RGPVNOTES.IN

Subject Name: **Computer Network**

Subject Code: **IT-5003**

Semester: **5th**

UNIT IV

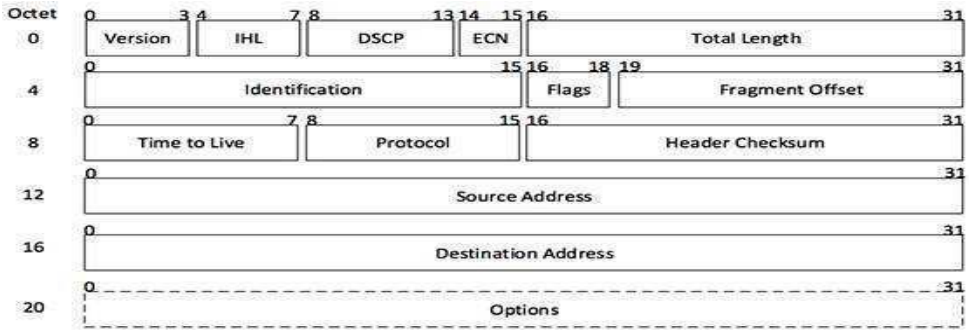
Logical Addressing

IPv4

Internet Protocol version 4 (IPv4) is the fourth version in the development of the Internet Protocol (IP) and the first version of the protocol to be widely deployed. IPv4 is described in IETF publication RFC 791 (September 1981), replacing an earlier definition (RFC 760, January 1980).

(IP Encapsulation)

The encapsulated data is referred to as IP Payload. IP header contains all the necessary information to deliver the packet at the other end.



[Image: IP Header]

***Version*:** Version no. of Internet Protocol used (e.g. IPv4).

***IHL*:** Internet Header Length; Length of entire IP header.

***DSCP*:** Differentiated Services Code Point; this is Type of Service.

ECN: Explicit Congestion Notification; It carries information about the congestion seen in the route.

***Total Length*:** Length of entire IP Packet (including IP header and IP Payload).

Identification: If IP packet is fragmented during the transmission, all the fragments contain same **Identification number**. to identify original IP packet, they belong to.

Flags*:* As required by the network resources, if IP Packet is too large to handle, these 'flags' tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.

Fragment Offset*:* This offset tells the exact position of the fragment in the original IP Packet.

Time to Live: To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.

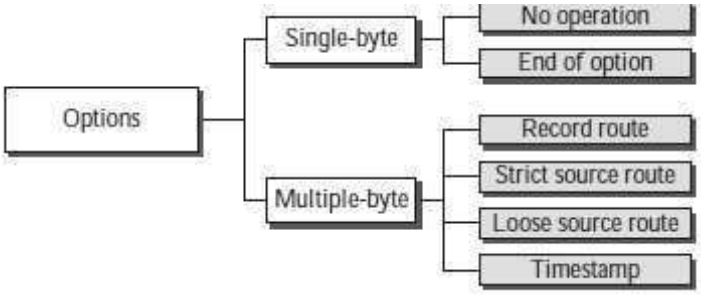
Protocol: Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example, protocol number of ICMP is 1, TCP is 6 and UDP are 17.

Header Checksum: This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.

Source Address: 32-bit address of the Sender (or source) of the packet.

Destination Address: 32-bit address of the Receiver (or destination) of the packet.

Options: This is an optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc.



IPv4 supports three different types of addressing modes.

Unicast Addressing Mode:

In this mode, data is sent only to one destined host. The Destination Address field contains 32- bit IP address of the destination host. Here the client sends data to the targeted server.

Broadcast Addressing Mode:

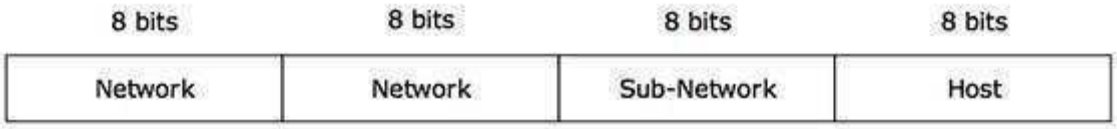
In this mode, the packet is addressed to all the hosts in a network segment. The Destination Address field contains a special broadcast address, i.e. 255.255.255.255. When a host sees this packet on the network, it is bound to process it. Here the client sends a packet, which is entertained by all the Servers:

Multicast Addressing Mode:

This mode is a mix of the previous two modes, i.e. the packet sent is neither destined to a single host nor all the hosts on the segment. In this packet, the Destination Address contains a special address which starts with 224.x.x.x and can be entertained by more than one host.

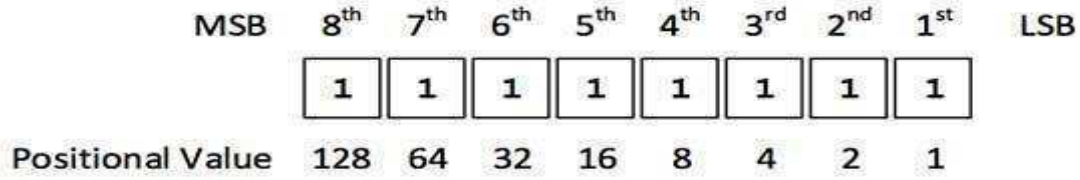
Hierarchical Addressing Scheme

IPv4 uses hierarchical addressing scheme. An IP address, which is 32-bits in length, is divided into two or three parts as depicted.



Binary Representation

The positional value method is the simplest form of converting binary from a decimal value. The IP address is a 32-bit value which is divided into 4 octets. A binary octet contains 8 bits and the value of each bit can be determined by the position of bit value '1' in the octet.



Positional value of bits is determined by 2 raised to power (position – 1), that is the value of a bit 1 at position 6 is 2^ (6-1) that is 2^5 that is 32. The total value of the octet is determined by adding up the positional value of bits. The value of 11000000 is 128+64 = 192.

Addressing Scheme

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet; an IP address is the address of the interface.

Address Space

A protocol like IPv4 that defines addresses has an address space. An address space is the total number of

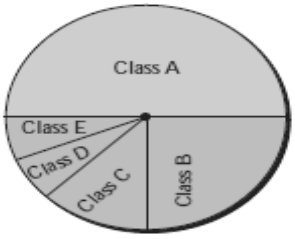
addresses used by the protocol. If a protocol uses b bits to define an address, the address space is 2b because each bit can have two different values (0 or 1). IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than four billion). Theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet.

The address space of IPv4 is 232 or 4,294,967,296.

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 that supersedes the original architecture.

Classful Addressing Classes

In classful addressing, the IP address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the whole address space.



Class A: 2^{31} = 2,147,483,648 addresses, 50%

Class B: 2^{30} = 1,073,741,824 addresses, 25%

Class C: 2^{29} = 536,870,912 addresses, 12.5%

Class D: 2^{28} = 268,435,456 addresses, 6.25%

Class E: 2^{28} = 268,435,456 addresses, 6.25%

Class A Address

The first bit of the first octet is always set to 0 (zero). Thus, the first octet ranges from 1 – 127, i.e.

00000001 – 01111111
1 – 127

Class B Address

An IP address which belongs to class B has the first two bits in the first octet set to 10, i.e.

10000000 – 10111111
128 – 191

Class C Address

The first octet of Class C IP address has its first 3 bits set to 110, that is:

11000000 – 11011111
192 – 223

Class D Address

Very first four bits of the first octet in Class D IP addresses are set to 1110, giving a range of:

11100000 – 11101111
224 – 239

Class E Address

This IP Class is reserved for experimental purposes only for R&D or Study. IP addresses in this class range from 240.0.0.0 to 255.255.255.254.

Private IP Addresses

Every class of IP, (A, B & C) has some addresses reserved as Private IP addresses. These IPs can be used within a network, campus, company and are private to it. These addresses cannot be routed on the Internet, so packets containing these private addresses are dropped by the Routers.

Class A IP Range	Subnet Mask
10.0.0.0 – 10.255.255.255	255.0.0.0
172.16.0.0 – 172.31.255.255	255.240.0.0
192.168.0.0 – 192.168.255.255	255.255.0.0

Loopback IP Addresses

The IP address range 127.0.0.0 – 127.255.255.255 is reserved for loopback, i.e. a Host's self-address, also known as localhost address.

Link-local Addresses

In case a host is not able to acquire an IP address from the DHCP server and it has not been assigned any IP address manually, the host can assign itself an IP address from a range of reserved Link-local addresses. Link-local address ranges from 169.254.0.0 -- 169.254.255.255.

Net-id and Host-id

In classful addressing, an IP address in classes A, B, and C is divided into net-id and host-id. These parts are of varying lengths, depending on the class of the address.

Default Mask

The routers on the Internet normally use an algorithm to extract the network address from the destination address of a packet. To do this, we need a network mask. A network mask or a default mask in classful addressing is a 32-bit number with n leftmost bits all set to 1s and (32 − n) rightmost bits all set to 0s. Since n is different for each class in classful addressing, we have three default masks in classful addressing.

Class	0----------32 bits			
	8 bits	8 bits	8 bits	8 bits
A	N-id 255	H-id 0	H-id 0	H-id 0
B	N-id 255	N-id 255	H-id 0	H-id 0
C	N-id 255	N-id 255	N-id 255	H-id 0
D	Multicast Addresses			
E	Reserved for future			

Network Mask

The methods we described previously for extracting the network address are mostly used to show the concept. The routers on the Internet normally use an algorithm to extract the network address from the destination address of a packet. To do this, we need a network mask. A network mask or a default mask in classful addressing is a 32-bit number with n leftmost bits all set to 1s and (32 − n) right most bits all set to 0s.

Network mask

8 bits 24 bits

Mask for class A 11111111 00000000 00000000 00000000 --255.0.0.0

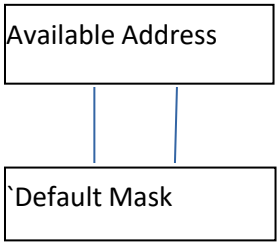
16 bits 16 bits

Mask for class B 11111111 11111111 00000000 00000000 -- 255.255.0.0

24 bits 8 bits

Mask for class C 11111111 11111111 11111111 000000000 – 255.255.255.0

Finding a network address using the default mask



AND operation

Network Address

Three-Level Addressing: Subnetting

The IP addresses were originally designed with two levels of addressing. To reach a host on the Internet, we must first reach the network and then the host. It soon became clear that we need more than two hierarchical levels, for two reasons. First, an organization that was granted a block in class A or B needed to divide its large network into several subnetworks for better security and management.

Subnet Mask

We discussed the network mask (default mask) before. The network mask is used when a network is not subnetted. When we divide a network to several subnetworks, we need to create a subnetwork mask (or subnet mask) for each subnetwork. A subnetwork has subnet and hosted.

Network Mask

Net-id (n bits)	Host-id (32-n bits)

Subnet Mask

Subnet-id (n bits)	Host-id (32-n bits)

Subnetting increases the length of the netid and decreases the length of hostid. When we divide a network to s number of subnetworks, each of equal numbers of hosts, we can calculate the subnetid for each subnetwork as **nsub = n + log2s**

Example: -

Address → **141. 14. 120. 77**

Mask → **255. 255 .192. 0**

Subnet Address → **141. 14. 64. 0**

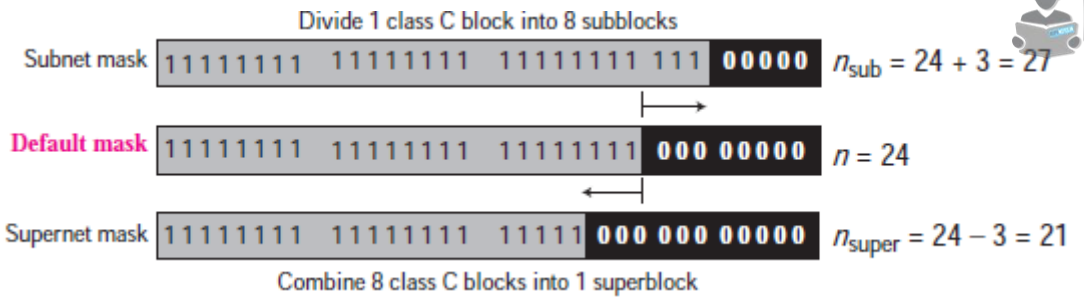
Super netting

Subnetting could not completely solve address depletion problems in classful addressing because most organizations did not want to share their granted blocks with others. Since class C blocks were still available but the size of the block did not meet the requirement of new organizations that wanted to join the Internet, one solution was super netting.

Supernet Mask

A Supernet mask is the reverse of a subnet mask. A subnet mask for class C has more 1s than the default mask for this class. A supernet mask for class C has less 1s than the default mask for this class.

Comparison of subnet, default, and supernet masks.



Calculating Super-net nsuper = n – log2c

CLASSLESS ADDRESSING

Subnetting and super netting in classful addressing did not really solve the address depletion problem and made the distribution of addresses and the routing process more difficult. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses be increased, which means the format of the IP packets needs to be changed. Although the long-range solution has already been devised and is called IPv6 a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called classless addressing.

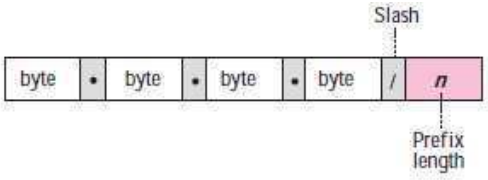
In classless addressing, the prefix defines the network and the suffix defines the host.

The prefix length in classless addressing can be 1 to 32.

Slash Notation

The netid length in classful addressing or the prefix length in classless addressing play a very important role when we need to extract the information about the block from a given address in the block. However, there is a difference here in classful and classless addressing.

The slash notation is formally referred to as classless Interdomain routing or CIDR (pronounced cider) notation.



Extracting Block Information

An address in slash notation (CIDR) contains all information we need about the block:
the first address (network address), the number of addresses, and the last address. These three pieces of information can be found as follows:

- The number of addresses in the block can be found as $N = 2^{32} - n$ in which n is the prefix length and N being the number of addresses in the block.

- The first address (network address) in the block can be found by ANDing the address with the network mask: **First address = (any address) AND (network mask)**
- Alternatively, we can keep the n leftmost bits of any address in the block and set the 32– n bits to 0s to

find the first address.

- The last address in the block can be found by either adding the first address with the number of addresses or, directly, by ORing the address with the complement (Noting) of the network mask: **Last address = (any address) OR [NOT (network mask)]**

Alternatively, we can keep the n leftmost bits of any address in the block and set the 32 − n bits to 1s to find the last address.

One of the addresses in a block is 17.63.110.114/24. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.255.0.

a. The number of addresses in the network is 232 − 24 = 256.

b. To find the first address the first address is 17.63.110.0/24.

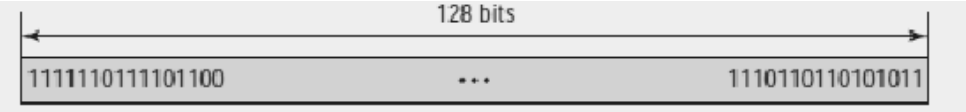
Address:	17. 63. 110. 114
Network mask:	255. 255. 255. 0
First address (AND):	17. 63. 110. 0

c. To find the last address, we use the complement of the network mask a. The last address is 17.63.110.255/24.

Address:	17. 63. 110. 114
Complement of the mask (NOT):	0. 0. 0. 255
Last address (OR):	17. 63. 110. 255

IPV6

An IPv6 address is 128 bits or 16 bytes (octet) long. The address length in IPv6 is four times of the length address in IPv4.



Notations

A computer normally stores the address in binary but is clear that 128 bits cannot easily be handled by humans. Several notations have been proposed to represent IPv6 addresses when they are handled by humans.

Dotted-Decimal Notation

To be compatible with IPv4 addresses, we are tempted to use dotted-decimal notation as shown for IPv4 addresses in Chapter 5. Although this notation is convenient for 4-byte IPv4 addresses, it seems too long for 16-byte IPv6 addresses as shown below:

221.14.65.11.105.45.170.34.12.234.18.0.14.0.115.255

Colon Hexadecimal Notation

To make addresses more readable, IPv6 specifies colon hexadecimal notation (or colon hex for short). In this notation, 128 bits are divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation require four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.

FDEC BA98 7654 3210 ADBF BBFF 2922 FFFF

Mixed Representation

Sometimes we see a mixed representation of an IPv6 address: colon hex and dotted decimal notation. This is appropriate during the transition period in which an IPv4 address is embedded in an IPv6 address (as the rightmost 32 bits).

FDEC:14AB:2311: BBFE: AAAA: BBBB:130.24.24.18

CIDR Notation

As we see shortly, IPv6 uses hierarchical addressing. For this reason, IPv6 allows classless addressing and CIDR notation.

FDEC BBFF 0 FFFF/60

Address Type in IPV6

In IPv6, a destination address can belong to one of three categories: unicast, anycast, and multicast.

Unicast Address

A unicast address defines a single interface (computer or router). The packet sent to a unicast address will be routed to the intended recipient. As we see shortly, IPv6 has designated a large block from which unicast addresses can be assigned to interfaces.

Anycast Address

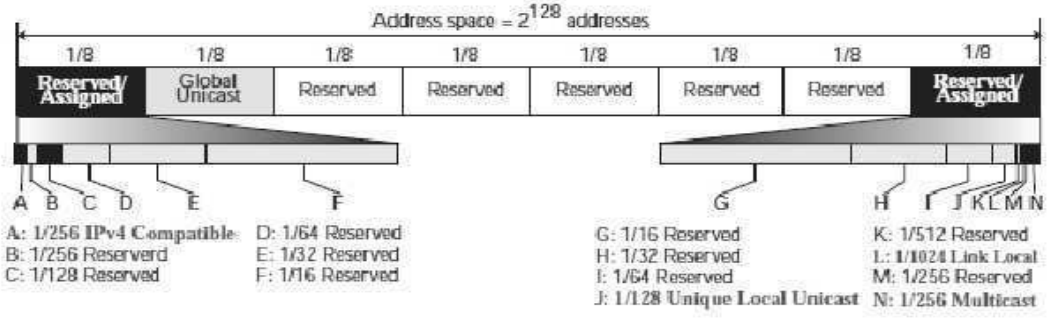
Anycast address defines a group of computers that all share a single address. A packet with anycast address is delivered to only one member of the group, the most reachable one. An anycast communication is used, for example, when there are several servers that can respond to an inquiry.

Multicast Address

A multicast address also defines a group of computers. However, there is a difference between any casting and multicasting. In multicasting, each member of the group receives a copy.

Address space allocation

Address space of IPv6 is divided into several blocks of varying size and each block is allocated for the special purpose. Most of the blocks are still unassigned and have been left aside for future use. To better understand the allocation and the location of each block in address space, we first divide the whole address space into eight equal ranges.



Comparison between IPV4 & IPV6

IPv4	IPv6
IPv4 addresses are the 32-bit length.	IPv6 addresses are the 128-bit length.
IPv4 addresses are binary numbers represented in decimals.	IPv6 addresses are binary numbers represented in hexadecimal.
IPsec support is only optional.	Inbuilt IPsec support.
Fragmentation is done by sender and forwarding routers.	Fragmentation is done only by the sender.
No packet flow identification.	Packet flow identification is available within the IPv6 header using the Flow Label field.
Checksum field is available in IPv4 header	No checksum field in IPv6 header.
Options fields are available in IPv4 header.	No option fields, but IPv6 Extension headers are available.
Address Resolution Protocol (ARP) is available to map IPv4 addresses to MAC addresses.	Address Resolution Protocol (ARP) is replaced with a function of Neighbor Discovery Protocol (NDP).
Internet Group Management Protocol (IGMP) is used to manage multicast group membership.	IGMP is replaced with Multicast Listener Discovery (MLD) messages.
Broadcast messages are available.	Broadcast messages are not available. Instead, a link-local scope "All nodes" multicast IPv6 address (FF02::1) is used for broadcast similar functionality.
Manual configuration (Static) of IPv4 addresses or DHCP (Dynamic configuration) is required to configure IPv4 addresses.	Auto-configuration of addresses is available.

CONGESTION

CONGESTION CONTROL

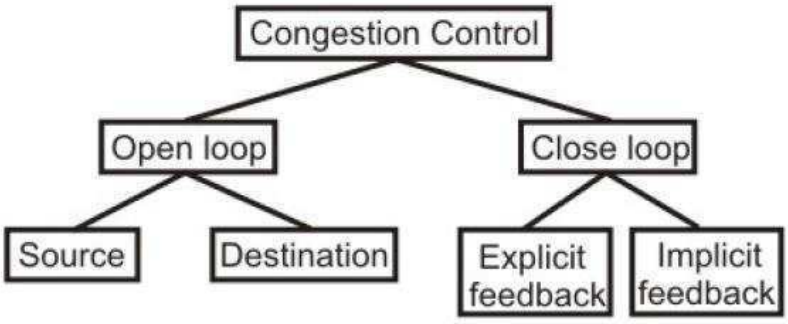
An important issue in a packet-switched network is congestion. Congestion in a network may occur if the load on the network the number of packets sent to the network is greater than the capacity of the network the number of packets a network can handle.

Causes of Congestion

Congestion can occur due to several reasons. For example, if all of a sudden, a stream of packets arrives on several input lines and need to be out on the same output line, then a long queue will be build up for that output. If there is *insufficient memory* to hold these packets, then packets will be lost (dropped). Adding more memory also may not help in certain situations. If the router has an infinite amount of memory even then instead of congestion being reduced, it gets worse; because by the time packets gets at the head of the queue, to be dispatched out to the output line, they have already timed-out (repeatedly), and duplicates may also be present.

Slow processors also cause Congestion. If the router CPU is slow at performing the task required for them (Queuing buffers, updating tables, reporting any exceptions etc.), the queue can build up even if there is an excess of line capacity. Similarly, *Low-Bandwidth* lines can also cause congestion. Upgrading lines but not changing slow processors, or vice-versa, often helps a little; these can just shift the bottleneck to some other point. The real problem is the mismatch between different parts of the system.

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion after it has happened.



In general, we can divide congestion control mechanisms into two broad categories:

- **Open-loop congestion control (prevention)**
- **Closed-loop congestion control (removal)**

The first category of solutions or protocols attempts to solve the problem by a good design, at first, to make sure that it doesn't occur at all. Once the system is up and running midcourse corrections are not made. These solutions are somewhat static in nature, as the policies to control congestion don't change much according to the current state of the system. Such Protocols are also known as *Open Loop* solutions. These rules or policies include deciding upon when to accept traffic, when to discard it, making scheduling decisions and so on. The main point here is that they decide without taking into consideration the current state of the network. The open-loop algorithms are further divided based on whether these acts on source versus that act upon destination.

The second category is based on the concept of feedback. During operation, some system parameters are measured and feedback to portions of the subnet that can act to reduce the congestion. This. The approach can be divided into 3 steps:

Monitor the system (network) to detect whether the network is congested or not and what's the actual location and devices involved.

To pass this information to the places where actions can be taken

Adjust the system operation to correct the problem.

Popular algorithms from the above categories.

Leaky Bucket Algorithm

Consider a Bucket with a small hole at the bottom, whatever may be the rate of water pouring into the bucket, the rate at which water comes out from that small hole is constant. Once the bucket is full, any additional water entering it spills over the sides and is lost (i.e. it doesn't appear in the output stream through the hole underneath).

Conceptually each network interface contains a *leaky bucket*. And the following steps are performed:

When the host should send a packet, the packet is thrown into the bucket.

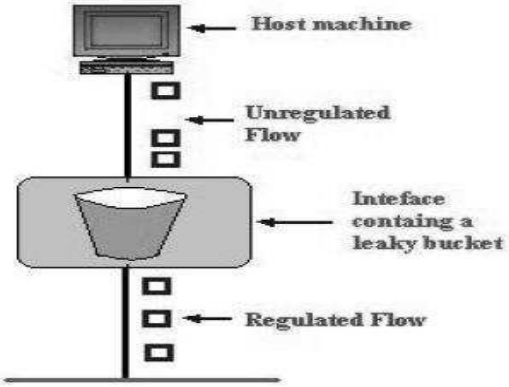
The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.

Burstyn traffic is converted to a uniform traffic by the leaky bucket.

In practice, the bucket is a finite queue that outputs at a finite rate.

This arrangement can be simulated in the operating system or can be built into the hardware.

Implementation of this algorithm is easy and consists of a finite queue. Whenever a packet arrives, if there is room in the queue it is queued up and if there is no room then the packet is discarded.



Token Bucket Algorithm

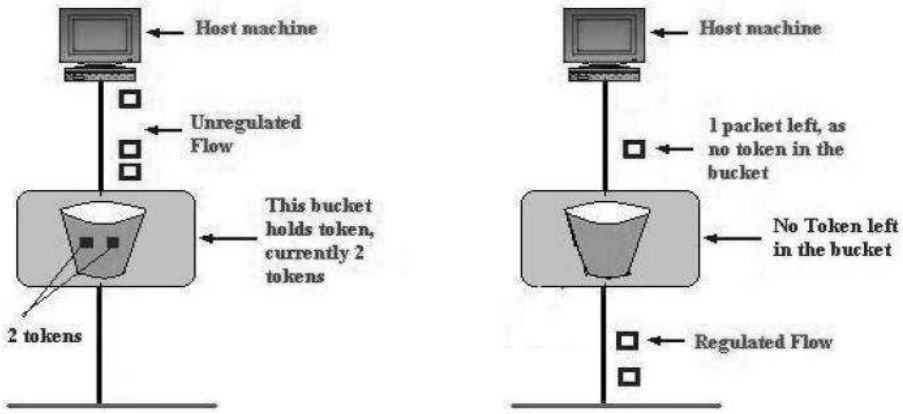
The leaky bucket algorithm described above, enforces a rigid pattern at the output stream, irrespective of the pattern of the input. For many applications, it is better to allow the output to speed up somewhat when a larger burst arrives than to lose the data. Token Bucket algorithm provides such a solution. In this algorithm leaky bucket holds token, generated at regular intervals. Main steps of this algorithm can be described as follows:

In regular intervals, tokens are thrown into the bucket.

The bucket has a maximum capacity.

If there is a ready packet, a token is removed from the bucket, and the packet is sent.

If there is no token in the bucket, the packet cannot be sent.



Routing

Is it the act of moving information across an internetwork from a source to a destination? Along the way, at least one intermediate node typically is encountered. It's also referred to as the process of choosing a path over which to send the packets. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer.

Desirable properties of a router are as follows:

- **Correctness and simplicity:** The packets are to be correctly delivered. Simpler the routing algorithm, it

is better.

- **Robustness:** The ability of the network to deliver packets via some route even in the face of failures.
- **Stability:** The algorithm should converge to equilibrium fast in the face of changing conditions in the network.
- **Fairness and optimality:** Obvious requirements, but conflicting.
- **Efficiency:** Minimum overhead While designing a routing protocol it is necessary to consider the following design parameters:
- **Performance Criteria:** Many hops, Cost, Delay, Throughput, etc.
- **Decision Time:** Per packet basis (Datagram) or per session (Virtual-circuit) basis
- **Decision Place:** Each node (distributed), Central node (centralized), Originated node (source)
- **Network Information Source:** None, Local, Adjacent node, Nodes along route, All nodes
- **Network Information Update Timing:** Continuous, Periodic, Major load change, Topology change

Routing Tables: -

Static versus Dynamic Routing Tables

A routing table can be either static or dynamic. A static table is one with manual entries. A dynamic table, on the other hand, is one that is updated automatically when there is a change somewhere on the internet.

Routing Protocol

Routing protocols have been created in response to the demand for dynamic routing tables. A routing protocol is a combination of rules and procedures that let routers on the internet inform each other of changes.

Routing protocols can be either an interior protocol or an exterior protocol.

- An interior protocol handles intradomain routing.
- An exterior protocol handles Interdomain routing

Routing Protocol	
Intradomain	**Interdomain**
Distance vector routing-RIP	**Path Vector-BGP**
link State routing-OSPF	

Intra-domain Routing	Inter-domain Routing
Routing takes place within an autonomous network.	Routing takes place between the two autonomous networks.
This protocol ignores the internet outside the autonomous system.	This protocol assumes that the internet consists of a collection of interconnected autonomous systems.
Protocols for Intra-domain routing are called as **interior gateway protocols.**	Protocol for Inter-domain routing is also called as **exterior gateway protocols.**
Examples: RIP and OSPF etc.	**Example:** BGP

An **autonomous system** is a group of the networks and the routers, which are operated by the network administrator. The Internet can be divided into autonomous systems. **Distance vector and link state routing** are the examples of Intra-domain routing.

Routing inside an autonomous system is referred to as intra-domain routing.

Routing between two or more autonomous systems can be referred to as **inter-domain routing**. **Path vector** is an example of an inter-domain routing.

DISTANCE VECTOR ROUTING

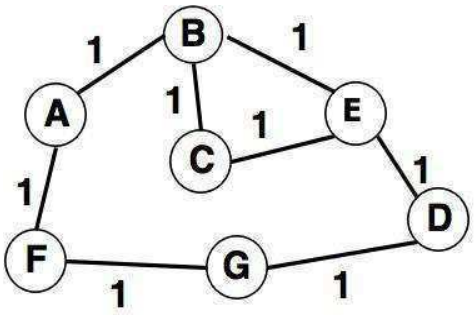
Distance vector routing is the dynamic routing algorithm and known as **Bellman-Ford** routing algorithm and **Ford- Fulkerson** algorithm.

It was designed for small network topologies.

In this algorithm, node router constructs a table containing the distance (total cost of the path) to all other nodes and distributes that vector to its immediate neighbors.

For distance vector routing, it is assumed that each node knows the cost of the link to each of its directly connected neighbors.

A link, which is 'down' (which is not working) is assigned as an infinite cost.



Distance Vector Routing

The shortest path can be computed as:

Information at Node	Cost to Reach Node						
	A	B	C	D	E	F	G
A	0	1	2	3	2	1	2
B	1	0	1	2	1	2	3
C	2	1	0	2	1	3	3
D	3	2	2	0	1	2	2
E	2	1	1	1	0	3	2
F	1	2	3	2	3	0	1
G	2	3	3	1	2	1	0

Every node sends a message to its directly connected neighbors **for example** A sends its information to B and F.

After communicating to each directly connected node the shortest path can be easy to compute (as shown in above table).

Some issues with the Distance Vector Routing are:

1. Vulnerability to the 'Count-to-Infinity' problem is a serious issue with the distance vector.
2. It takes a long time for convergence due to growth in the network.

Count to Infinity

A problem with distance vector routing is that any decrease in cost (good news) propagates quickly, but any increase in cost (bad news) propagates slowly. For a routing protocol to work properly if a link is broken (cost becomes infinity), every other router should be aware of it immediately, but in distance vector routing, this takes some time. The problem is referred to as count to infinity. It takes several updates before the cost

for a broken link is recorded as infinity by all routers.

Solution to Count to Infinity problem

Defining Infinity

Split Horizon

Link State Routing

It is a dynamic type routing algorithm.

In this method, one or more routers can be connected by using LAN.

When a router is booted, it sends a special request (HELLO packet) message on each point-to-point line. Then the second router sends back a reply and asks who is it and the communication starts.

To determine the cost of line or path, the router sends an ECHO packet over the line which the other router is required to send back immediately. By measuring the round-trip time and dividing it by two, the router (sender) can get a reasonable estimate of the delay.

The link-state packet can be constructed periodically or after the occurrence of some significant event. For example: if a line or neighbor is down or it may be coming back.

Basic algorithm to distribute the link state packets:

Each state packet has a sequence number and it is incremented for each sent packet.

Routers can track all the source routers and sequence.

When a new link state packet arrives, it is checked against the list of packets already entered. If the packet is new, it is forwarded on all lines (except on which it arrives i.e. flooding) and discarded, if the packet is duplicated. If the sequence number is lower (than the highest one), it is rejected.

Some changes to improve the basic algorithm:

Once the router accumulates a full set of link state packets, it can construct the entire subnet graph and Dijkstra's algorithm can be used to construct the shortest path to all possible destination.

Link-state routing protocol uses event driven updates rather than periodic updates.

A link-state routing protocol is widely used in an actual networking system.

Dijkstra's algorithm

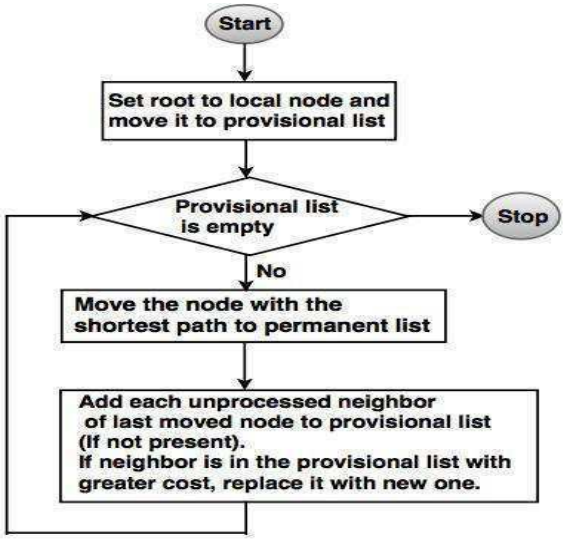
The Dijkstra algorithm creates the shortest path tree from a given graph.

The algorithm divides the nodes into two sets.

I. Provisional

ii. Permanent

The algorithm finds the neighbors of a current node, enlists them as provisional and if neighbors fulfill the criteria, it is stored permanently.



Dijkstra's Shortest Path Algorithm.

Different steps are:

Step 1: Select root node as 'A'

Step 2: Move 'A' to permanent list and add 'B', 'C' and 'D' to provisional list.

Step 3: Move 'C' and add 'E' to provisional list.

Step 4: Move 'D' to permanent list.

Step 5: Move 'B' to permanent list.

Step 6: Move 'E' to permanent list.

RIP

The Routing Information Protocol (RIP) is an intradomain (interior) routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing. RIP implements distance vector routing directly with some considerations:

1. In an autonomous system, we are dealing with routers and networks (links), what was described as a node.
2. The destination in a routing table is a network, which means the first column defines a network address.
3. The metric used by RIP is very simple; the distance is defined as the number of links (networks) that have to be used to reach the destination. For this reason, the metric in RIP is called a hop count.
4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
5. The next node column defines the address of the router to which the packet is to be sent to reach its destination.

RIP packet format: -

		Command	Version	Reserved
Repeated		Family		All 0s
		Network address		
		All 0s		
		All 0s		
		Distance		

Command. This 8-bit field specifies the type of message: request (1) or response (2).

Version. This 8-bit field defines the version. In this book, we use version 1, but at the end of this section,

we give some new features of version 2.

Family. This 16-bit field defines the family of the protocol used. For TCP/IP, the value is 2.

Network address. The address field defines the address of the destination net- work. RIP has allocated 14 bytes for this field to be applicable to any protocol. However, IP currently uses only 4 bytes. The rest of the address is filled with 0s.

Distance. This 32-bit field defines the hop count (cost) from the advertising router to the destination network.

Note that part of the message is repeated for each destination network. We refer to this as an entry.

Requests and Responses

RIP has two types of messages: request and response.

Request

A request message is sent by a router that has just come up or by a router that has some time-out entries.

Response

A response can be either solicited or unsolicited. A solicited response is sent only in answer to a request.

Timers in RIP

Periodic Timer

The periodic timer controls the advertising of regular update messages. Although the protocol specifies that this timer must be set to 30 s, the working model uses a random number between 25 and 35 s.

Expiration Timer

The expiration timer governs the validity of a route. When a router receives update information for a route, the expiration timer is set to 180 s for that particular route.

Garbage Collection Timer

When the information about a route becomes invalid, the router does not immediately purge that route from its table. Instead, it continues to advertise the route with a metric value of 16.

Encapsulation

RIP messages are encapsulated in UDP user datagrams. A RIP message does not include a field that indicates the length of the message. This can be determined from the UDP packet. The well-known port assigned to RIP in UDP is port 520.

OSPF

The Open Shortest Path First (OSPF) protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.

Areas

To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas. An area is a collection of networks, hosts, and routers all contained within an autonomous system.

Metric

The OSPF protocol allows the administrator to assign a cost, called the metric, to each route. The metric can be based on a type of service (minimum delay, maximum throughput, and so on).

Types of Links

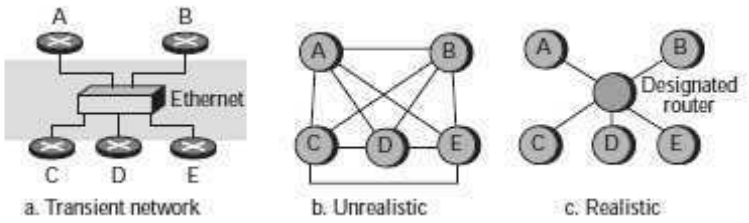
In OSPF terminology, a connection is called a link. Four types of links have been defined: point-to-point, transient, stub, and virtual.

Point-to-Point Link

A point-to-point link connects two routers without any other host or router in between. In other words, the purpose of the link (network) is just to connect the two routers.

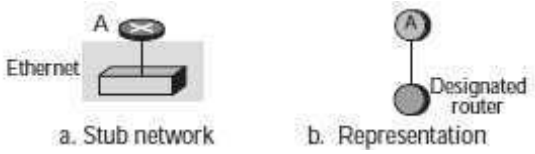
Transient Link

A transient link is a network with several routers attached to it. The data can enter through any of the routers and leave through any router.



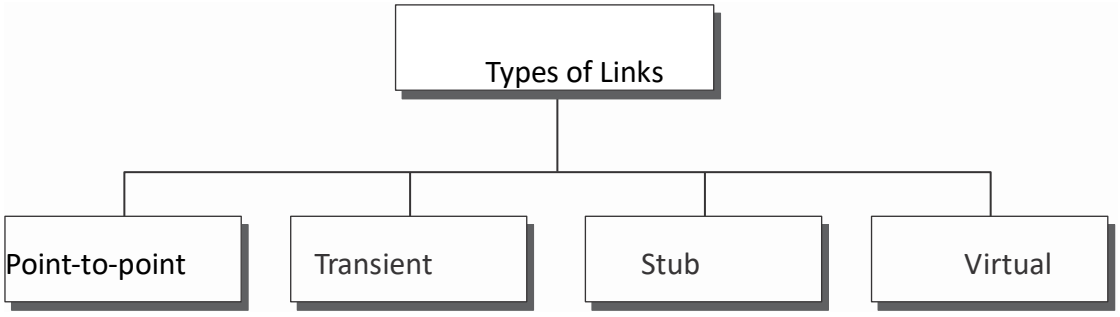
Stub Link

A stub link is a network that is connected to only one router. The data packets enter the network through this single router and leave the network through this same router. This is a special case of the transient network.

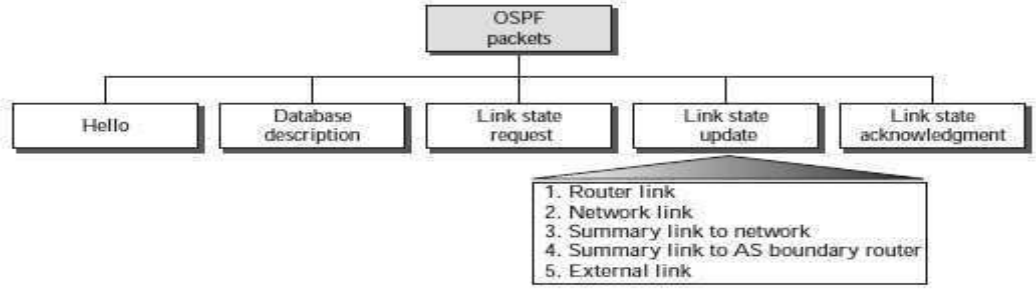


Virtual Link

When the link between two routers is broken, the administration may create a virtual link between them using a longer path that probably goes through several routers.



Types of OSPF packets



Packet Format of OSPF

All OSPF packets have the same common header (see Figure 11.28). Before studying the different types of packets, let us talk about this common header.

Version: This 8-bit field defines the version of the OSPF protocol. It is currently version 2.
Type. This 8-bit field defines the type of the packet. As we said before, we have five types, with values 1 to 5 defining the types.
Message length: This 16-bit field defines the length of the total message including the header.
Source router IP address: This 32-bit field defines the IP address of the router that sends the packet.
Area identification: This 32-bit field defines the area within which the routing takes place.
Checksum: This field is used for error detection on the entire packet excluding the authentication type and authentication data field.
Authentication type: This 16-bit field defines the authentication protocol used in this area. Now, two types of authentication are defined: 0 for none and 1 for a password.
Authentication. This 64-bit field is the actual value of the authentication data. In the future, when more authentication types are defined, this field will contain the result of the authentication calculation.
Packet Format:

<table>
<tr><td>Version</td><td>type</td><td>Message length</td></tr>
<tr><td colspan="3">Source router IP address</td></tr>
<tr><td colspan="3">Area Identification</td></tr>
<tr><td colspan="2">Checksum</td><td>Authentication type</td></tr>
<tr><td colspan="3">Authentication
(32 bits)</td></tr>
</table>

PATH VECTOR ROUTING

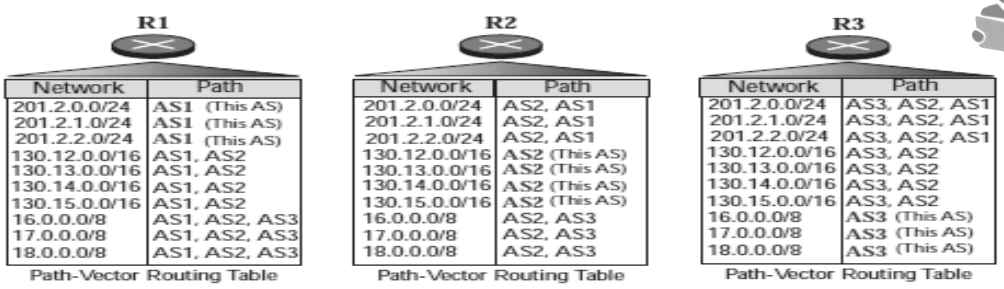
Distance vector and link state routing are both interior routing protocols. They can be used inside an autonomous system as intra-domain or intra-AS (as sometimes are called), but not between autonomous systems. Both routing protocols become intractable when the domain of operation becomes large. Distance vector routing is subject to instability if there are more than a few hops in the domain of operation. Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call path vector routing.

Reachability

To be able to provide information to other ASs, each AS must have at least one path vector routing that collects reachability information about each network in that AS. The information collected in this case only means which network, identified by its network address (CIDR prefix), exists (can be reached in this AS).

Routing Tables

A path vector routing table for each router can be created if ASs share their reachability list with each other. In Figure 11.50, router R1 in AS1 can send its reachability list to router R2. Router R2, after combining its reachability list, can send the result to both R1 and R3. Router R3 can send its reachability list to R2, which in turn improves its routing table, and so on.



BGP

Border Gateway Protocol (BGP) is an Interdomain routing protocol using path vector routing.

Types of Autonomous Systems

As we said before, the Internet is divided into hierarchical domains called autonomous systems (ASs). For example, a large corporation that manages its own network and has full control over it is an autonomous system.

Stub AS

A stub AS has only one connection to another AS. The Interdomain data traffic in a stub AS can be either created or terminated in the AS. The hosts in the AS can send data traffic to another ASs. The hosts in the AS can receive data coming from hosts in another ASs. Data traffic, however, cannot pass through a stub AS. A stub AS is either a source or a sink. A good example of a stub AS is a small corporation or a small local ISP.

Multihomed AS

A multihomed AS has more than one connection to other ASs, but it is still only a source or sinks for data traffic. It can receive data traffic from more than one AS. It can send data traffic to more than one AS, but there is no transient traffic. It does not allow data coming from one AS and going to another AS to pass through. A good example of a multihomed AS is a large corporation that is connected to more than one regional or national AS that does not allow transient traffic.

Transit AS

A transit AS is a multihomed AS that also allows transient traffic. Good examples of transit ASs are national and international ISPs (Internet backbones).

Path Attributes

The path was presented as a list of autonomous systems, but is, in fact, a list of attributes. Each attribute gives some information about the path.

Attributes are divided into two broad categories: well-known and optional. A well-known attribute is one that every BGP router must recognize. An optional attribute is one that needs not be recognized by every router.

Well-Known Attribute

Mandatory: - A well-known mandatory attribute is one that must appear in the description of a route.

Discretionary: - A well-known discretionary attribute is one that must be recognized by each router, but is not required to be included in every update message.

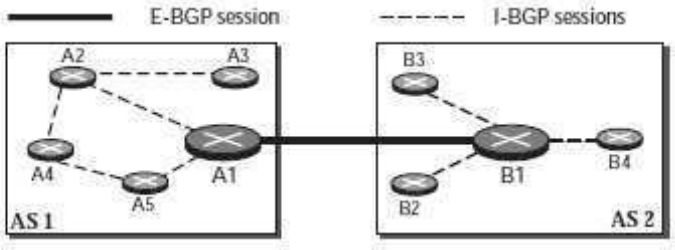
Optional attributes

Transitive: - Is the one that must be passed to the next router by the router that has not implemented this attribute.

Nontransitive: - An optional nontransitive attribute is one that must be discarded if the receiving router has not implemented it.

External and Internal BGP

If we want to be precise, BGP can have two types of sessions: external BGP (E-BGP) and internal BGP (I-BGP) sessions. The E-BGP session is used to exchange information between two speaker nodes belonging to two different autonomous systems. The IBGP session, on the other hand, is used to exchange routing information between two routers inside an autonomous system.



Types of Packets

BGP uses four different types of messages: open, update, keepalive, and notification.

Open Message

To create a neighborhood relationship, a router running BGP opens a TCP connection with a neighbor and sends an open message.

Keepalive message

If the neighbor accepts the neighborhood relationship, it responds with a keepalive message, which means that a relationship has been established between the two routers.

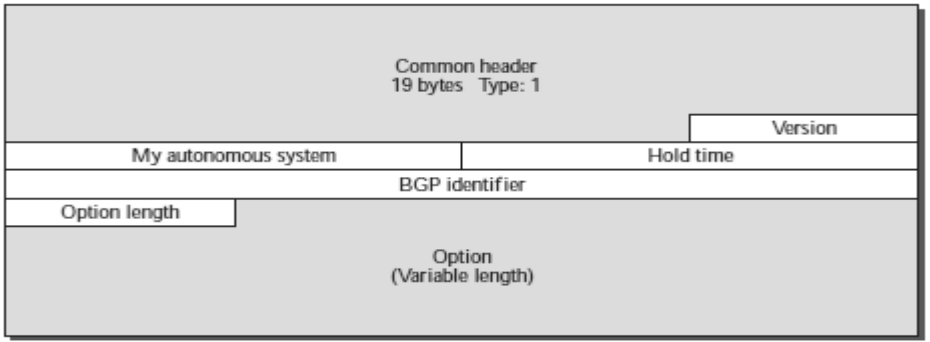
Update Message

The update message is the heart of the BGP protocol. It is used by a router to withdraw destinations that have been advertised previously, announce a route to a new destination, or both.

Notification Message

A notification message is sent by a router whenever an error condition is detected or a router wants to close the connection.

Ex. Open Message Format



RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in