mask and uses it to determine the Subnet ID of host C. It examines its routing table to find the interface connected to host C's subnet and forwards the packet on that interface.

Subnet segmentation

A subnet itself also may be segmented into smaller subnets, giving organizations the flexibility to create smaller subnets for things like point-to-point links or for subnetworks that support a few devices.

The example below uses an 8-bit Subnet ID. The number of bits in the subnet mask depends on the organization's requirements for subnet size and the number of subnets. Other subnet mask lengths are common. While this adds some complexity to network addressing, it significantly improves the efficiency of network address utilization.



A subnet can be delegated to a suborganization, which itself may apply the subnetting process to create additional subnets, as long as sufficient address space is available. Subnetting performed by a delegated organization is hidden from other organizations. As a result, the Subnet ID field length and where subnets are assigned can be hidden from the parent (delegating) organization, a key characteristic that allows networks to be scaled up to large sizes.

In modern routing architectures, routing protocols distribute the subnet mask with routes and provide mechanisms to summarize groups of subnets as a single routing table entry. Older routing architectures relied on the default Class A, B and C IP address classification to determine the mask to use.

CIDR notation is used to identify Network Prefix and Mask, where the subnet mask is a number that indicates the number of ones in the Mask (e.g., 172.16.2.0/24). This is also known as Variable-Length Subnet Masking (VLSM) and CIDR. Subnets and subnetting are used in both IPv4 and IPv6 networks, based on the same principles.

What are subnets used for?

- Reallocating IP addresses. Each class has a limited number of host allocations; for example, networks with more than 254 devices need a Class B allocation. If a network administrator is working with a Class B or C network and needs to allocate 150 hosts for three physical networks located in three different cities, they would need to either request more address blocks for each network -- or divide a network into subnets that enable administrators to use one block of addresses on multiple physical networks.
- Relieving network congestion. If much of an organization's traffic is meant to be shared regularly between the same cluster of computers, placing them on the same subnet can reduce network traffic. Without a subnet, all computers and servers on the network would see data packets from every other computer.
- Improving network security. Subnetting allows network administrators to reduce network-wide threats by quarantining compromised sections of the network and by making it more difficult for trespassers to move around an organization's network.

IP Version 6

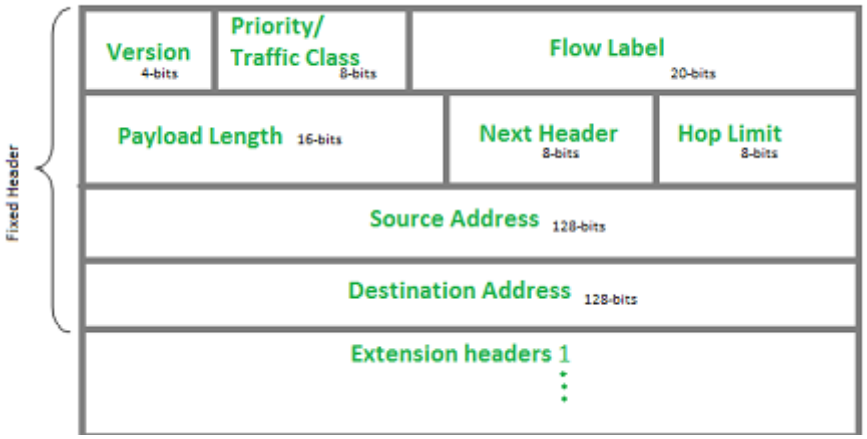
Internet Protocol version 6 (IPV 6) is the replacement for version 4 (IPV 4). The phenomenal development of the Internet has begun to push IP to its limits. It provides a large address space, and it contains a simple header as compared to IPv4.

Features of IPV6

There are various features of IPV6, which are as follows−

- **Larger address space:** An IPV6 address is 128 bits long. It is compared with the 32-bit address of IPV4. It will allow for unique IP-addresses up to 3.4 x 1038 whereas IPV4 allows up to 4.3 x 108 unique address.
- **Better Header format:** New header form has been designed to reduce overhead. It is done by moving both non-essential fields and optional fields to extension field header that are placed after the IPV6 header.
- **More Functionality:** It is designed with more options like priority of packet for control of congestion, Authentication etc.
- **Allowance for Extension:** It is designed to allow the extension of the protocol if required by new technologies.
- **Support of resource allocation:** In IPV6, the type of service fields has been removed, but a new mechanism has been added to support traffic control or flow labels like real-time audio and video.

IP version 6 Header Format:



- **Version (4-bits):** Indicates version of Internet Protocol which contains bit sequence 0110.
- **Traffic Class (8-bits):** The Traffic Class field indicates class or priority of IPv6 packet which is similar to Service Field in IPv4 packet. It helps routers to handle the traffic based on the priority of the packet. If congestion occurs on the router then packets with the least priority will be discarded.

  As of now, only 4-bits are being used (and the remaining bits are under research), in which 0 to 7 are assigned to Congestion controlled traffic and 8 to 15 are assigned to Uncontrolled traffic.

Priority assignment of Congestion controlled traffic :

Priority	Meaning
0	There is no specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended Bulk data traffic
5	Reserved
6	Interactive Traffic
7	Control Traffic

Uncontrolled data traffic is mainly used for Audio/Video data. So we give higher priority to Uncontrolled data traffic.

The source node is allowed to set the priorities but on the way, routers can change it. Therefore, the destination should not expect the same priority which was set by the source node.

- **Flow Label (20-bits):** Flow Label field is used by a source to label the packets belonging to the same flow in order to request special handling by intermediate IPv6 routers, such as non-default quality of service or real-time service. In order to distinguish the flow, an intermediate router can use the source address, a destination address, and flow label of the packets. Between a source and destination, multiple flows may exist because many processes might be running at the same time. Routers or Host that does not support the functionality of flow label field and for default

router handling, flow label field is set to 0. While setting up the flow label, the source is also supposed to specify the lifetime of the flow.

- **Payload Length (16-bits):** It is a 16-bit (unsigned integer) field, indicates the total size of the payload which tells routers about the amount of information a particular packet contains in its payload. The payload Length field includes extension headers(if any) and an upper-layer packet. In case the length of the payload is greater than 65,535 bytes (payload up to 65,535 bytes can be indicated with 16-bits), then the payload length field will be set to 0 and the jumbo payload option is used in the Hop-by-Hop options extension header.
- **Next Header (8-bits):** Next Header indicates the type of extension header(if present) immediately following the IPv6 header. Whereas In some cases it indicates the protocols contained within upper-layer packets, such as TCP, UDP.
- **Hop Limit (8-bits):** Hop Limit field is the same as TTL in IPv4 packets. It indicates the maximum number of intermediate nodes IPv6 packet is allowed to travel. Its value gets decremented by one, by each node that forwards the packet and the packet is discarded if the value decrements to 0. This is used to discard the packets that are stuck in an infinite loop because of some routing error.
- **Source Address (128-bits):** Source Address is the 128-bit IPv6 address of the original source of the packet.
- **Destination Address (128-bits):** The destination Address field indicates the IPv6 address of the final destination(in most cases). All the intermediate nodes can use this information in order to correctly route the packet.
- **Extension Headers:** In order to rectify the limitations of the IPv4 Option Field, Extension Headers are introduced in IP version 6. The extension header mechanism is a very important part of the IPv6 architecture. The next Header field of IPv6 fixed header points to the first Extension Header and this first extension header points to the second extension header and so on.

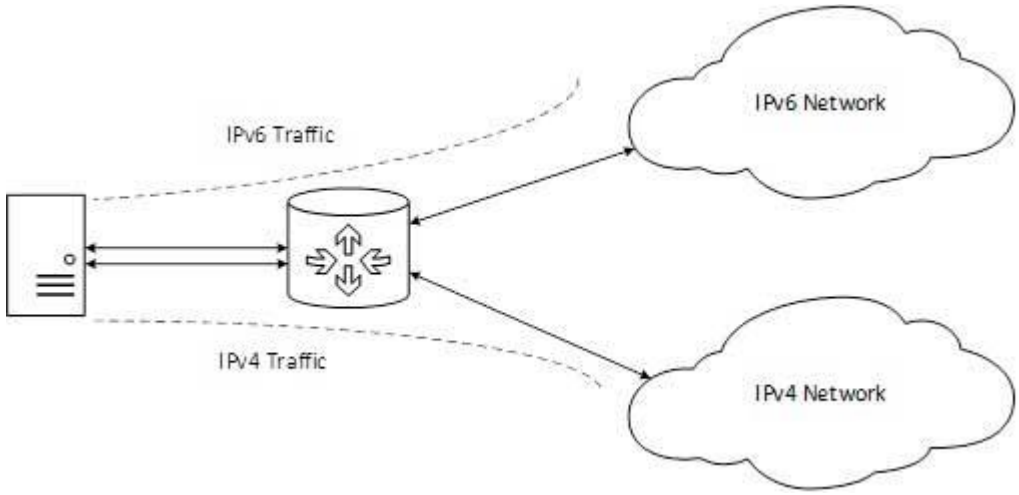
Transition From IPv4 to IPv6

Complete transition from IPv4 to IPv6 might not be possible because IPv6 is not backward compatible. This results in a situation where either a site is on IPv6 or it is not. It is unlike implementation of other new technologies where the newer one is backward compatible so the older system can still work with the newer version without any additional changes.

To overcome this short-coming, we have a few technologies that can be used to ensure slow and smooth transition from IPv4 to IPv6.

Dual Stack Routers

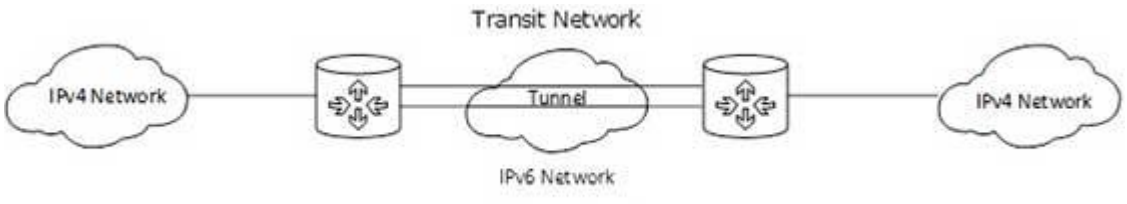
A router can be installed with both IPv4 and IPv6 addresses configured on its interfaces pointing to the network of relevant IP scheme.



In the above diagram, a server having IPv4 as well as IPv6 address configured for it can now speak with all the hosts on both the IPv4 as well as the IPv6 networks with the help of a Dual Stack Router. The Dual Stack Router, can communicate with both the networks. It provides a medium for the hosts to access a server without changing their respective IP versions.

Tunneling

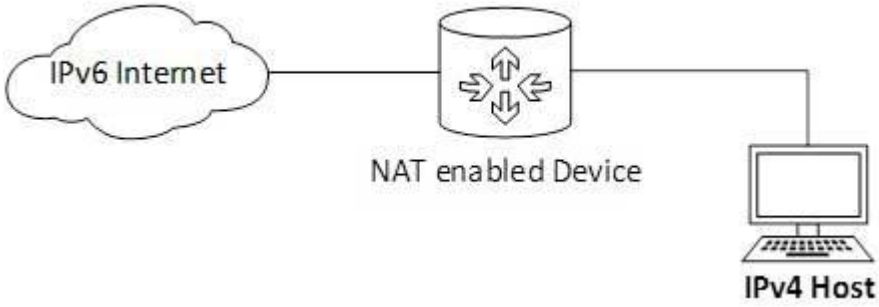
In a scenario where different IP versions exist on intermediate path or transit networks, tunneling provides a better solution where user's data can pass through a non-supported IP version.



The above diagram depicts how two remote IPv4 networks can communicate via a Tunnel, where the transit network was on IPv6. Vice versa is also possible where the transit network is on IPv6 and the remote sites that intend to communicate are on IPv4.

NAT Protocol Translation

This is another important method of transition to IPv6 by means of a NAT-PT (Network Address Translation – Protocol Translation) enabled device. With the help of a NAT-PT device, actual can take place happens between IPv4 and IPv6 packets and vice versa. See the diagram below:



A host with IPv4 address sends a request to an IPv6 enabled server on Internet that does not understand IPv4 address. In this scenario, the NAT-PT device can help them communicate. When the IPv4 host sends a request packet to the IPv6 server, the NAT-PT device/router strips down the IPv4 packet, removes IPv4 header, and adds IPv6 header and passes it through the Internet. When a response from the IPv6 server comes for the IPv4 host, the router does vice versa.

	Ipv4	**Ipv6**
Address length	IPv4 is a 32-bit address.	IPv6 is a 128-bit address.
Fields	IPv4 is a numeric address that consists of 4 fields which are separated by dot (.).	IPv6 is an alphanumeric address that consists of 8 fields, which are

		separated by colon.
Classes	IPv4 has 5 different classes of IP address that includes Class A, Class B, Class C, Class D, and Class E.	IPv6 does not contain classes of IP addresses.
Number of IP address	IPv4 has a limited number of IP addresses.	IPv6 has a large number of IP addresses.
VLSM	It supports VLSM (Virtual Length Subnet Mask). Here, VLSM means that Ipv4 converts IP addresses into a subnet of different sizes.	It does not support VLSM.
Address configuration	It supports manual and DHCP configuration.	It supports manual, DHCP, auto-configuration, and renumbering.
Address space	It generates 4 billion unique addresses	It generates 340 undecillion unique addresses.
End-to-end connection integrity	In IPv4, end-to-end connection integrity is unachievable.	In the case of IPv6, end-to-end connection integrity is achievable.
Security features	In IPv4, security depends on the application. This IP address is not developed in keeping the security feature in mind.	In IPv6, IPSEC is developed for security purposes.
Address representation	In IPv4, the IP address is represented in decimal.	In IPv6, the representation of the IP address in hexadecimal.
Fragmentation	Fragmentation is done by the senders and the forwarding routers.	Fragmentation is done by the senders only.
Packet flow identification	It does not provide any mechanism for packet flow identification.	It uses flow label field in the header for the packet flow identification.
Checksum field	The checksum field is available in IPv4.	The checksum field is not available in IPv6.

Transmission scheme	IPv4 is broadcasting.	On the other hand, IPv6 is multicasting, which provides efficient network operations.
Encryption and Authentication	It does not provide encryption and authentication.	It provides encryption and authentication.
Number of octets	It consists of 4 octets.	It consists of 8 fields, and each field contains 2 octets. Therefore, the total number of octets in IPv6 is 16.

Internet Control Message Protocol (ICMP

Internet Control Message Protocol (ICMP) works in the network layer of the OSI model and the internet layer of the TCP/IP model. It is used to send control messages to network devices and hosts. Routers and other network devices monitor the operation of the network. When an error occurs, these devices send a message using ICMP. Messages that can be sent include "destination unreachable", "time exceeded", and "echo requests".

- ICMP is a network layer protocol.
- ICMP messages are not passed directly to the data link layer. The message is first encapsulated inside the IP datagram before going to the lower layer.

Types of ICMP messages

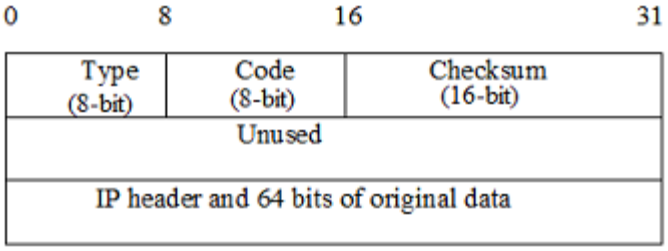
- **Information Messages:** In this message, the sender sends a query to the host or router and expects an answer. For example, A host wants to know if a router is alive or not.
- **Error-reporting message:** This message report problems that a router or a host (destination) may encounter when it processes an IP packet.
- **Query Message:** It helps a router or a network manager to get specific information from a router or another host.

Category	Type	Message
Error-Reporting Messages	3	Destination unreachable
	4	Source quench
	11	Time Exceeded
	12	Parameter Problem
	5	Redirection
Query Message	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router Solicitation or advertisement

- **Source Quench:** It requests to decrease the traffic rate of message sending from source to destination.
- **Time Exceeded:** When fragments are lost in a network the fragments hold by the router will be dropped and then ICMP will take the source IP from the discarded packet and inform the source, that datagram is discarded due to the time to live field reaches zero, by sending time exceeded message.
- **Fragmentation Required:** When a router is unable to forward a datagram because it exceeds the MTU of the next-hop network and the DF (Don't Fragment) bit is set, the router is required to return an ICMP Destination Unreachable message to the source of the datagram, with the Code indicating fragmentation is needed and DF (Don’t Fragment) set.
- **Destination Unreachable:** This error message indicates that the destination host, network, or port number that is specified in the IP packet is unreachable. This may happen due to the destination host device is down, an intermediate router is unable to find a path to forward the packet, and a firewall is configured to block connections from the source of the packet.
- **Redirect Message:** A redirect error message is used when a router needs to tell a sender that it should use a different path for a specific destination. It occurs when the router knows a shorter path to the destination.

ICMP Basic Error Message Format

A basic ICMP error message would have the following format



- **Type:** The type field identifies the type of the message.
- **Code:** The code field in ICMP describes the purpose of the message.
- **Checksum:** The checksum field is used to validate ICMP messages.

Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) is a communication protocol used to find the MAC (Media Access Control) address of a device from its IP address. This protocol is used when a device wants to communicate with another device on a Local Area Network or Ethernet.

Types of ARP

There are four types of Address Resolution Protocol, which is given below:

- Proxy ARP
- Gratuitous ARP
- Reverse ARP (RARP)
- Inverse ARP

Proxy ARP - Proxy ARP is a method through which a Layer 3 devices may respond to ARP requests for a target that is in a different network from the sender. The Proxy ARP configured router responds to the ARP and map the MAC address of the router with the target IP address and fool the sender that it is reached at its destination.

At the backend, the proxy router sends its packets to the appropriate destination because the packets contain the necessary information.

Example - If Host A wants to transmit data to Host B, which is on the different network, then Host A sends an ARP request message to receive a MAC address for Host B. The router responds to Host A with its own MAC address pretend itself as a destination. When the data is transmitted to the destination by Host A, it will send to the gateway so that it sends to Host B. This is known as proxy ARP.

Gratuitous ARP - Gratuitous ARP is an ARP request of the host that helps to identify the duplicate IP address. It is a broadcast request for the IP address of the router. If an ARP request is sent by a switch or router to get its IP address and no ARP responses are received, so all other nodes cannot use the IP address allocated to that switch or router. Yet if a router or switch sends an ARP request for its IP address and receives an ARP response, another node uses the IP address allocated to the switch or router.

There are some primary use cases of gratuitous ARP that are given below:

- The gratuitous ARP is used to update the ARP table of other devices.
- It also checks whether the host is using the original IP address or a duplicate one.

Reverse ARP (RARP) - It is a networking protocol used by the client system in a local area network (LAN) to request its IPv4 address from the ARP gateway router table. A table is created by the network administrator in the gateway-router that is used to find out the MAC address to the corresponding IP address.

When a new system is set up or any machine that has no memory to store the IP address, then the user has to find the IP address of the device. The device sends a RARP broadcast packet, including its own MAC address in the address field of both the sender and the receiver hardware. A host installed inside of the local network called the RARP-server is prepared to respond to such type of broadcast packet. The RARP server is then trying to locate a mapping table entry in the IP to MAC address. If any entry matches the item in the table, then the RARP server sends the response packet along with the IP address to the requesting computer.

Inverse ARP (InARP) - Inverse ARP is inverse of the ARP, and it is used to find the IP addresses of the nodes from the data link layer addresses. These are mainly used for the frame relays, and ATM networks, where Layer 2 virtual circuit addressing are often acquired from Layer 2 signaling. When using these virtual circuits, the relevant Layer 3 addresses are available.

ARP conversions Layer 3 addresses to Layer 2 addresses. However, its opposite address can be defined by InARP. The InARP has a similar packet format as ARP, but operational codes are different.

Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) is a network management protocol used to dynamically assign an IP address to nay device, or node, on a network so they can communicate using IP (Internet Protocol). DHCP automates and centrally manages these configurations. There is no need to manually assign IP addresses to new devices. Therefore, there is no requirement for any user configuration to connect to a DHCP based network.

DHCP can be implemented on local networks as well as large enterprise networks. DHCP is the default protocol used by the most routers and networking equipment. DHCP is also called RFC (Request for comments) 2131.

DHCP does the following:

- DHCP manages the provision of all the nodes or devices added or dropped from the network.
- DHCP maintains the unique IP address of the host using a DHCP server.
- It sends a request to the DHCP server whenever a client/node/device, which is configured to work with DHCP, connects to a network. The server acknowledges by providing an IP address to the client/node/device.

DHCP is also used to configure the proper subnet mask, default gateway and DNS server information on the node or device.

There are many versions of DCHP are available for use in IPV4 (Internet Protocol Version 4) and IPV6 (Internet Protocol Version 6).

How DHCP works

DHCP runs at the application layer of the TCP/IP protocol stack to dynamically assign IP addresses to DHCP clients/nodes and to allocate TCP/IP configuration information to the DHCP clients. Information includes subnet mask information, default gateway, IP addresses and domain name system addresses.

DHCP is based on client-server protocol in which servers manage a pool of unique IP addresses, as well as information about client configuration parameters, and assign addresses out of those address pools.

The DHCP lease process works as follows:

- First of all, a client (network device) must be connected to the internet.
- DHCP clients request an IP address. Typically, client broadcasts a query for this information.
- DHCP server responds to the client request by providing IP server address and other configuration information. This configuration information also includes time period, called a lease, for which the allocation is valid.
- When refreshing an assignment, a DHCP clients request the same parameters, but the DHCP server may assign a new IP address. This is based on the policies set by the administrator.

Components of DHCP

When working with DHCP, it is important to understand all of the components. Following are the list of components:

- **DHCP Server:** DHCP server is a networked device running the DCHP service that holds IP addresses and related configuration information. This is typically a server or a router but could be anything that acts as a host, such as an SD-WAN appliance.
- **DHCP client:** DHCP client is the endpoint that receives configuration information from a DHCP server. This can be any device like computer, laptop, IoT endpoint or anything else that requires

connectivity to the network. Most of the devices are configured to receive DHCP information by default.

- **IP address pool:** IP address pool is the range of addresses that are available to DHCP clients. IP addresses are typically handed out sequentially from lowest to the highest.
- **Subnet:** Subnet is the partitioned segments of the IP networks. Subnet is used to keep networks manageable.
- **Lease:** Lease is the length of time for which a DHCP client holds the IP address information. When a lease expires, the client has to renew it.
- **DHCP relay:** A host or router that listens for client messages being broadcast on that network and then forwards them to a configured server. The server then sends responses back to the relay agent that passes them along to the client. DHCP relay can be used to centralize DHCP servers instead of having a server on each subnet.

Benefits of DHCP

There are following benefits of DHCP:

- **Centralized administration of IP configuration:** DHCP IP configuration information can be stored in a single location and enables that administrator to centrally manage all IP address configuration information.
- **Dynamic host configuration:** DHCP automates the host configuration process and eliminates the need to manually configure individual host. When TCP/IP (Transmission control protocol/Internet protocol) is first deployed or when IP infrastructure changes are required.
- **Seamless IP host configuration:** The use of DHCP ensures that DHCP clients get accurate and timely IP configuration IP configuration parameter such as IP address, subnet mask, default gateway, IP address of DND server and so on without user intervention.
- **Flexibility and scalability:** Using DHCP gives the administrator increased flexibility, allowing the administrator to move easily change IP configuration when the infrastructure changes.