UNIT 1 NETWORK LAYER

Structure **Page No.**

1.0 INTRODUCTION

As you know, the network layer is one of the important layers of OSI model. It is responsible for different tasks of networking, but mainly its role is to determine addresses and finding a route between a source and destination node or between two intermediate devices. It establishes and maintains a logical connection between these two nodes, either a connectionless or a connection oriented communication. The basic purpose of the network layer is to provide a network to network communication capability in contrast to machine to machine common provided by data line layer. The network layer controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be determined based on static tables that are “wired into” the network and rarely changed. If too many packets are present in the subnet at the same time, they will get in one another’s way forming bottlenecks. The controlling such congestion also belongs to the network layer. The quality of service also depends on network layer issue.

In this unit, we will study the fundamental Issues of network layer. These issues are designing interface between the host and the network, the routing methods, congestion control methods and Internetworking issues. In this unit we will study how routing is done at network layer using adaptive and non adaptive algorithm. We will also discuss the Network addressing. Further, some Error reporting protocols ICMP and IGMP on network layer will be discussed.

1.1 OBJECTIVES

After going through this unit, you should be able to:

- Know the basic issues of network layer
- Understand the different switching methods used at network layer
- Know the routing mechanisms

- Understand the congestion control methods
- Differentiate between adaptive and non adaptive algorithm
- Know process of Error reporting protocols at network layer

1.2 SWITCHING

As you have studied earlier in block 1, unit 3, that Switching is used to determine the path to be used for forwarding the information to the receiver. You also know that the Switching methods are mainly divided into Circuit, Message and Packet switching. In this section, we will explore the other switching mechanism like virtual circuit and datagram.

Virtual circuit is a connection oriented communication service that is delivered by means of packet mode communication. After a connection or virtual circuit is established between two nodes or application processes, a bit stream may be delivered between the nodes. It is similar to the circuit switching only in virtual circuit permanent/physical connection are not established. If router fails all virtual circuits that pass through, the failed router are terminated.

Datagram is opposite of Virtual circuit, it is connection less service. A datagram or packet needs to be self-contained without any dependency on earlier data-transfer because there is no connection of fixed duration between the two communicating nodes as shown in figure 1. Following Table 1 show the difference between circuit switching, virtual circuit and datagram:

Table 1: Difference Between Circuit Switching, Virtual Circuit and Datagram

S.No.	**Circuit Switching**	**Virtual Circuit**	**Datagram**
1.	Dedicated path between sender and receiver	Non-dedicated between sender and receiver.	Non-dedicated.
2.	Connection oriented Highly Reliable	Connection oriented	Connection less
3.	Data transfer in continuous form.	Data transfer in packets	Data transfer in packets.
4.	Bandwidth is fixed.	Not fined dynamic.	Dynamic
5.	E.g. telephone services	Subnet	Internet
6.	Data transfer in voice form.	Data transfer usually in text from.	Text form
7.	Call setup delay is maximum.	Delay Moderate	Call set up negligible.
8.	Transmission delay minimum	Moderate	Maximum transmission delay
9.	Unused bandwidth is just wasted	Moderate	No wasted.

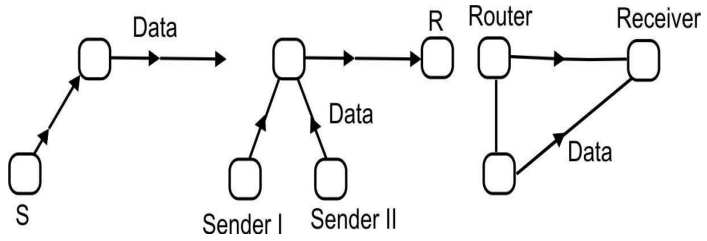


Figure 1: Circuit switching, Virtual circuit and Datagram (left to right)

1.3 ROUTING ALGORITHM

The main function of the network layer is routing packets from the source machine to the destination machine. So, the algorithm that choose the routes and the data structure that they use are a major area of network layer design.

It is that part of the network layer responsible for deciding which output line an incoming packet should be transmitted on.

Desired Properties of a Routing Algorithm

1. **Correction:** The routing should be done properly and correctly so that the packets may reach their proper destination.

2. **Simplicity:** The routing should be done in a simple manner so that the over head is as low as possible.

3. **Robustness:** Once a major network becomes operative, it may be expected to run continuously for years without any failure.

4. **Stability:** The routing algorithm should be stable under all possible circumstances.

5. **Fairness:** Every node connected to the network gets a fair change of transmitting their packets. This is generally done on a first come first serve basis.

6. **Optimality:** The routing algorithms should be optimal in terms of throughput and minimizing mean packet delays. Here there is a trade off and one has to choose depending on his suitability.

1.3.1 Classification of Routing Algorithms

Routing algorithm may be classified as follows:

1. Adaptive Algorithm

2. Non-adaptive algorithms

Adaptive algorithms use such dynamic information as current topology, load delay etc to select routes.

Non adaptive algorithms, routes never changes once initial routes have been selected. Also, called static routing.

Adaptive Routing Algorithm (Dynamic Routing)

It changes their routing decision to reflect changes in the topology and in traffic as well. These get their routing information from adjacent routers or from all routers. Routing decision may be changed when network topology and/or traffic load changes. The optimization parameters are the distance, number of hops and estimated transit time.

Adaptive routing algorithms can be further classified as follows:

1. **Isolated:** Each router makes its routing decisions using only the local information it has on hand. Specifically, routers do not even exchange information with their neighbors.

2. **Centralized:** A centralized node makes all routing decision specifically the centralized node has access to global information.

3. **Distributed:** Algorithm that uses a combination of local and global information.

Isolated: In this method, the node decides the routing without seeking information from other node. The disadvantage is that the packet may be sent through a congested route resulting in a delay.
Some of the examples of this type of algorithm for routing are :

- **Hot Potato Routing:** Form of routing in which the nodes of a network have no buffer to store packets in before they are moved on to their final predetermined destination.
- In normal routing situation, when multiple packets contend for a single outgoing channel, packets that are not buffered are dropped to avoid congestion.
- Backward Learning: In this method the routing tables at each node gets modified by information from the incoming packets. Backward learning routing algorithm used for routing traffic that makes decisions by assume that a can optimally reach B through C.

Centralized Routing

Advantage: Only one node is required to keep the information.

Disadvantage: If the central node goes down the entire network is down, i.e. single point of failure.

Distributed: It receives information from its neighboring nodes and then takes the decision about which way to send the packet.

Disadvantages: If in between the interval it receives information and sends the packet, something changes then packet may be delayed.

Optimality Principle

Optimality principle is a general statement about optimal routes regardless of network topology or traffic.

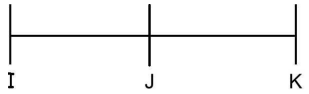


Figure 2: An example of Optimality Principle

With reference to Figure 2 above, Optimality principle states that if router I 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.

To prove the above statement we can say, if there was a better way from J to K, then you could use that with the path from I to J for a better path from I to K, so your starting point (the path from I to K was optimal) is contradicted.

1.3.2 Non-Adaptive Routing Algorithm (Static Routing)

These algorithms do not take their routing decisions on measurements and estimates of the current traffic and topology. Instead the route to be taken from one node to the other is computed in advance. This is also known as static routing.

1. **Shortest Path Routing:** According to this algorithm build a graph of the subnet, with each node of graph representing a router and each arc of the graph representing communication line. To choose a route between a pair of routers, just finds the shortest path between them on the graph.

   Two ways of measuring distance

   a) Distance in terms of link delay.

   b) Measuring path length in number of hops.

      In the most general case, the labels on the arc could be computed as a function of the bandwidth, average traffic communication cost, mean queue length measured delay, and other factors.

   c) **Flooding:** According to this algorithm every incoming packet is sent out on every outgoing line except the cone it arrived on.

      Flooding generates vast number of duplicate packets 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, which is decremented at each hop with the packet being discarded when the counter reaches zero. The hop counter should be initialized to the length of path from source to destination. If the sender does not know how long the path is, it can be initialize the counter to the worst case, namely, the full diameter of the subnet. An alternative technique for damming the flood is to keep track of which packets have been flooded, to avoid sending them out a second time.

   d) **Selective Flooding**

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

      Though flooding is not practical in most application, but (yes) is does have some uses.

      i) **In military applications:** Tremendous robustness of flooding is highly desirable.

      ii) **In distributed data base application:** It is sometimes necessary to update all the data bases concurrently, in which case flooding can be useful.

Disadvantages:

1. Duplicacy

2. Infinite looping.

3. **Flow-Based Routing:** This algorithm considers two strategies in account to decide the route.

   a) Topology.
   b) Load for routing
   Previous static algorithm only considers topology in account not the load for routing. The basic idea behind the analysis is that for a given line, if capacity and average flow is known, it is possible to compute the mean packet delay on that line from queuing theory. The routing problem then reduces to finding the routing algorithm that produces the minimum average delay for the subnet.

   For this technique, certain information must be known in advance.

   c) Topology

   d) Traffic Matrix Fij

   e) Line capacity matrix Cij

   Now $T = \frac{1}{\mu_c - \lambda}$

   Where T = Mean delay

   $\frac{1}{\mu}$ = Mean packet size

   □ = Mean flow in packet/sec. (No of arrival frame on particular line).

   C = Capacity

   Two common methods are used to calculate the shortest path between two routers.

1. Distance vector routing (Bellman ford routing algorithm and the Ford Fulkerson algorithm).
2. Link state routing (based on Dikastra's algorithm).

1.3.3 Dynamic Routing Algorithm (Adaptive)

Routing algorithms can be classified based on inter-domain and intra-domain as shown in figure 3 given below.

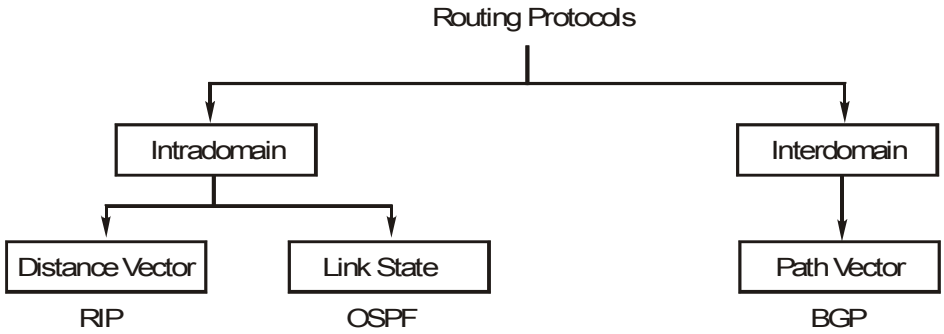


Figure 3: Classification of routing algorithms

RIP □ Routing information protocol

OSPF □ Open shortest path first

BGP □ Border gateway protocol.

1. **Distance vector routing**

Distance Vector Routing: According to this algorithm, each router maintains a table (vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors. This algorithm is also called **Bellman-Ford** or the **Ford-Fulkerson Algorithm**.

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for each router in the subnet. This entry contains two parts

a) The preferred outgoing line to use for that destination.

b) An estimate of the time as distance to that destination.

The router is assumed to know the distance to each of its neighbors. For example, consider a subnet as given below in figure 4.

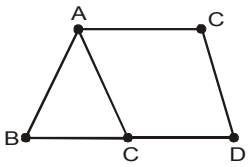


Figure 4: Subnet Diagram

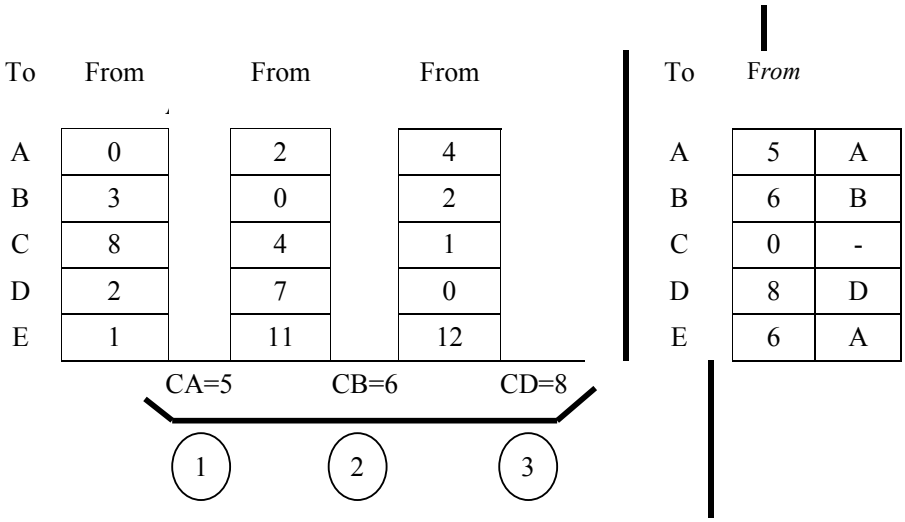


Figure 5: An example of Distance Vector Routing

Part (a) shows a subnet.

Part (b) the first 3 column shows the delay received from neighbor of router 'C' is A and B and D.

For example : as shown in the figure 5, 'A' claims to have 3 msec delay to B 8 msec delay to 'C' and so on. Similarly 'B' claims to have 2 msec delay to A, 4 msec delay to 'C' and so on.

'C' has estimated his delay to neighbour A, B, D as 5, 6, 8 respectively (CA = 5, CB = 6, CD = 8).

Now (4) column shows how router 'C' decides his new route to router 'E'. There are three ways

a) If 'C' follows line 'A' then delay is CE = CA ® AE = 5 + 1 = 6 msec.

c) If ‘C’ follows line ‘B’ then delay is CE = CB ® BE = 6 + 11 = 17 msec.

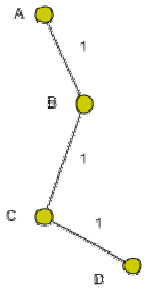
d) If ‘C’ follows line ‘D’ then delay is CD ® DE ® 8 + 12 = 20 msec.

Min delay time is via neighbor route ‘A’ so from C to E line is chosen ‘A’ in column (4)

The same calculations is performed for all destination, with the new routing table as (4)

Problem of Distance Vector Routing

1.**Count to Infinity Problem**-Distance vector routing has a serious drawback in its receptivity. In particular, it reacts rapidly to good news, but slowly to bad news. Following illustration shows an imagined network and denotes the distances from router A to every other router in the figure 6. Until now everything works fine.



B	C	D
1	2	3

Figure 6: Count to Infinity Problem

The illustration shows that link (A, B) is broken. Router B observed it, but in its routing table he sees, that router C has a route to A with 2 hops.

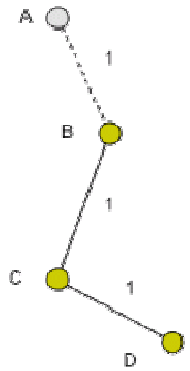
The problem is, that router B doesn't know that C has router B as successor in his routing table on the route to A.

That occurs followed count-to-infinity problem. B actualizes his routing table and takes the route to A over router C.

In the next figure 7; we can see the new distances to A. In C's routing the route to A contains router B as next hop router, so if B has increase his costs to A, C is forced to do so. Router C increases his cost to A about B + 1 = 4.

Now we see the consequence of the distributed Bellman-Ford protocol: Because router B takes the path over C to A, it updates its routing table and so on!
At the end, this problem is going to immobilize the whole network.

1.



B	C	D
3	2	3

B	C	D
3	4	3

B	C	D
5	4	5

Figure 7: Count to Infinity Problem illustration

2. **Hierarchical Routing:** As networks grow in size, the router routing tables grow proportionally. Not only the router memory consumed but also the more CPU time is needed to scan them and more bandwidth is needed.

The problem can be solved to some extent by using Hierarchical routing. In this routers are divided into regions as depicted in figure 8, with each router knowing all the details how to route packets to destination within it own region, but nothing about the internal structure of other regions.

For huge network, 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.

For example : When different networks are connected together, it is natural to regard each one as a separate region in order to free the routers in one network from having to know the topological structure of other ones.

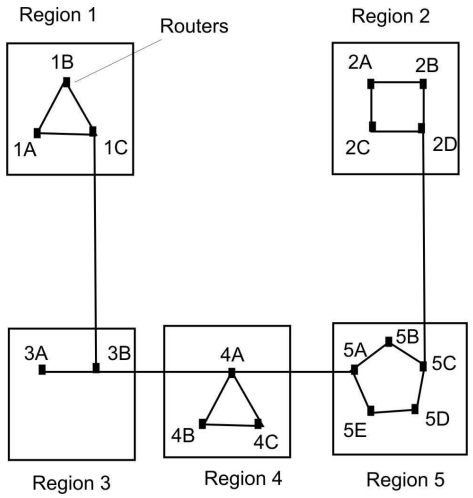


Figure 8: Hierarchical Routing

Full Table for 1A (Dest=Destination)

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	2
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	16	4

The full routing table as shown above for 1A has 17 entries but when routing is done hierarchically there are only 7 entries (e entries for local routers 4 entries for regions which are considered as single router. All the traffic for region 2 goes by 1B-2A line but rest of traffic goes by 1C-3B line.

The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. The collection of best paths will then form the node's routing table.

Disadvantages:

1. There is a penalty to be paid in the form of 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.

2. If single network become very large then multilevel hierarchy can be used.

   The presence of congestion means that the load is (temporarily) greater than the resources can handle.

3. Link state Routing

   Link State routing protocols (an adaptive routing algorithm) do not view networks in terms of adjacent routers and hop counts, but they build a comprehensive view of the overall network which fully describes the all possible routes along with their costs. Using the SPF (Shortest Path First) algorithm, the router creates a "topological database" which is a hierarchy reflecting the network routers it knows about. It then puts it's self on the top of this hierarchy, and has a complete picture from it's own perspective.

The complete working of algorithm can be divided into Five Steps:

1. Discover your neighbors and learn their addresses.

   In this process send “Hello”, packet on each point-to-point line. After receiving the hello packet Destination, node replies with its address.

2. Measure the cost (delay) to each neighbor.

   Send an “ECHO” packet over the line. Destination is required to respond to “ECHO” packet immediately. Measure the time required for this operation.

3. Construct a packet containing all this information

   The information tables are creating having all details of neighboring nodes.

4. Send this packet to all other routers.

   Use selective flooding. Sequence numbers prevent duplicate packets from being propagated. Lower sequence numbers are rejected as obsolete

5. Compute the shortest path to every other router.

   Dijkstra’s Shortest Path algorithm is used to determine the shortest path to each destination.

   When a router using a Link State protocol, such an OSPF (Open Shortest Path First) knows about a change on the network, it will broadcast this change instantly, there for flooding the network with this information. The information routers require to build their databases is provided in the form of Link State advertisement packets (LSAP). Routers do not advertise their entire routing tables; instead each router advertises only its information regarding immediately adjacent routers.

1.3.4 Comparison Link State Versus Distance Vector Routing

- Link state has big memory requirements
- In link state shortest path computations require many CPU circles
- Link state, If network is stable little bandwidth is used; react quickly to topology changes
- In link state announcements cannot be “filtered”. All items in the database must be sent to neighbors
- In link state all neighbors must be trusted
- In link state authentication mechanisms can be used to avoid undesired adjacencies
- In link state no split horizon techniques are possible

  Even though Link State protocols work more efficiently, problem can arise. Usually problems occur cause of changes in the network topology (links go up-down), and all routers don't get updated immediately cause they might be on different line speeds, there for, routers connected via a fast link will receive these changes faster than the others on a slower link.

Different techniques have been developed to deal with these problem and these are:

1. Dampen update frequency
2. Target link-state updates to multicast
3. Use link-state area hierarchy for topology
4. Exchange route summaries at area borders
5. Use Time-stamps Update numbering & counters
6. Manage partitions using a area hierarchy

1.4 CONGESTION CONTROL

Congestion: When too many packets are in a subnet or a part of subnet, performance degrades as depicted in figure 9. This situation is called congestion.

Factors Causing the Congestion

1. Many input lines demanding the same output lines.
2. Slow receiver fast sender.
3. Low bandwidth lines can also cause congestion.
4. Congestion itself (duplicacy).
5. Traffic is bursty.

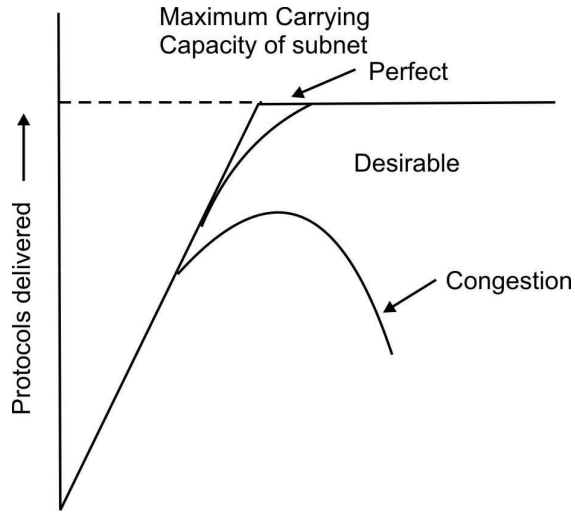


Figure 9: capacity of subnet and congestion

Congestion control principles are divided into two categories:

a) Open loop: In open loop solution the good designs are being developed to solve the problem so that congestion does not occur at first place once the system is setup and running, no mid pores connection is made. In open loop control, tools are included to decide when to accept new traffic, when to discard packets and which ones. And making scheduling decisions at various points in the network. The decisions are offline decision is not based on current state of network close loop solution.

   The concept of feedback loop is used in closed loop solution. This approach has three parts, when apply to the congestion control.

   i) Monitor the system to detect when and where congestion occurs.

   ii) Pass the information to places where the action can be taken.

   iii) Adjust system operation to correct the problem.

1.4.1 Algorithm for Congestion Control

Two major criterion of congestion control are

1. To decrease load

2. To increase capacity

Traffic Shaping (Congestion Control Policy in ATM)

One of the main causes of congestion is that traffic is often bursty. If hosts could be made to transmit a uniform rate, congestion would be less common. Another open loop method to help manage congestion is forcing the packet to be transmitted at a more predictable rate. This approach to congestion management is widely used in ATM networks and is called traffic shaping.

Traffic shaping is about regulating the average rate (and burstiness) of data transmission.

Leaky Bucket Algorithm

Imagine a bucket with a small hole in the bottom as depicted in figure 10. No matter at what rate enters the bucket, the outflow is at a constant rate, when there is any

water is bucket and zero when the bucket is empty. Also once the bucket is full, any additional water entering it spills over the sides and is lost.

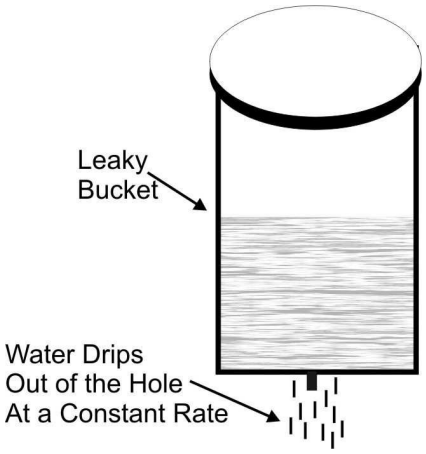


Figure 10: Normal Leaky bucket

The same idea can be applied to packets conceptually; each host is connected to the network by an interface containing a leaky bucket, i.e. a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. This arrangement can be built into the hardware interface. It was first proposed by turner and is called leaky bucket algorithm as given figure 11.

Leaky bucket algorithm can be understood as "The leaky bucket consists of finite queue when a packet arrives, if there is room on the queue it is appended to the queue, otherwise it is discarded. At every clock tick, one packet is transmitted".

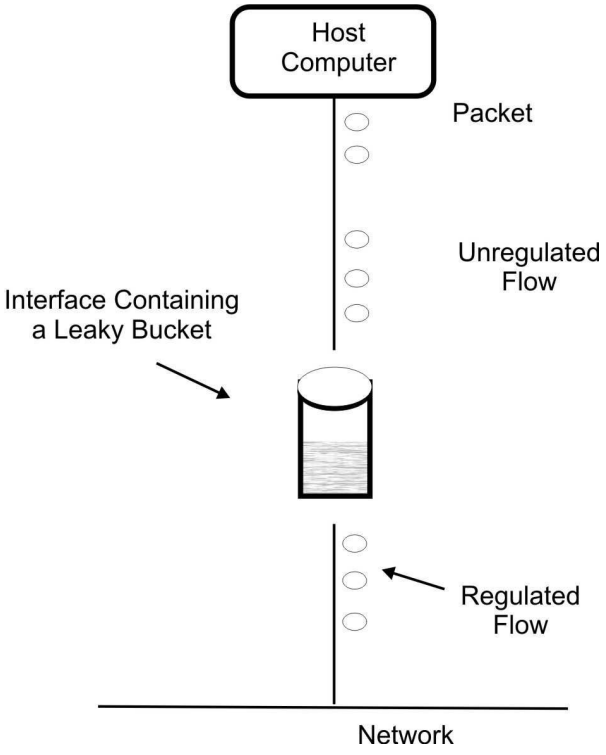


Figure 11: Leaky bucket algorithm

Advantage

This algorithm smoothens the bursts and greatly reduces the chances of congestion.

Disadvantages

1. When the queue is full, packets are discarded.

2. Sometimes it is necessary to speed up the output which is not possible in leaky bucket algorithm.

Token Bucket Algorithm: The leaky bucket algorithm has a rigid output pattern at the average rate, no matter how bursty the traffic is.

In many applications, it is better to allow the output to speed up somewhat when large bursts arrive so a more flexible algorithm is needed, preferably one that never losses data one such algorithm is token bus algorithm.

In token bucket algorithm, the leaky bucket holds 'tokens' generated by a clock at the rate of one token every DT sec.

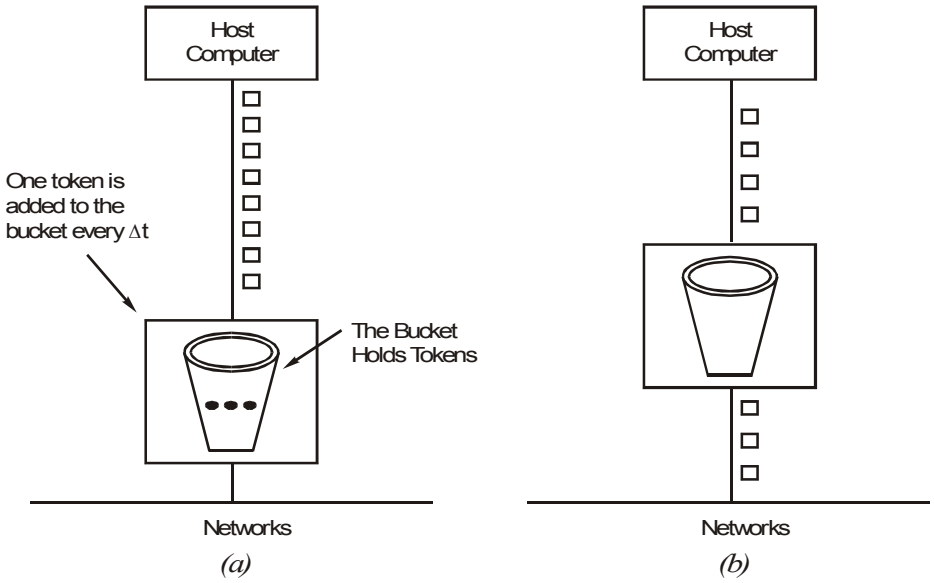


Figure 12: Token Bucket Algorithm

In above figure 12, we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. Three of fine packets have gone through, but other two are waiting for tokens to be generated.

The token bucket algorithm provides a different kind of traffic shaping than the leaky bucket algorithm. Bust of up to n. packets can be sent at once, allowing some burstiness in the output stream and giving faster response to sodden bursts of input.

A token bucket algorithm throws away tokens when the bucket fills up but never discards packets.

Table 1: Token Bucket V/S Leaky Bucket

S.No.	Token Bucket	Leaky Bucket
1.	The algorithm shaping is quite different.	In this there is a trade off between memory and bandwidth and packet life time.
2.	It allows saving up to maximum size of 'n' i.e. burst can be send of size 'n' at once.	It has constant traffic depending on the Leakage.
3.	Token bucket discarded token when bucket fills up.	This discards the packets when bucket fills

☞ Check Your Progress 1

1. The shortest path in routing can refer to……..

   a) The least expensive path
   b) The least distant path
   c) The path with the smallest number of hops
   d) Any or a combination of above

   ……

   ……

2. In distance vector routing, each router receives vector from

   a) Every router in the network
   b) Every router less than two units away
   c) A table stored by the software
   d) Its neighbor only

   ……

   ……

3. In link state routing, flooding allows changes to be recorded by

   a) All router
   b) Neighbor router only
   c) Some routers
   d) All networks

   ……

   ……

4. In which type of switching, do all the datagrams of a message follow the same channel of a path

   a) Circuit switching
   b) Datagram packet switching
   c) Virtual circuit packet switching
   d) Message switching

   ……

   ……

5. Which type of switching uses the entire capacity of a dedicated link

   a) Circuit switching
   b) Datagram packet switching
   c) Virtual circuit packet switching
   d) Message switching

   ……

   ……

1.5 NETWORK ADDRESSING

IP address versions

IP became the official protocol for the internet in 1983.As the internet has evolved, so has the IP. There have been six versions since its inception. Three versions are main.

1. Version 4(IPv4)
2. Version 5(Ipv5)
3. Version 6(Ipv6)

IP addressing

The identifier used in the IP layer of the TCP/IP protocol suite to identify each device connected to the Internet is called the Internet address or IP address.

OR

An IP address (Ipv4) is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet.

There are **three** common ways in which IP addresses can be represented.

1. There is the binary notation which uses the base two number system to represent numbers.
2. There is the decimal notation which uses the base ten number system to represent numbers.
3. There is the hexadecimal notation which uses the base sixteen number system to represent numbers.

Do you know?

An IP address is a 32-bit(4-bytes) address.

Example 1

Assume, IGNOU's IP address is 142.190.23.180. This IP address consists of four bytes. The first byte has the value of 142. The second byte has the value of 190. The third byte has the value of 23, and the fourth byte has the value of 18.

Do you know?

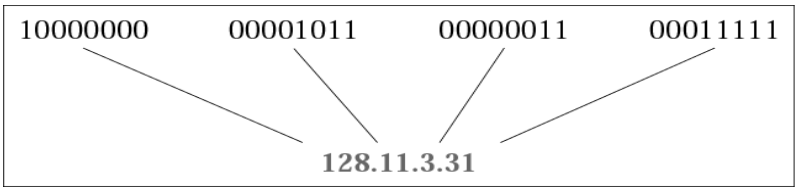
IP addresses are unique.

IP addresses are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address.

Do you know?

The address space of IPv4 is 2^{32} or 4,294,967,296.

Notations of IP addresses



Example 1

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

a) 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 11100111 11011011 10001011 01101111
d. 11111001 10011011 11111011 00001111

Solution

We replace each group of 8 bits with its equivalent decimal number and add dots for separation:

a) 129.11.11.239
b) 193.131.27.255
c) 231.219.139.111
d) 249.155.251.15

Example 2

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

a) 111.56.45.78
b) 221.34.7.82
c) 241.8.56.12
d) 75.45.34.78

Solution

We replace each decimal number with its binary equivalent:

a) 01101111 00111000 00101101 01001110
b) 11011101 00100010 00000111 01010010
c) 11110001 00001000 00111000 00001100
d) 01001011 00101101 00100010 01001110

Example 3

Find the error, if any, in the following IP addresses:

a) 111.56.045.78
b) 221.34.7.8.20
c) 75.45.301.14
d) 11100010.23.14.67

Solution

a) There are no leading zeroes in dotted-decimal notation (045).
b) We may not have more than four numbers in an IP address.
c) In dotted-decimal notation, each number is less than or equal to 255; 301 is outside this range.
d) A mixture of binary notation and dotted-decimal notation is not allowed.

Example 4

Change the following IP addresses from binary notation to hexadecimal notation.

a) 10000001 00001011 00001011 11101111
b) 11000001 10000011 00011011 11111111

Solution

We replace each group of 4 bits with its hexadecimal equivalent (see Appendix B). Note that hexadecimal notation normally has no added spaces or dots; however, 0X (or 0x) is added at the beginning or the subscript 16 at the end to show that the number is in hexadecimal.

a) 0X810B0BEF or 810B0BEF16

b) 0XC1831BFF or C1831BFF16

Classless Inter-Domain Routing allocates address space to Internet service providers and end users on any address bit boundary, instead of on 8-bit segments. CIDR notation is a syntax of specifying IP addresses and their associated routing prefix. It appends to the address a slash character and the decimal number of leading bits of the routing prefix, e.g., 192.0.2.0/24 for IPv4

1.5.1 Classful Addressing

IP addresses, when started a few decades ago, used the concept of classes. This architecture is called classful addressing. In the mid-1990s, a new architecture, called classless addressing, was introduced and will eventually supersede the original architecture. However, part of the Internet is still using classful addressing, but the migration is very fast.

Occupation of the address space

Class	Number of Addresses	Percentage
A	2^{31} = 2,147,483,648	50%
B	2^{30} = 1,073, 741, 824	25%
C	2^{29} = 536,870,912	12.5%
D	2^{28} = 268,435,456	6.25%
E	2^{28} = 268,435,456	6.25%

Table 13: Addresses per Class

When IP addresses were first started, they used the concept of classes. The range of IP addresses were divided into five classes: As shown in the figure 13 and 14, A, B, C, D, and E. Class A used up 50% of the address space, class B used up 25%, class C used up 12.5%, class D used up 6.25%, and class E also used up 6.25%.

Figure 14: Address Spaces of IPv4 classes

The way you recognize which class an IP address belongs to is by analyzing the first byte. If the number in the first byte is between 0-127, then the IP address is in the Class A range as shown in figure 15. If it is between 128-191 it is in Class B. If it is between 192-223 it is in the Class C range. If it is between 224-239 it is in the Class D range, and if it is between 240-255, then it belongs to Class E.

Class in decimal notations

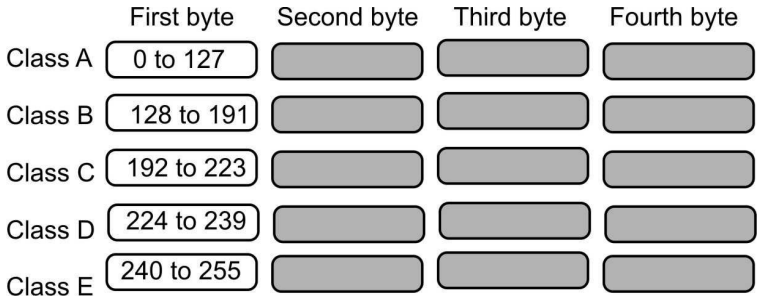


Figure 15: Classes of IPv4in decimal notation

Example

If IGNOU's IP address is 140.192.23.180. Looking at this address we can see that the first byte is 140. Since 140 is between the numbers 128-191, we know that it is in the Class B range.

Class in binary notation

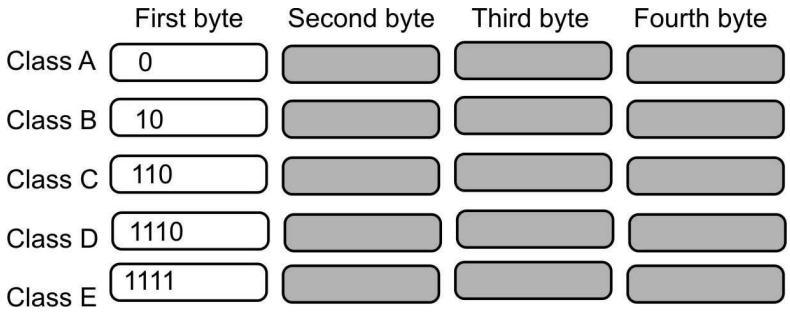


Figure 16: Classes of IPv4in binary notation

According to the figure 16, we can device a mechanism as given figure 17 for finding the address class in binary notation like:

If first left most bit is 0 then it is class A
If first bit is 1 and second bit is 0 then it is class B
If first two bits are 1 and third bit is 0 then it is class C
If first three bits are 1 and fourth bit is 0 then it is class D
If all the four bits are 1 then it is class E

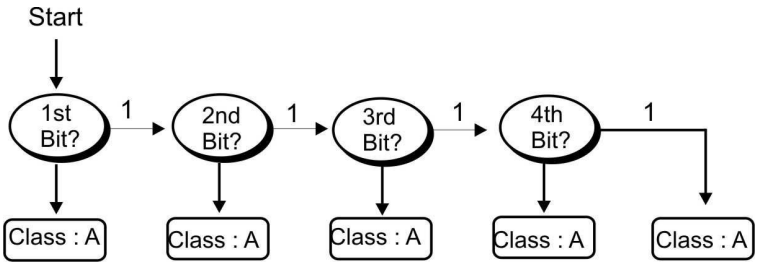


Figure 17: Finding the address class in binary notation

Example 5

How can we prove that we have 2,147,483,648 addresses in class A?

Solution

In class A, only 1 bit defines the class. The remaining 31 bits are available for the address. With 31 bits, we can have 2^{31} or 2,147,483,648 addresses.

Example 6

Find the class of each address:

a) 00000001 00001011 00001011 11101111
b) 11000001 10000011 00011011 11111111
c) 10100111 11011011 10001011 01101111
d) 11110011 10011011 11111011 00001111

Solution

a) The first bit is 0. This is a class A address.
b) The first 2 bits are 1; the third bit is 0. This is a class C address.
c) The first bit is 1; the second bit is 0. This is a class B address.
d) The first 4 bits are 1s. This is a class E address

Example 7

Find the class of each address:

a) 227.12.14.87
b) 193.14.56.22
c) 14.23.120.8
d) 252.5.15.111
e) 134.11.78.56

Solution

a) The first byte is 227 (between 224 and 239); the class is D.
b) The first byte is 193 (between 192 and 223); the class is C.
c) The first byte is 14 (between 0 and 127); the class is A.
d) The first byte is 252 (between 240 and 255); the class is E.
e) The first byte is 134 (between 128 and 191); the class is B.

Example 8

In Example 5 we showed that class A has 2^{31} (2,147,483,648) addresses. How can we prove this same fact using dotted-decimal notation?

Solution

The addresses in class A range from 0.0.0.0 to 127.255.255.255. We need to show that the difference between these two numbers is 2,147,483,648. This is a good exercise because it shows us how to define the range of addresses between two addresses. We notice that we are dealing with base 256 numbers here. Each byte in the notation has a weight. The weights are as follows

$$256^3, 256^2, 256^1, 256^0$$

Now to find the integer value of each number, we multiply each byte by its weight:

Last address: $127 \times 256^3 + 255 \times 256^2 +$

$$255 \times 256^1 + 255 \times 256^0 = 2{,}147{,}483{,}647$$

First address: = 0
If we subtract the first from the last and add 1 to the result (remember we always add 1 to get the range), we get 2,147,483,648 or 2^{31}.

1.5.2 NetID and HostID

An IP address is divided into a network ID (netid) and a host ID (hostid) as depicted in figure 18. The lengths of the netid vary depending on the class the IP address belongs to. In class A the netid occupies the first byte and the hostid occupies the remaining three bytes. In class B the netid occupies the first two bytes and the hostid occupies the remaining two bytes. In class C the first three bytes define the netid and the last remaining byte defines the hostid. Class D and E are not divided into netid and hostid. The following figure 18 shows how the netid and hostid are divided in Classes A, B, and C.

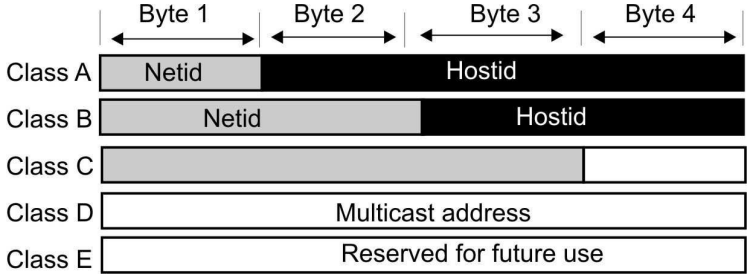


Figure 18: Network ID and a host ID of IPv4 classes

Do you know?

Class D addresses are used for multicasting; there is only one block in this class and Class E addresses are reserved for future purposes; most of the block is wasted.

In classful addressing the netid and hostid are easily distinguishable by looking at the IP address. First you have to determine which class the IP address belongs to and from there you can tell which part is the netid and which part is the hostid. If it is in Class A, then the first byte represents the netid and the last three represent the hostid, and so on.

Disadvantages of class full addressing

It wastes a lot of IP addresses and since the Internet keeps growing larger, we can't afford to throw away IP addresses.

That is why a new addressing scheme was devised. It is called classless addressing because it doesn't use the classes which were used in classful addressing.

Do you know?

Millions of class A and class B addresses and are wasted in class full addressing

Do you know?

The number of addresses in class C is smaller than the needs of most organizations.

Example 9

Given the network address 17.0.0.0, find the class, the block, and the range of the addresses.

Solution

The class is A because the first byte is between 0 and 127. The block has a netid of 17. The addresses range from 17.0.0.0 to 17.255.255.255.

Example 10

Given the network address 132.21.0.0, find the class, the block, and the range of the addresses.

Solution

The class is B because the first byte is between 128 and 191. The block has a netid of 132.21. The addresses range from 132.21.0.0 to 132.21.255.255.

Example 11

Given the network address 220.34.76.0, find the class, the block, and the range of the addresses.

Solution

class is C because the first byte is between 192 and 223. The block has a netid of 220.34.76. The addresses range from 220.34.76.0 to 220.34.76.255.

☞ Check Your Progress 2

1. Which IP address class has few hosts per network

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

   ……

   ……

2. Which of the following is true about IP addresses

   a) It is divided into exactly two classes
   b) It contains a fixed length host-id
   c) It was established as a user friendly interface
   d) It is 32 bits long

   ……

   ……

3. Which of the following is class C host address

   a) 230.0.0.0
   b) 130.4.4.6
   c) 200.1.2.3
   d) 30.4.5.6

   ……

   ……

1.6 FRAGMENTATION

Each network imposes some maximum size on its packets. A problem appears when a large packet wants to travel through a network whose maximum packet size is too small. One solution is to make sure the problem does not occur in the first place. In other words, the internet should use a routing algorithm that avoids sending packets through networks that cannot handle them. However, this solution is no solution at all. What happens if the original source packet is too large to be handled by the destination network? The routing algorithm can hardly bypass the destination.

Basically, the only solution to the problem is to allow gateways to break up packets into **fragments,** sending each fragment as a separate internet packet. However, converting a large object into small fragments is considerably easier than the reverse process.

Two opposing strategies exist for recombining the fragments back into the original packet.

1. Transparent Fragmentation
2. Non Transparent Fragmentation

Transparent Fragmentation

The first strategy is to make fragmentation caused by a "small-packet" network transparent to any subsequent networks through which the packet must pass on its way to the ultimate destination. This option is shown in Figure 19 (a). In this approach, the small-packet network has gateways that interface to other networks. When an oversized packet arrives at a gateway, the gateway breaks it up into fragments. Each fragment is addressed to the same exit gateway, where the pieces are recombined. In this way, passage through the small-packet network has been made transparent. Subsequent networks are not even aware that fragmentation has occurred.

Do you know?

ATM networks have special hardware to provide transparent fragmentation of packets into cells and then reassembly of cells into packets. In the ATM world, fragmentation is called segmentation

Transparent fragmentation is straightforward as shown in figure 19 a.

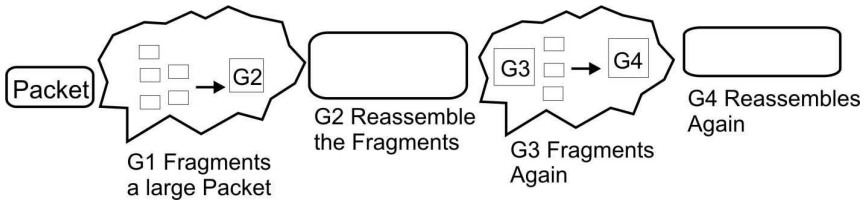


Figure 19 a: Transparent fragmentation

Drawback of transparent fragmentation

1. The exit gateway must know when it has received all the pieces, so either a count field or an "end of packet" bit must be provided.

2. All packets must exit via the same gateway. By not allowing some fragments to follow one route to the ultimate destination and other fragments a disjoint route. some performance may be lost.

3. A last problem is the overhead required to repeatedly reassemble and then re-fragment a large packet passing through a series of small packet networks.

Tips

ATM requires transparent fragmentation.

Non Transparent Fragmentation

The nontransparent fragmentation strategy refrains from recombining fragments at any intermediate gateways. Once a packet has been fragmented, each fragment is treated as though it were an original packet. All fragments are passed through the exit gateway (or gateways), as shown in Figure 19 (b). Recombination occurs only at destination host. IP works this way.

Non Transparent fragmentation also has some problems. For example, it requires every host to be able to do reassembly. Yet another problem is that when the large packet is fragmented the total overhead increases, because each fragment must have a header.

An advantage of this method is that multiple exit gateways can now be used and higher performance can be achieved

When a packet is fragmented, the fragments must be numbered in such a way that the original data stream can be reconstructed. One way of numbering the fragments is to use a tree. If packet 0 must be split up, the pieces are called 0.0, 0.1, 0.2 etc. If these fragments themselves must be fragmented later on, the pieces are numbered 0.0.0, 0.0.1, 0.0.2, ….0.1.2 etc. If enough fields have been reserved in the header for the worst case and no duplicates generated anywhere, this scheme is sufficient to ensure that all the pieces can be correctly reassembled at the destination, no matter what order they arrive in.
However, if even one network loses or discards packets, end-to-end retransmissions are needed, with unfortunate effects for the numbering system. Suppose that a 1024-bit packet is initially fragmented into four equal-sized fragments, 0.0, 0.1, 0.2 and 0.3. Fragment 0.1 is lost, but the other parts arrive at the destination. Eventually, the source times out and retransmits the original packet again. Only this time the route taken passes through a network with a 512-bit limit, so two fragments are generated. When the new fragment 0.1 arrives at the destination, the receiver will think that all four pieces are now accounted for a reconstruct the packet incorrectly.

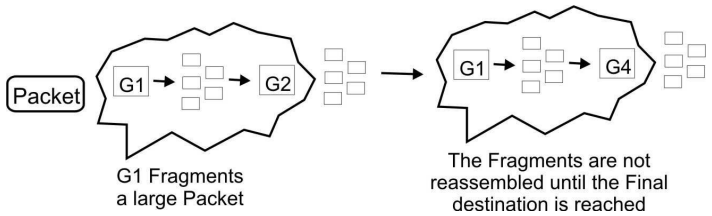


Figure 19 b: Nontransparent fragmentation

1.7 ERROR MESSAGING SERVICES

1.7.1 ICMP (Internet Control Message Protocol)

The Internet Control Message Protocol (ICMP) is a helper protocol that supports IP with facility for Error reporting Simple queries.

ICMP messages are sent in following situations

- when a datagram cannot reach its destination,
- when the gateway does not have the buffering capacity to forward a datagram,
- When the gateway can direct the host to send traffic on a shorter route.

Do you know?
ICMP is considered an integral part of IP as shown in figure 20.

The Internet Protocol (IP) is not designed to be absolutely reliable. The purpose of these control messages is to provide feedback about problems in the communication environment, not to make IP reliable. There are still no guarantees that a datagram will be delivered or a control message will be returned. Some datagrams may still be undelivered without any report of their loss. The higher level protocols that use IP must implement their own reliability procedures if reliable communication is required. The ICMP messages typically report errors in the processing of datagrams. To avoid the infinite regress of messages about messages etc., no ICMP messages are sent about ICMP messages.

TIPS
ICMP provides error reporting, flow control and first-hop gateway redirection.

Position of ICMP in the network layer

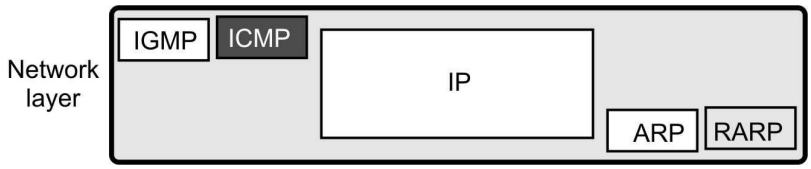


Figure 20: Protocols of Internet layer (TCP/IP)

ICMP header format

It consists of following fields

- Type
- Code
- ICMP header checksum
- Data

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								Code								ICMP header checksum															
Data																															

The fields can be described as follows

1. **Type.** It is of 8 bits. It specifies the format of the ICMP message.
2. **Code.** It is of 8 bits. It further qualifies the ICMP message.
3. **ICMP Header Checksum.** It is of 16 bits. It is checksum that covers the ICMP message. This is the 16-bit one's complement of the one's complement sum of the ICMP message starting with the Type field. The checksum field should be cleared to zero before generating the checksum.
4. **Data.** It is of variable length. It contains the data specific to the message type indicated by the Type and Code fields

Types of ICMP messages

Each ICMP message contains three fields that define its purpose and provide a checksum. They are

- TYPE,
- CODE, and
- CHECKSUM fields (described above).

The TYPE field identifies the ICMP message, the CODE field provides further information about the associated TYPE field, and the CHECKSUM provides a method for determining the integrity of the message.

Tips

ICMP message are sent as packet so these are also called ICMP packet

ICMP messages are divided into two categories

- Error-reporting messages.
- Query messages.

Tips

ICMP messages are identified by "type" numbers

The **error-reporting** messages report problems that a router or a host (destination) may encounter. The **query messages** get specific information from a router or another host. For example, this can be used by the hosts to discover the routers present in their network. The host would send a ICMP query asking for routers to respond. The outers present in the network will respond with an ICMP reply message. The host would get information about the router from this reply.

Examples of error reporting messages

- Destination unreachable
- Source quench
- Time exceeded
- Parameter problem
- Redirection

Example of query messages

- Echo request and reply
- Timestamp request and reply
- Address mask request and reply
- Router solicitation and advertisement

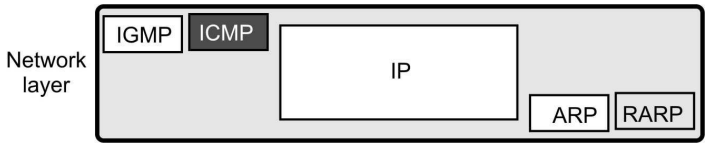
1.7.2 IGMP(Internet Group Message Protocol)

IGMP is a protocol that manages group membership. The IGMP protocol gives the multicast routers information about the membership status of hosts (routers) connected to the network.

Tips

Internet Group Management Protocol (IGMP) is the protocol used to support multicasting.

Position of IGMP in the network layer



Types of messages in IGMP

IGMP has three types of messages:

- the query,
- the membership report,
- and the leave report.

There are two types of query messages as shown in figure 21, general and special

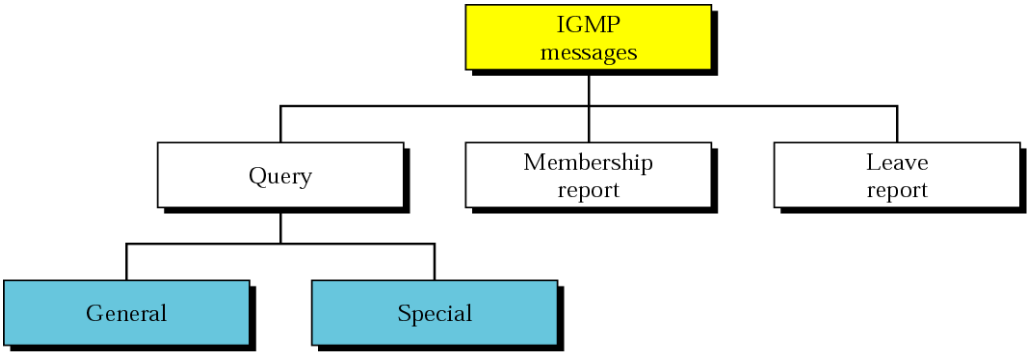


Figure 21: Types of query messages in IGMP

IGMP frame format

It consists of following fields

- Type
- Maximum response time
- Checksum
- Group addresses

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								Code								IGMP Checksum															
Identifier																															
Group Address																															
Access Key																															

The fields can be described as follows

1. **Type.** It is of 8 bits. Following types of IGMP messages are possible

2. **Code.** It is of 8 bits. In a Create Group Request message, this field indicates if the new host group is to be public or private. In all other Request messages, this field is set to zero.

   In a Reply message, the *Code* field specifies the outcome of the request.

3. **IGMP Checksum.** It is of 16 bits. The checksum is the 16-bit one's complement of the one's complement sum of the IGMP message starting with the IGMP Type. For computing the checksum, the checksum field should first be cleared to 0. When the data packet is transmitted, the checksum is computed and inserted into this field. When the data packet is received, the checksum is again computed and verified against the checksum field. If the two checksums do not match then an error has occurred.

4. **Identifier.** It is of 32 bits. In a confirm Group Request message, the identifier field contains zero. In all other Request messages, the identifier field contains a value to distinguish the request from other requests by the same host. In a Reply message, the identifier field contains the same value as in the corresponding Request message.

5. **Group Address.** It is of 32 bits. In a Create Group Request message, the group address field contains zero. In all other Request messages, the group address field contains a host group address. In a Create Group Reply message, the group address field contains either a newly allocated host group address (if the request is granted) or zero (if denied). In all other Reply messages, the group address field contains the same host group address as in the corresponding Request message.

6. **Access Key.** This field is of 64 bits. In a Create Group Request message, the access key field contains zero. In all other Request messages, the access key field contains the access key assigned to the host group identified in the Group Address field (zero for public groups). In a Create Group Reply message, the access key field contains either a non-zero 64-bit number (if the request for a private group is granted) or zero. In all other Reply messages, the access key field contains the same access key as in the corresponding Request.

Do you know?

IGMP is defined in RFC 1112.

1.8 SUMMARY

In this unit, we studied various design issues of network layer. Network layer provides best route from source to destination using adaptive routing algorithm like distance vector routing and link state routing. A serous drawback of distance vector routing is count to infinity problem. It is also responsible for congestion control using leaky bucket and token leaky bucket algorithm. The four main protocols that operates on network layer are ARP, RARP, ICMP, IGMP. Network layer mainly works on IP address. IP addresses are 32bits.IP addresses have been divided into five classes namely A,B,C,D,E.ICMP and .ICMP and IGMP are error reporting protocols.ARP and RARP are used for address translation.

1.9 REFERENCES/FURTHER READING

1. Introduction to Data Communication & Networking, 3rd Edition, Behrouz Forouzan, Tata McGraw Hill.

2. Computer Networks, A. S. Tanenbaum 4th Edition, Practice Hall of India, New Delhi. 2003.
3. Douglas E. Comer, Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture (4th Edition).
4. James F. Kurose, Computer Networking: A Top-Down Approach Featuring the Internet (3rd Edition).
5. Larry L. Peterson, Computer Networks: A Systems Approach, 3rd Edition (The Morgan Kaufmann Series in Networking).
6. www. wikipedia.org
7. W. Richard Stevens, The Protocols (TCP/IP Illustrated, Volume 1).
8. William Stallings, Data and Computer Communications, Seventh Edition.

1.10 SOLUTION/ANSWERS

☞ Check Your Progress 1

1. D
2. D
3. A
4. C
5. A

☞ Check Your Progress 2

1. C
2. D
3. C