**NETWORK LAYER**

**(IP Addressing, Subnetting, Routing)**

1. **Introduction Network layer**

* Each layer has three unique feature:

1. Services
2. **Interface:** Interfaces are networking communication points for the computer. Each interface is associated with a physical or virtual networking device.
3. Protocol:
   1. **Network Layer Services**

* A linked list is a way to store
* F
* F
* f
  1. **Network Layer Interfaces**
* The interfaces or interconnecting devices are switche, Bridge, Router & gateway
* F

1. **IP Address**
   1. **Definition**

* An address is required to identify a host uniquely in the network.
* **Ethernet address (MAC address)** are globaly unique, but are flat. They have no structure and provide very few clues to routing protocols.
* **IP addresses** are also other ways to address each device uniquely, connected to the internet, but are hierarchical (i.e. means they are made up of several parts). the IP addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the internet.
* IP is a logically addressing. IP addressing has two versions, IPV4 and IPV6.
  1. **Address Space**
* It is the **total number of addresses** used by the protocol.
* If a protocol uses n bits to define an address, the address space is 2n.
* **IPV4 uses a 32-bit addresses.** The address space is 232 =4, 294, 967, 296 which means max. of this number of devices could be connected to the internet.
* **IPV6 uses a 64-bit addresses.**
  1. **Notations used to represent IP Address**
* There are **three common notations** to show an IP address.
* **Binary Notation:** The IP address is displayed as 32-bits. To make the address morre readable, one or more spaceis usually inserted between each octet (8 bits). Each octet is often referred as a byte. So IP address is referred as a 32-bit address, a 4-octet address or 4-byte address.
* **Dotted-Decimal Notation:** To make the IP address more compact and easier to read, IP addresses are usually written in decimal form with a decimal point (dot) separating the bytes. Each number in he dotted decimal notation is between 0 to 255. The IP address range is from 0.0.0.0 to 255.255.255.255
* **Hexa-Decimal Notation:** Each hexa decial digit is equivalent to four bits. Tis means that a 32-bit address has 8 hexadecimal digits.
* **Example**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hexa-Decial Notation | 0X800B031F or (800B031F)16 | | | |
|  | First Octet or First Byte | Second Octet or Second Byte | Third Octet or Third Byte | Fourth Byte or Fourth Byte |
| **Binary Notation** | 10000000 | 00001011 | 00000011 | 00011111 |
| Dotted-Decimal notation | 128.11.3.31 | | | |

* 1. **Classfull Addressing**
* Here, the IP addresses space is divided into ***five classes.***
* **Finding the Class in binary and dotted-decimal notation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Binary Notation**  **(First Byte starts with fixed bit (s))** | **Dotted-Decimal**  **(First Byte decimal is within the range)** | **Used for** |
| **Class-A** | 0.…. | 0-127 | LAN & WAN |
| **Class-B** | 10….. | 128-191 |
| **Class-C** | 110…. | 192-223 |
| **Class-D** | 1110…. | 224-234 | Multicasting |
| **Class-E** | 1111…. | 240-255 | Reserved for future |

* **Netid and Hostid :** In classful addressing, an IP address is class A, B and C is divided into parts, i.e.netid and hostid. These parts are of varying lengths, depending on the class of the address. The network part or netid identify the network to which the host is attached. The host part or hostid identifies each host uniquely on a particular network.
* Any specific netid with all hostid as zeros (i.e. all octet of hostid as 0 in dotted notation) is specially used as **network address.**
* Any specific netid with all hostid as ones (i.e. all octet of hostid as 255 in dotted notation) is specially used as **broadcast address.**
* **Valid IP addresses** are the addresses in between network address and broadcast address. Only valid IP addresses are assigned to the hosts.

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Range of IP Address** | **Dotted-Decimal**  **(netid+hostid)** | **Remarks** |
| **Class-A** | 0.X.X.X  to  127.X.X.X | N.H.H.H | Network bits = one octet = 8 bits  Host bits = three octet = 24 bits  With 8 bits total number of network possible where 1 bit is fixed for class A  = 28-1 = 27 =128  But netid or block starting with 0 and 127 (first and last block respectively), **two networks** in this class are reserved for special purpose.  **One block** or netid 10 is used for private addresses (i.e. for LAN not connected to internet, used for home or office networks).  So the number of organisations that can have class A address is  =126-2-1=125  Number of hosts in each network  = 224 - 2 = 16, 277, 216 - 2 =16, 277, 214  (-2 is for network and broadcast id) |
| **Class-B** | 128.X.X.X  to  191.X.X.X | N.N.H.H | Network bits = two octet = 16 bits  Host bits = two octet = 12 bits  Number of netwoks with 16 bits where two bits are fixed for class B  = 216-2 = 214 =16, 384  But 16 blocks or netids (**172.16**.0.0 to **172.31**.255.255) are reserved for private addresses.  So the number of organisations that can have class B address is  = 16, 384 - 16 = 16, 368  Number of hosts in each network=  = 216 - 2 = 65, 536 - 2 = 65, 534  (-2 is for network and broadcast id) |
| **Class-C** | 192.X.X.X  to  223.X.X.X | N.N.N.H | Network bits = three octet = 24 bits  Host bits = one octet = 8 bits  Number of netwoks with 24 bits where three bits are fixed for class C  = 224-3 = 221 = 2, 097, 152  But 256 blocks or netids (**192.168.0**.0 to **192.168.255**.255) are reserved for private addresses.  So the number of organisations that can have class C address is  = 2, 097, 152 - 256 = 2, 096, 896  Number of hosts in each network=  = 28 - 2 = 256 - 2 = 254  (-2 is for network and broadcast id) |
| **Class-D** | 224.X.X.X  to  239.X.X.X | Multicasting | |
| **Class-E** | 240.X.X.X  to  255.X.X.X | Reserved for future | |

* 1. **Mask**
* A mask is a 32 bit number that is used to identify which part of an IPv4 address is the network part when bitwise ANDed with an address in the block.
* In the mask, all the bits of the network part are represented as "1" and all the bits of the host part are represented as "0".
* **Default mask**

|  |  |  |
| --- | --- | --- |
| **Class** | Mask in binary | Mask in dotted-decimal |
| **Class-A** | 11111111 00000000 00000000 00000000 | 255.0.0.0 |
| **Class-B** | 11111111 11111111 00000000 00000000 | 255.255.0.0 |
| **Class-C** | 11111111 11111111 11111111 00000000 | 255.255.255.255.0 |

* The network address is the beginning address of each block. It can be found by applying the default mask to any of the addresses inthe block (including itself). It retains the netid of the block and sets the hostid as zero.
  1. **Problems**
* **Problem-1:** Change the following IP addresses from binary notation to dotted decimal and hexa-decimal separately.

1. 10000001 00001011 00001011 11101111 **Ans:** 129.11.11.239
2. 11000001 10000011 00011011 11111111 **Ans:** 193.131.27.255
3. 11100111 11011011 10001011 01101111 **Ans:** 231.219.139.111
4. 11111001 10011011 11111011 00001111 **Ans:** 249.155.251.15

* **Problem-2:** Change the following IP addresses from dotted decimal notations to binary notation and hexa-decimal separately.

1. 111.56.45.78 **Ans:** 01101111 00111000 00101101 01001110
2. 221.34.7.82 **Ans:** 11011101 00100010 00000111 01010010
3. 241.8.56.12 **Ans:** 11110001 00001000 00111000 00001100
4. 75.45.34.78 **Ans:** 01001011 00101101 00100010 01001110

* **Problem-3:** Change the following IP addresses from hexa-decimal notation to dotted decimal notations and binary notation separately.

1. 0x1347FEAB
2. 0xAB234102
3. 0x0123A2BE
4. 0x00001111

* **Problem-4:** Find the error, if any in the following addresses.

1. 111.56.045.78 **Ans:** No leading zeros on dotted-decimal notation is allowed (045)
2. 221.34.7.8.20 **Ans:** No more than four parts in IP address is allowed.
3. 75.45.301.14 **Ans:** Each part must less than or equals to 255. 301 is not allowed.
4. 11100010.23.14.67 **Ans:** Mixture of binary and dotted decimal notation is not allowed.

* **Problem-5:** How can we prove that we have 2, 147, 483, 648 addresses in class A..

**Answer:**

* **Problem-6:** Find the class of each addresses given below.

1. 00000001 00001011 00001011 11101111 **Ans:** Class A
2. 11000001 10000011 00011011 11111111 **Ans:** Class C
3. 10100111 11011011 10001011 01101111 **Ans:** Class B
4. 11110011 10011011 11111011 00001111 **Ans:** Class E
5. 227.12.14.87 **Ans:** Class D
6. 193.14.56.22 **Ans:** Class C
7. 14.23.120.8 **Ans:** Class A
8. 252.5.15.111 **Ans:** Class E
9. 134.11.78.56 **Ans:** Class B

* **Problem-7:** How many digits are needed to define the netid in hexadecimal notation in each of the following classes.

1. Class A
2. Class B
3. Class C

* **Problem-8:** Find the class, the block and the range of addresses of the following network addresses.

1. 17.0.0.0
2. 132.21.0.0.
3. 23.56.7.91

* **Problem-9:** Find the beginning address (network address) of the following addresses.

1. 23.56.7.91
2. 132.6.17.85
3. 201.180.56.5
4. **Casting**

* Sending a packet from one host to other host is called casting.
* There are two types casting:

**Casting**

**Uni Casting**

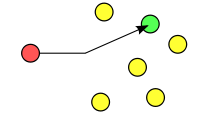
**Broad Casting**

**Limited Broad Casting (LBC)**

**Directed Broad Casting (DBC)**

1. **Unicasting**

* It is a method of transferring a message/packet from **one host to a particular host.**
* In computer networking, **unicas**t refers to a **one-to-one transmission** from one point in the network to another point; that is, one sender and one receiver, **each identified by a network address.**
* **Unicast addressing** uses a one-to-one association between a sender and destination: each destination address uniquely identifies a single receiver endpoint.



**(Unicast)**

**Host**

**25.1.2.3**

**(Sender IP)**

**Host**

**30.1.2.3**

**(Receiver IP)**

**Netwok Address**

**25.0.0.0**

**Netwok Address**

**30.0.0.0**

**M**

**SIP**

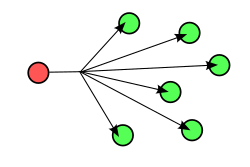
**RIP**

**Packet**

**M** - Message to be sent, **SIP** - Sender IP Address, **RIP** - Receiver IP Address

1. **Broadcasting**

* It is a method of transferring a message/packet from **one host to all hosts** simultaneously.
* In computer networking, telecommunication and information theory, broadcasting is a method of transferring a message to all recipients simultaneously.
* Broadcast uses a one-to-all association; a single datagram from one sender is routed to all of the possibly multiple endpoints associated with the broadcast address. The network automatically replicates datagrams as needed to reach all the recipients within the scope of the broadcast, which is generally an entire network subnet.



**(Broadcast)**

* The **local (limited) broadcast address** is 255.255.255.255. The traffic sent to that address will be sent to all the hosts on the local network, it is not a routable address, so a router would not route it anywhere.
* The **directed broadcast address** is sent to all hosts on a specific subnet, for example the directed broadcast address for the subnet 10.0.0.0/8 is 10.255.255.255. This address is routable, so a router would forward it to the end destination gateway if the router is configured to do so.

1. **Subnetting**
   1. **Definition**

* Dividing a bigger network into smaller ones logically is called subnetting.
* Maintenance
* Security
* f
  1. **Definition of Graph**
* A linked list is a way tF
* F
* F
* f

<http://www.pantz.org/software/tcpip/subnetchart.html>

<http://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php>

What is Subnet Mask?

An IPv4 address has two components, a "Network" part and a "Host" part. To identify which part of an IPv4 address is the "Network" part and which part of the IPv4 address is "Host" part, we need another identifier, which is known as "Subnet Mask". IPv4 address is a combination of IPv4 address and Subnet mask and the purpose of subnet mask is to identify which part of an IPv4 address is the network part and which part is the host part. Subnet mask is also a 32 bit number where all the bits of the network part are represented as "1" and all the bits of the host part are represented as "0".

If we take an example for a Class C network, 192.168.10.0, the address part and the subnet mask can be represented as below.

What is a Network Address?

A network address is used to identify the subnet that a host may be placed on and is used to represent that network. Network Address is the very first address of an IPv4 address block.

For Example, 10.0.0.0 is the network address of all IPv4 addresses starting from 10.0.0.1 to 10.255.255.254, having a subnet mask of 255.0.0.0

What is Limited Broadcast?

IPv4 Address 255.255.255.255 is used to send messages to all devices in the LAN and this IPv4 address is known as limited broadcast IPv4 address. A limited broadcast IPv4 Address can never be a source IPv4 address in an IPv4 datagram.

What is Directed Broadcast?

The host id value containing all 1's in the bit pattern indicates a directed broadcast address. A directed broadcast address can never be a source IPv4 address in an IPv4 datagram. A directed broadcast address will be seen by all nodes on that network. For example, the broadcast id for the network 192.168.10.0 with a subnet mask of 255.255.255.0 will be 192.168.10.255.

What is Default Network?

The IPv4 address of 0.0.0.0 is used for the default network. When a program sends a packet to an address that is not added in the on the computer's routing table, the packet is forwarded to the gateway for 0.0.0.0, which may able to route it to the correct address.

What are Loopback IPv4 Addresses?

IPv4 has a special reserved range of addresses known as IPv4 loopback addresses. Loopback range of IPv4 addresses ranges from 127.0.0.1 to 127.255.255.254. IP datagrams sent by a device to IPv4 loopback addresses not passed down to the data link layer for transmission to other devices. The IP datagrams sent to any address ranging from 127.0.0.1 to 127.255.255.254 are looped back to the source device at network layer.

If the TCP/IP protocol stack is working properly in your device, whenever you ping to any IPv4 loopback addresses, you will get a reply. Most of the operating systems map the IPv4 loopback address 127.0.0.1 with a name "localhost" by adding an entry in "hosts" file.

Automatic Private IPv4 addresses (APIPA)

Automatic Private IPv4 addresses (APIPA) are assigned to a device which is configured to automatically (dynamically) obtain an IPv4 address from a DHCP server, is not able to contact the DHCP server because of some network problem. APIPA addresses are under 169.254.0.0/16 range.

You have learned IPv4 addresses, different classes of IPv4 addresses, Class A IPv4 address, Class B IPv4 address, Class C IPv4 address, Class D IPv4 address, Class E IPv4 address, public IPv4 address, private IPv4 address, multicast IPv4 address (Class D IPv4 Addresses), Limited broadcast IPv4 address, direct broadcast IPv4 addresses, loopback IPv4 addresses and Automatic Private IPv4 addresses (APIPA).Visit the following link to know more about IPv4 private addresses (RFC 1918 addresses). Click next lesson to continue learning.

1. **Classless Inter Domain Routing (CIDR)**

**<http://www.wirelesstek.com/cidr.htm>**

**<https://whatismyipaddress.com/cidr>**

* 1. **Introduction**
* IANA - Internet Assign number Authority is the agency who have contol over IP addresses.
* If you have less number of machines (i.e.<=256), then you may purchase a class C network or block as any block or network of class C can give IP to max. 256 hosts/machines. If you need to connect more hosts then you will go for class B network.
* f
  1. **What is CIDR?**
* CIDR is a **new addressing scheme, classless addressing** **scheme** for the Internet which allows for more efficient allocation of IP addresses than the old Classfull A, B, and C addressing scheme.
* Classless Inter Domain Routing (CIDR) e Variable Length Subnet Mask (VLSM) were introduced to use the IP addressing space in a more efficient way, but also to improve the scalability for the global routing mechanism.
* Classless Inter-Domain Routing (CIDR) is a new method for **allocating IP addresses** and IP routing.
  1. **Why Do We Need CIDR?**
* The exponential growth of the Internet in the past 20 years highlighted shortcomings in the original IP protocol. CIDR and VLSM allow greater flexibility in the creation of sub-networks, overcoming the strict rules of the A-B-C Class mechanism
* With a new network being connected to the Internet every 30 minutes the Internet was faced with two critical problems:

- Running out of IP addresses

- Running out of capacity in the global routing tables

* Running Out of IP Addresses : There is a maximum number of networks and hosts that can be assigned unique addresses using the Internet's 32-bit long addresses. Traditionally, the Internet assigned "classes" of addresses: Class A, Class B and Class C were the most common. Each address had two parts: one part to identify a unique network and the second part to identify a unique host in that network. Another way the old Class A, B, and C addresses were identified was by looking at the first 8 bits of the address and converting it to its decimal equivalent.

|  |  |  |  |
| --- | --- | --- | --- |
| **Address Class** | **# Network Bits** | **# Hosts Bits** | **Decimal Address Range** |
| Class A | 8 bits | 24 bits | 1-126 |
| Class B | 16 bits | 16 bits | 128-191 |
| Class C | 24 bits | 8 bits | 192-223 |

* Using the old Class A, B, and C addressing scheme the Internet could support the following:

- 126 Class A networks that could include up to 16,777,214 hosts each

- Plus 65,000 Class B networks that could include up to 65,534 hosts each

- Plus over 2 million Class C networks that could include up to 254 hosts each

* (Some addresses are reserved for broadcast messages, etc.). Because Internet addresses were generally only assigned in these three sizes, there was a lot of wasted addresses. For example, if you needed 100 addresses you would be assigned the smallest address (Class C), but that still meant 154 unused addresses. The overall result was that while the Internet was running out of unassigned addresses, only 3% of the assigned addresses were actually being used. CIDR was developed to be a much more efficient method of assigning addresses.
  1. **Global Routing Tables At Capacity**
* A related problem was the sheer size of the Internet global routing tables. As the number of networks on the Internet increased, so did the number of routes. A few years back it was forecasted that the global backbone Internet routers were fast approaching their limit on the number of routes they could support.
* Even using the latest router technology, the maximum theoretical routing table size is approximately 60,000 routing table entries. If nothing was done the global routing tables would have reached capacity by mid-1994 and all Internet growth would be halted.
  1. **How Were These Problems Solved?**
* Two solutions were developed and adopted by the global Internet community:

1.Restructuring IP address assignments to increase efficiency

2.Hierarchical routing aggregation to minimize route table entries

* 1. **CIDR Notation**
* In CIDR notation, an IP is represented as

a.b.c.d/n

Where,

n represents the number of bits are used for the block id or netid.

Example

20.30.50.7/20 means NID=20 bits HID=32-20=12 bits

* According to the CIDR standard, the first part of an IP address is a prefix, which identifies the network. The prefix is followed by the host identifier so that information packets can be sent to particular computers within the network.
* In CIDR notation, IP addresses are written as a prefix, and a suffix is attached to indicate how many bits are in the entire address. The suffix is set apart from the prefix with a slash mark.
* For instance, in the CIDR notation 192.0.1.0/24, the prefix is 192.0.1.0, and the total number of bits in the address is 24.
* **CIDR Blocks**: The ability to group blocks of addresses into a single routing network is the hallmark of CIDR, and the prefix standard used for interpreting IP addresses makes this possible. CIDR blocks share the first part of the bit sequence that comprises the binary representation of the IP address, and blocks are identified using the same decimal-dot CIDR notation system that is used for IPv4 addresses. For example, 10.10.1.16/32 is an address prefix with 32 bits, which is the highest number of bits allowed in IPv4. Addresses with identical prefixes and the same number of bits always belong to the same block. In addition, larger blocks can be easily distinguished from smaller blocks by the length of the prefix. Short prefixes allow for more addresses while large prefixes identify small blocks.
  1. **Rules for CIDR block**
* All Ip addresses should be contiguous.
* Block size means how many hosts per block or network should be a power of 2. if k bits represents host bits, then block size should be 2k.
* The first IP address in the block should be evenly divisible by the size of the block.
  1. **Introduction**
* IANA - Internet Assign number Authority is the agency who have contol over IP addresses.
* If you have less number of machines (i.e.<=256), then you may purchase a class C network or block as any block or network of class C can give IP to max. 256 hosts/machines. If you need to connect more hosts then you will go for class B network.
* f
  1. **What is CIDR?**
* CIDR is a **new addressing scheme, classless addressing** **scheme** for the Internet which allows for more efficient allocation of IP addresses than the old Classfull A, B, and C addressing scheme.

1. **The IP datagram structure**

* Data transmitted over an internet using IP is carried in messages called IP datagrams.
* In network layer, IP uses a specific format for its datagrams.
* The IPv4 datagram is conceptually **divided into two pieces**: the **header** and the **payload**. The **header contains addressing and control fields,** while the **payload carries the actual data** to be sent over the internetwork.
* Unlike some message formats, IP datagrams do not have a footer following the payload.

|  |  |
| --- | --- |
|  |  |
| **Header** | **Data** |

* The IP header format is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **32 bits** | | | | |
| **Version**  **(4 bits)** | **IHL**  **(4 bits)** | **ToS**  **(8 bits)** | **Total length**  **(16 bits)** | |
| **Identification**  **(16 bits)** | | | **Flags**  **(3 bits)** | **Fragment Offset**  **(13 bits)** |
| **TTL**  **(8 bits)** | | **Protocol**  **(8 bits)** | **Header Checksum**  **(16 bits)** | |
| **Source Address**  **(32 bits)** | | | | |
| **Destination Address**  **(32 bits)** | | | | |
| **Options & Padding** | | | | |
|  | | | | |

* The IP header format is as follows:

1. **Version (4 bits):** Itis the IP version number (currently binary 0100 (IPV4), but can now also be version 6). All nodes must use the same version.
2. **IHL (4bits)** : **Internet Header Length;** It is the length of entire IP header as a number of 32-bit words. The minimum value of IHL is 5 (5 32-bit words = 5\*4 = 20 bytes), as that is the minimum size of an IP header that contains all the correct fields is 160 bits, or 20 bytes. This allows the receiver to know exactly where the payload data begins.
3. **ToS (8 bits) : Types of Service;** It contains a 3-bit precedence field (that is ignored today), 4 service bits, and 1 unused bit. This field is used to determine the type of service that must be provided by the Network/Internet layer depending on the type of application for which the data transfer needs to be done. The types of services that can be provided by IP are maximizing reliability and throughput and minimizing cost and delay. The size of the ToS component is 4 bits. The Transport layer provides the value of this field to the Internet layer. However, the values in these bits are just guidelines and not rules. The devices that operate from the Internet layer use these values to transfer data. This is a "hint" of what characteristics of the physical layer to use. The values that can be assigned to this component and a description of each value is provided below:

|  |  |
| --- | --- |
| **Binary Value** | **Description** |
| 0000 | Normal |
| 0001 | Minimizing cost |
| 0010 | Maximize reliability |
| 0100 | Maximize throughput |
| 1000 | Minimize delay |

For example, if the application is an Online Transaction Processing (OLTP) system, it will require the delay to be minimized and therefore the value would be 10002. However, in a situation where a bulk transfer of data is to be done, maximizing throughput will be appreciated and thus the value will be 01002.

1. **Total length (16 bits)** : It determine thelength of entire IP Packet (including IP header and IP Payload). The maximum size that an IP datagram can be is 65,535 bytes.

**Total length of the datagram = Length of the header + Length of the data**

If a datagram can be accommodated in a frame, data transmission becomes very simple. However, if the size of the datagram is more than the value that can be accommodated in the frame, the datagram must be divided into logical groups called fragments. In few cases, the size of a datagram will be much less than the size of the data that can be passed over the physical medium at one point in time. In that case, padding is done to fill in extra spaces. To find the exact length of the data that is sent over the frame, the Total Length field is used.

1. **Identification (16 bits) :** If IP packet is fragmented during the transmission, all the fragments contain same identification number. to identify original IP packet they belong to. These values allow datagrams to be fragmented for transmission and reassembled at the destination. The Identification is a unique number assigned to a datagram fragment to help in the reassembly of fragmented datagrams.
2. **Flags (3 bits)** : Bit 0 is always 0 and is reserved. Bit 1 indicates whether a datagram can be fragmented (0) or not (1). Bit 2 indicates to the receiving unit whether the fragment is the last one in the datagram (1) or if there are still more fragments to come (0).
3. **Fragment Offset (13 bits) :** This offset tells the exact position of the fragment in the original IP Packet. When a datagram is fragmented, it is necessary to reassemble the fragments in the correct order. The fragment offset numbers the fragments in such a way that they can be reassembled correctly.
4. **TTL (8 bits) : Time to Live;** This field determines how long a datagram will exist. At each hop along a network path, the datagram is opened and it's time to live field is decremented by one (or more than one in some cases). When the time to live field reaches zero, the datagram is said to have 'expired' and is discarded, then and an ICMP message is sent to the source host. This prevents congestion on the network that is created when a datagram cannot be forwarded to it's destination. Most applications set the time to live field to 30 or 32 by default. Here, this time supposed to be interpreted as a number of seconds but more often treated as a "hop count". It usually set to 32 or 64.
5. **Protocol (8 bits) :** It tells IP where to send the datagram up to. 6 means TCP, 17 means UDP. This field indicates what type of higher layer transport layer protocol is carried out in IP datagram and what type of network layer protocols, encapsulated within the IP datagram. Protocols are identified by a unique number as listed in an online database at [www.iana.org.](http://www.iana.org.) It 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 is 17. Some of the common values seen in this field include:

|  |  |
| --- | --- |
| **Value (Decimal)** | **Protocol** |
| 1 | Internet Control Message Protocol (ICMP) |
| 2 | Internet Group Management Protocol (IGMP) |
| 3 | Gateway-to-Gateway Protocol (GGP) |
| 4 | Internet Protocol (IP) |
| 6 | Transmission Control Protocol (TCP) |
| 8 | Exterior Gateway Protocol (EGP) |
| 9 | Interior Gateway Protocol (IGP) |
| 17 | User Datagram Protocol (UDP) |
| 41 | Internet Protocol Version 6 (IPv6) |
| 86 | Dissimilar Gateway Protocol (DGP) |
| 88 | Interior Gateway Routing Protocol (IGRP) |
| 89 | Open Shortest Path First (OSPF) |

1. **Header Checksum (16 bits) :** The checksum allows IP to detect datagrams with corrupted headers and discard them. Since the time to live field changes at each hop, the checksum must be re-calculated at each hop. In some cases, this is replaced with a cyclic redundancy check algorithm.
2. **Source Address (32 bits) :** This is the IP address of the sender of the IP datagram.
3. **Destination Address (32 bits) :** This is the IP address of the intended receiver(s) of the datagram. If the host portion of this address is set to all 1's, the datagram is an 'all hosts' broadcast.
4. **Options & Padding (variable): *Options is 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. ***Padding*** - there is padding added sometimes just to make sure that the datagram is confined within a 32 bit boundary in multiples of 32 bits.

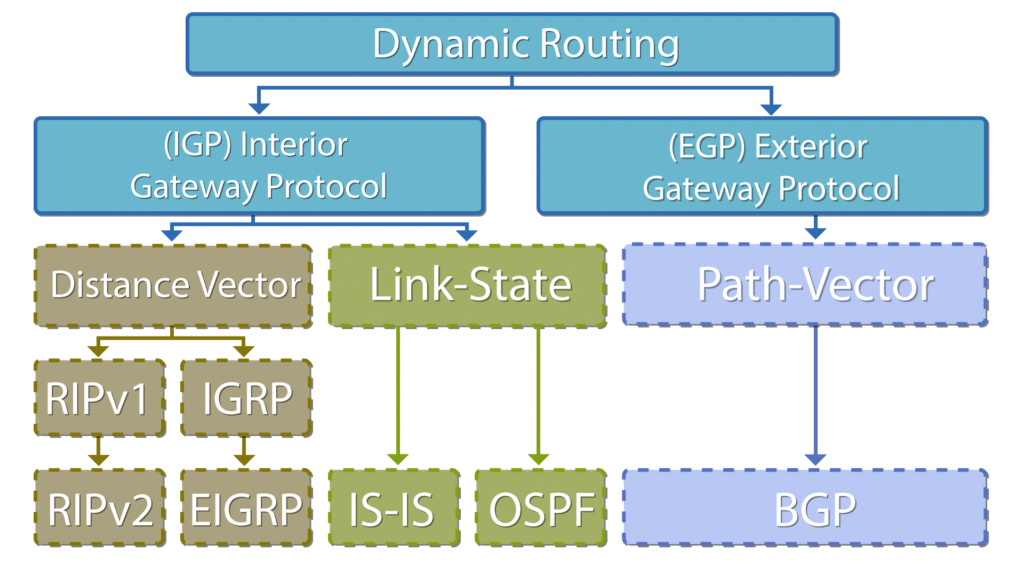
<http://mars.netanya.ac.il/~unesco/cdrom/booklet/HTML/NETWORKING/node020.html>

1. **Network Routing Protocols** 
   1. **Routing**

* **Routing is a process** in which the layer 3 devices (either router or layer 3 switch) finds the optimal route/path (a route/path which passes through the least number of nodes to reach the destination) to deliver a packet from source to destination.
* A **router** has **two processes** inside it.

1. **Forwarding:** one of the process handles each packet as it arrives, looking up the outgoing line to use for it in the routing table. This process is the forwarding.
2. **Filling and updating the routing table:** the other process is responsible for filling the routing table. That is where the routing algorithm comes into play. The routing algorithm is a part of network layer responsible for deciding which output line an incoming packet should be transmitted on.

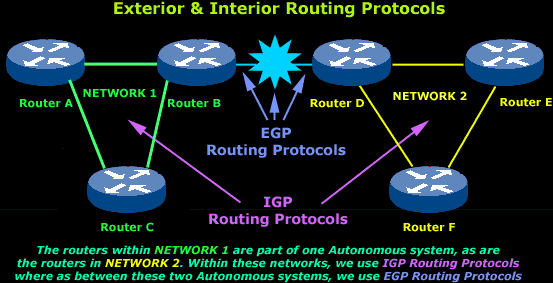
* To route IP packets, a host or a router has a routing table with entries for each destination or a combination of destinations.
* The main function of the network is to route the packets from source to destination. More than one route is possible in every network, however the shortest route should be selected.
* The shortest route means, a route which passes through the least number of nodes to reach the destination.
* The routing algorithm is designed to find the shortest root and it is part of a network software.
  1. **Classification of Routing Protocols**
* The router learns about remote networks from neighbor routers or from an administrator. The router then builds a routing table. If the network is directly connected then the router already knows how to get to the network. If the networks are not attached, the router must learn how to get to the remote network with either static routing (administrator manualy enters the routes in the router's table) or dynamic routing (happens automaticlly using routing protocols).
* The routers then update each other about all the networks they know. If a change occurs e.g a router goes down, the dynamic routing protocols automatically inform all routers about the change. If static routing is used, then the administrator has to update all changes into all routers and therefore no routing protocol is used.
* Only Dynamic routing uses routing protocols, which enable routers to:
* Dynamically discover and maintain routes
* Calculate routes
* Distribute routing updates to other routers
* Reach agreement with other routers about the network topology
* Statically programmed routers are unable to discover routes, or send routing information to other routers. They send data over routes defined by the network Administrator.
* The routing protocols can be classified into two types:
* **Static (non-adaptive) routing protocols**
* In this type, the network topology determines the final path. All the possible routing paths are already computed in advance , offline and are loaded into the routing table by the administrator (or when it is booted).
* Static routing is suitable for small networks.
* The disadvantage of static routing is, inability to respond quickly in case of network failure. It does not respond to failures.
* **Dynamic (Adaptive) routing protocols**
* The dynamic routing protocols, which enable routers that can change their routing decision on the basis of some changes made in the topology.
* Each router can check the network status by communicating with the neighbors. So, the changes in the topology are reflected to all routers.
* Finally, the router can calculate the suitable path to the final destination.
* The disadvantage of this type is, its complexity in the router.



* **RIP** : Routing Information Protocol
* **IGRP** : Interior Gateway Routing Protocol
* **IS-IS** : Intermediate System to Intermediate System
* **OSPF** : Open Shortest Path First
* **BGP** : Border Gateway Protocol
* **Difference** Between Interior gateway protocol (RGP) or Intra-domain Routing Protocols and Exterior gateway protocol (EGP) or Inter-domain Routing Protocol

|  |  |
| --- | --- |
| **Interior gateway protocol (RGP) or Intra-domain Routing Protocol** | **Exterior gateway protocol (EGP) or Inter-domain Routing Protocol** |
| 1. Routing takes place within an autonomous network system. 2. This protocol ignores the internet outside the autonomous system. | 1. Routing takes place between the two autonomous network system. 2. This protocol assumes that internet consists of a collection of interconnected autonomous systems. |
| **Examples:**  Distance Vector Routing Protocols: RIP, IGRP  Link State routing protocols: OSPF, IS-IS | **Examples:**  Path Vector Routing Protocols BGP: |

N.B.: Autonomous System.(AS) An autonomous system is a collection of networks or routers working under single administration.



* 1. **Distance Vector Routing (DVR) Protocol**
* It sees an AS with all the routers and networks as a graph.
* In DVR framework, each node maintains a routing table called distance vector. The table contains three information:
* Destination Node
* Estimated cost to destination
* Next hop via which to reach destination.
* Every node sends a message to its directly connected neighbors. Each node exchanges routing table info with all its neighbors.
* **Action at a router : Bellman-Ford equation**

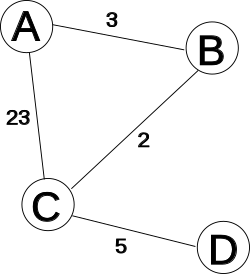
|  |
| --- |
| When a node x receives new DV estimate from any neighbor v, it saves v’s distance vector and it updates its own DV using B-F equation:  Dx(y) = min { C(x,v) + Dv(y)} for each node y ∈ N  Where,  Dx(y) = Estimate of least cost from node x to node y  C(x,v) = Node x knows cost to each neighbor v  Dx = {Dx(y): y ∈ N } = Node x maintains distance vector |

* Distance vector routing is the dynamic routing algorithm and also known as Bellman-Ford routing algorithm and Ford- Fulkerson algorithm.
* In this algorithm, node router constructs a table containing the distance (total cost of 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.

<https://en.wikipedia.org/wiki/Distance-vector_routing_protocol>

Example:

In this network we have 4 routers A, B, C and D:

[](https://en.wikipedia.org/wiki/File:Networkabcd.svg)

We mark the current time (or iteration) in the algorithm with T, and begin (at time 0, or T=0) by creating distance matrices for each router to its immediate neighbors. As we build the routing tables below, the **shortest path is highlighted in green,** and a **new shortest path is highlighted in yellow.** Grey columns indicate nodes that are not neighbors of the current node, and are therefore not considered as a valid direction in its table. Red indicates invalid entries in the table since they refer to distances from a node to itself, or via itself.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T=1 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from A** | **via A** | **via B** | **via C** | **via D** | | **to A** |  |  |  |  | | **to B** |  | 3 |  |  | | **to C** |  |  | 23 |  | | **to D** |  |  |  |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from B** | **via A** | **via B** | **via C** | **via D** | | **to A** | 3 |  |  |  | | **to B** |  |  |  |  | | **to C** |  |  | 2 |  | | **to D** |  |  |  |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from C** | **via A** | **via B** | **via C** | **via D** | | **to A** | 23 |  |  |  | | **to B** |  | 2 |  |  | | **to C** |  |  |  |  | | **to D** |  |  |  | 5 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from D** | **via A** | **via B** | **via C** | **via D** | | **to A** |  |  |  |  | | **to B** |  |  |  |  | | **to C** |  |  | 5 |  | | **to D** |  |  |  |  | |

At this point, all the routers (A,B,C,D) have new "shortest-paths" for their DV (the list of distances that are from them to another router via a neighbor). They each broadcast this new DV to all their neighbors: A to B and C, B to C and A, C to A, B, and D, and D to C. As each of these neighbors receives this information, they now recalculate the shortest path using it.

For example: A receives a DV from C that tells A there is a path via C to D, with a distance (or cost) of 5. Since the current "shortest-path" to C is 23, then A knows it has a path to D that costs 23+5=28. As there are no other shorter paths that A knows about, it puts this as its current estimate for the shortest-path from itself (A) to D, via C.

|  |
| --- |
|  |
| T=2 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from A** | **via A** | **via B** | **via C** | **via D** | | **to A** |  |  |  |  | | **to B** |  | 3 | 25 |  | | **to C** |  | 5 | 23 |  | | **to D** |  |  | 28 |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from B** | **via A** | **via B** | **via C** | **via D** | | **to A** | 3 |  | 25 |  | | **to B** |  |  |  |  | | **to C** | 26 |  | 2 |  | | **to D** |  |  | 7 |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from C** | **via A** | **via B** | **via C** | **via D** | | **to A** | 23 | 5 |  |  | | **to B** | 26 | 2 |  |  | | **to C** |  |  |  |  | | **to D** |  |  |  | 5 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from D** | **via A** | **via B** | **via C** | **via D** | | **to A** |  |  | 28 |  | | **to B** |  |  | 7 |  | | **to C** |  |  | 5 |  | | **to D** |  |  |  |  | |

Again, all the routers have gained in the last iteration (at T=1) new "shortest-paths", so they all broadcast their DVs to their neighbors; This prompts each neighbor to re-calculate their shortest distances again.

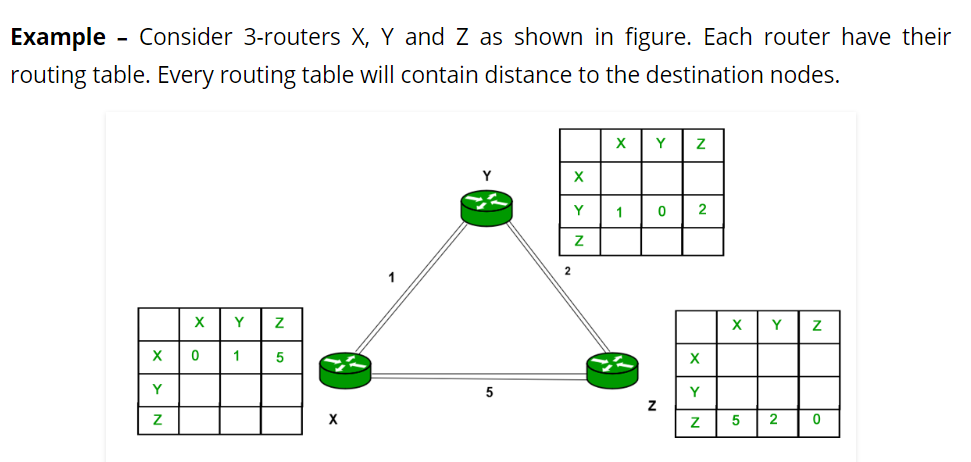
For instance: A receives a DV from B that tells A there is a path via C to D, with a distance (or cost) of 7. Since the current "shortest-path" to B is 3, then A knows it has a path to D that costs 7+3=10. This path to D of length 10 (via B) is shorter than the existing "shortest-path" to D of length 28 (via C), so it becomes the new "shortest-path" to D.

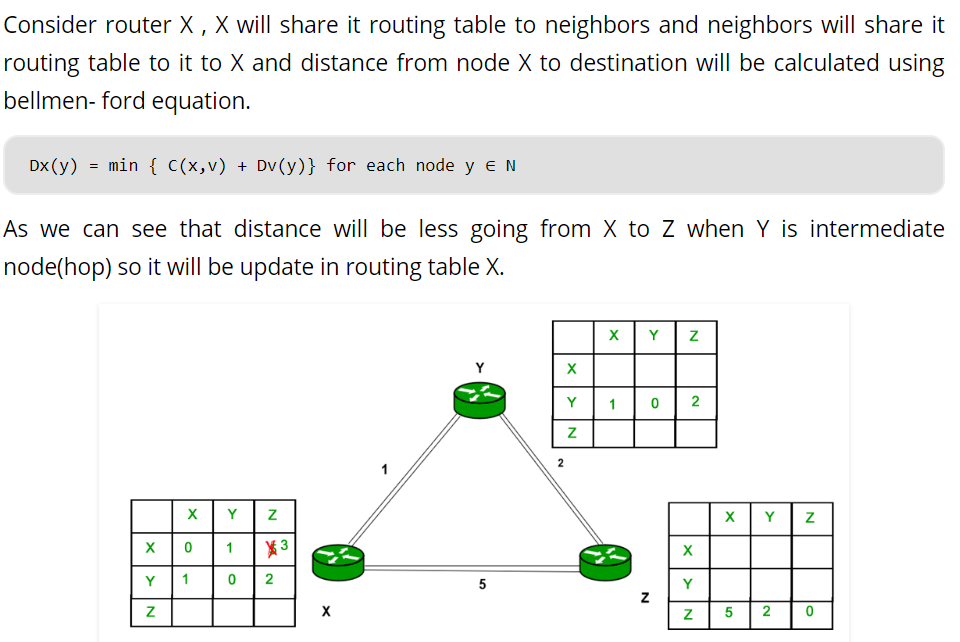
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T=3 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from A** | **via A** | **via B** | **via C** | **via D** | | **to A** |  |  |  |  | | **to B** |  | 3 | 25 |  | | **to C** |  | 5 | 23 |  | | **to D** |  | 10 | 28 |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from B** | **via A** | **via B** | **via C** | **via D** | | **to A** | 3 |  | 7 |  | | **to B** |  |  |  |  | | **to C** | 8 |  | 2 |  | | **to D** | 31 |  | 7 |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from C** | **via A** | **via B** | **via C** | **via D** | | **to A** | 23 | 5 |  | 33 | | **to B** | 26 | 2 |  | 12 | | **to C** |  |  |  |  | | **to D** | 51 | 9 |  | 5 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from D** | **via A** | **via B** | **via C** | **via D** | | **to A** |  |  | 10 |  | | **to B** |  |  | 7 |  | | **to C** |  |  | 5 |  | | **to D** |  |  |  |  | |  |

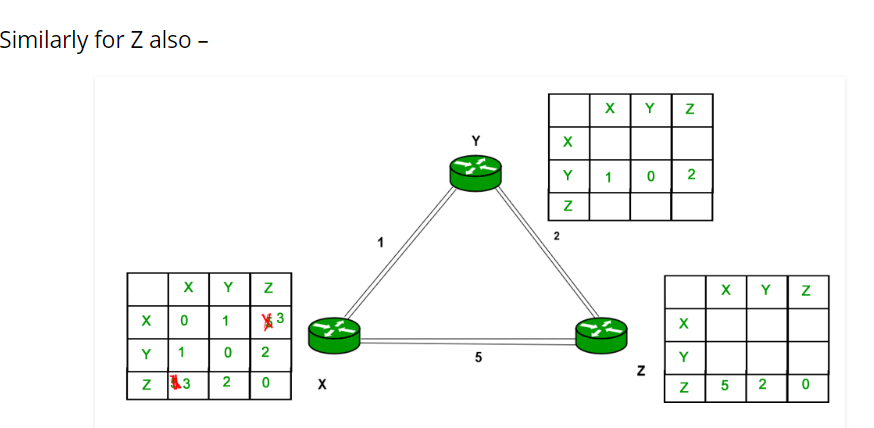
This time, only routers A and D have new shortest-paths for their DVs. So they broadcast their new DVs to their neighbors: A broadcasts to B and C, and D broadcasts to C. This causes each of the neighbors receiving the new DVs to re-calculate their shortest paths. However, since the information from the DVs doesn't yield any shorter paths than they already have in their routing tables, then there are no changes to the routing tables.

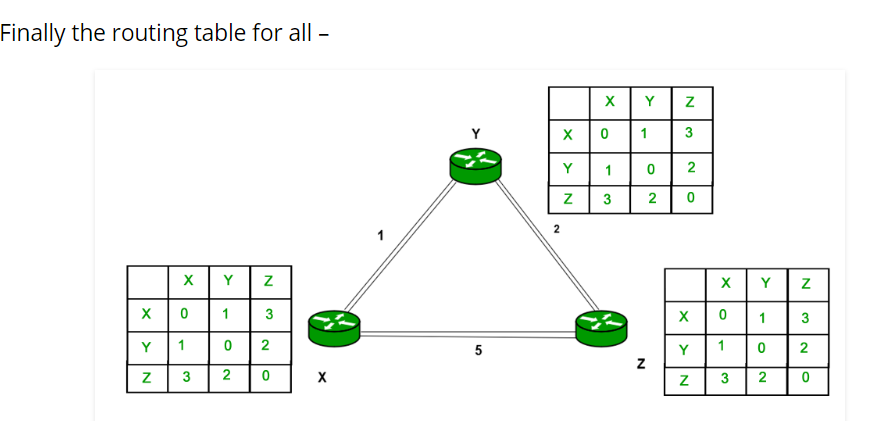
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T=4 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from A** | **via A** | **via B** | **via C** | **via D** | | **to A** |  |  |  |  | | **to B** |  | 3 | 25 |  | | **to C** |  | 5 | 23 |  | | **to D** |  | 10 | 28 |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from B** | **via A** | **via B** | **via C** | **via D** | | **to A** | 3 |  | 7 |  | | **to B** |  |  |  |  | | **to C** | 8 |  | 2 |  | | **to D** | 13 |  | 7 |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from C** | **via A** | **via B** | **via C** | **via D** | | **to A** | 23 | 5 |  | 15 | | **to B** | 26 | 2 |  | 12 | | **to C** |  |  |  |  | | **to D** | 33 | 9 |  | 5 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **from D** | **via A** | **via B** | **via C** | **via D** | | **to A** |  |  | 10 |  | | **to B** |  |  | 7 |  | | **to C** |  |  | 5 |  | | **to D** |  |  |  |  | |  |

None of the routers have any new shortest-paths to broadcast. Therefore, none of the routers receive any new information that might change their routing tables. The algorithm comes to a stop.

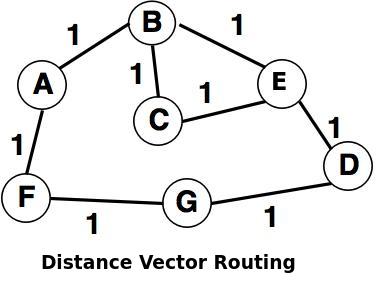




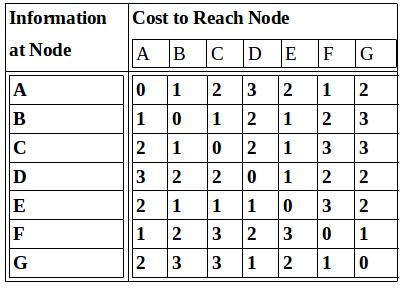




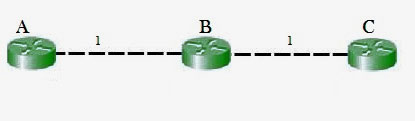
* **Advantages of Distance Vector routing –**
* It is simpler to configure and maintain than link state routing.
* Easy routing table reduction: Distance vector algorithm makes it easy to reduce the routing table as it makes use of less information compared to Link state Algorithm.
* Simpler to use: easy to implement the distance vector routing algorithm.
* **Dis-advantages of Distance Vector routing –**
* It is slower to converge than link state.
* It is at risk from the count-to-infinity problem.
* It creates more traffic than link state since a hop count change must be propagated to all routers and processed on each router. Hop count updates take place on a periodic basis, even if there are no changes in the network topology, so bandwidth-wasting broadcasts still occur.
* For larger networks, distance vector routing results in larger routing tables than link state since each router must know about all other routers. This can also lead to congestion on WAN links.
* Note – Distance Vector routing uses UDP (User datagram protocol) for transportation.
* **Slow convergence problem:** Changes are reflected slowly across the network as all the routing tables must be recalculated.
* **Unsuitable for large inter networks:** In large networks the routing tables could increase in size making them difficult to manage and unsuitable for large networks
* A link, which is 'down' (which is not working) is assigned as an infinite cost.
* 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).



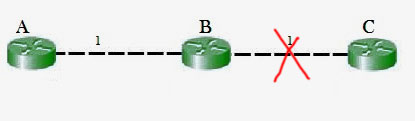
The shortest path can be computed as:



* Some issues with the Distance Vector Routing are:
* Vulnerability to the 'Count-to-Infinity' problem is a serious issue with the distance vector.
* It takes long time for convergence due to growth in the network.
  1. **Routing Loops or Count-to-Infinity Problem**
* The main issue with Distance Vector Routing (DVR) protocols is **Routing Loops**, since Bellman-Ford Algorithm cannot prevent loops. This routing loop in DVR network causes Count to Infinity Problem. **Routing loops usually occur when any interface goes down or two-routers send updates at the same time.**
* The count-to-infinity problem happens when a router is unable to reach an adjoining network. A second router, 1 hop away from the first router, thinks that the unreachable network is 2 hops away. Meanwhile, the first router then updates its records to say it is 3 hops away from the unreachable network based on the fact it is 1 hop from the second router, which says it is 2 hops from the unreachable network. The routers continue incrementing their hop count until the maximum (15), "infinity", is reached.
* There are three methods of preventing this problem: Split Horizon, Split Horizon with Poison Reverse, and Triggered updates.
* **Example:**



So in this example, the Bellman-Ford algorithm will converge for each router, they will have entries for each other. B will know that it can get to C at a cost of 1, and A will know that it can get to C via B at a cost of 2.

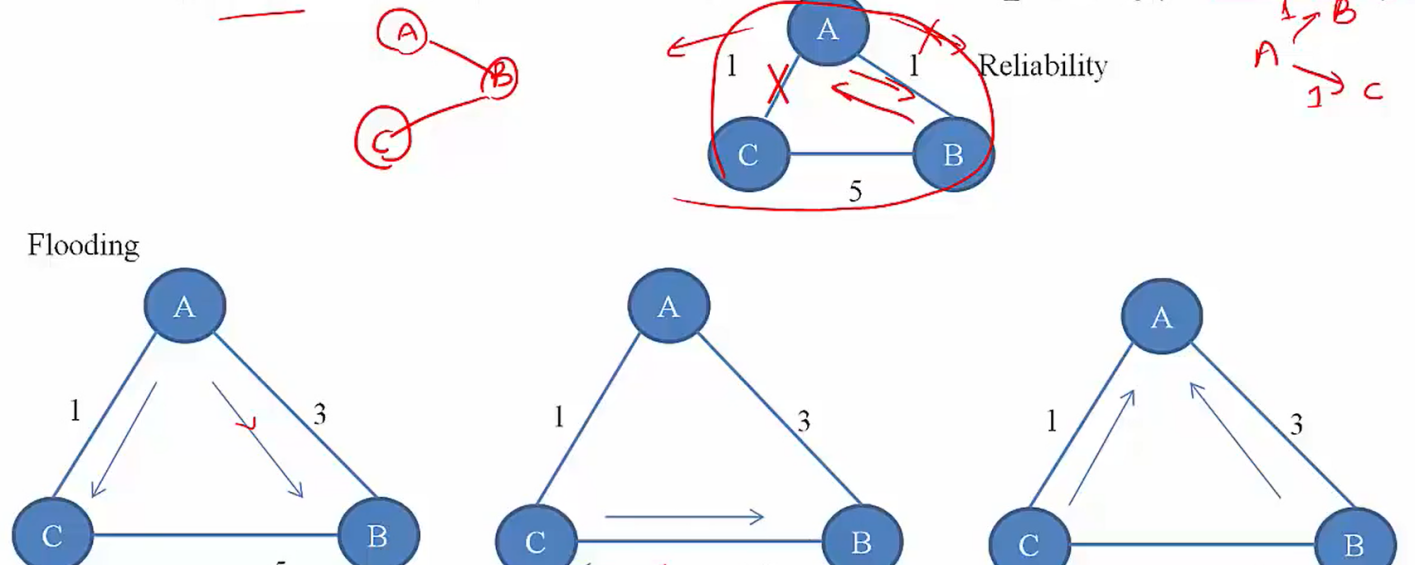


If the link between B and C is disconnected, then B will know that it can no longer get to C via that link and will remove it from it’s table. Before it can send any updates it’s possible that it will receive an update from A which will be advertising that it can get to C at a cost of 2. B can get to A at a cost of 1, so it will update a route to C via A at a cost of 3. A will then receive updates from B later and update its cost to 4. They will then go on feeding each other bad information toward infinity which is called as Count to Infinity problem.

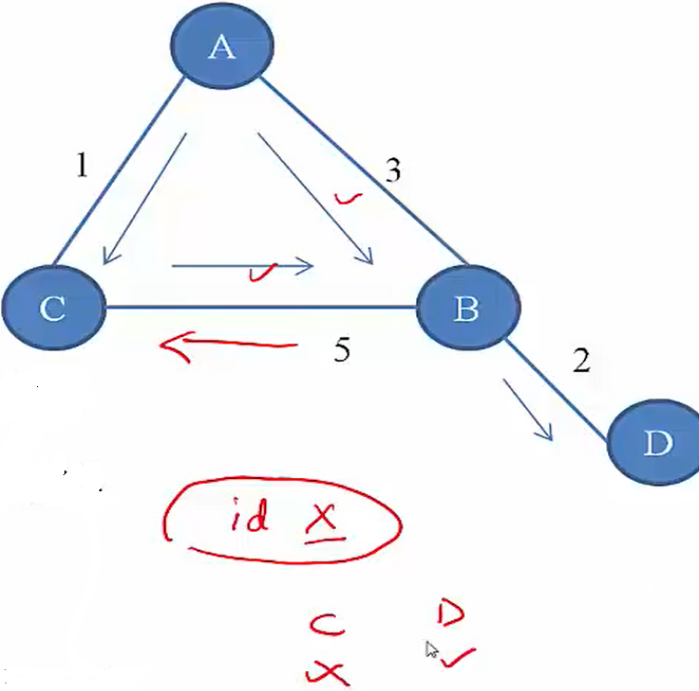
* **Solutions to Count-to-Infinity Problem**
* Defining Infinity or Route Poisoning: : SIP V-1 =>16:
* **Advantages of Distance Vector routing –**
* It is simpler to configure and maintain than link state routing.

1. **Link-State Routing**

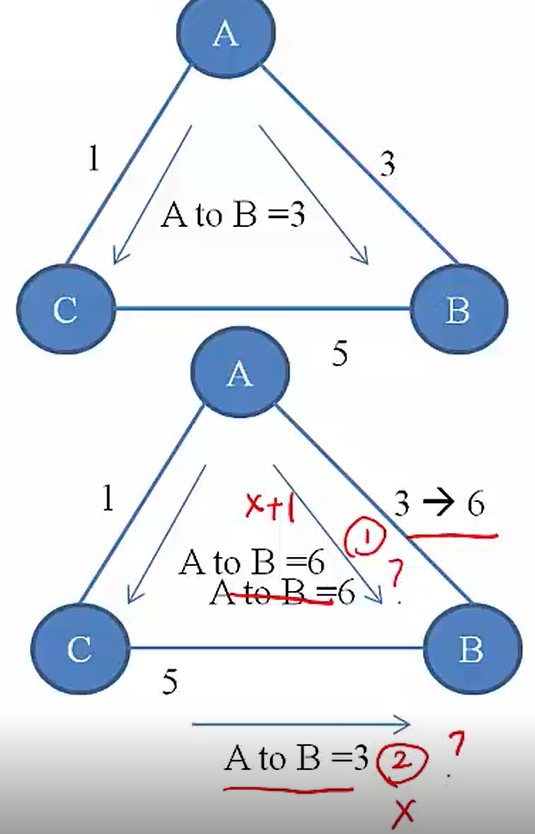
* The goal of routing is to find out the shortest path between nodes.
* The basic idea of link state routing is simple. There are **two phases** involved.
* **Phase-1: Reliable flooding:**
* **Initial State:** Each node knows the cost to its neighbors.
* **Final State:** Each node knows the entire graph (network topology)
* **Pase-2: Route calculation:**
* Each node uses Dijkstra’s algorithm on the graph to calculate optimal routes to all nodes.
* The phase-2 is simple. The challenge is on phase-1. How can one go from initial state to final state? This can be achieved through reliable flooding.
  1. **Reliable Flooding**
* Each node sends its link-state (neighbourhood information) to all nodes of the topology reliably.



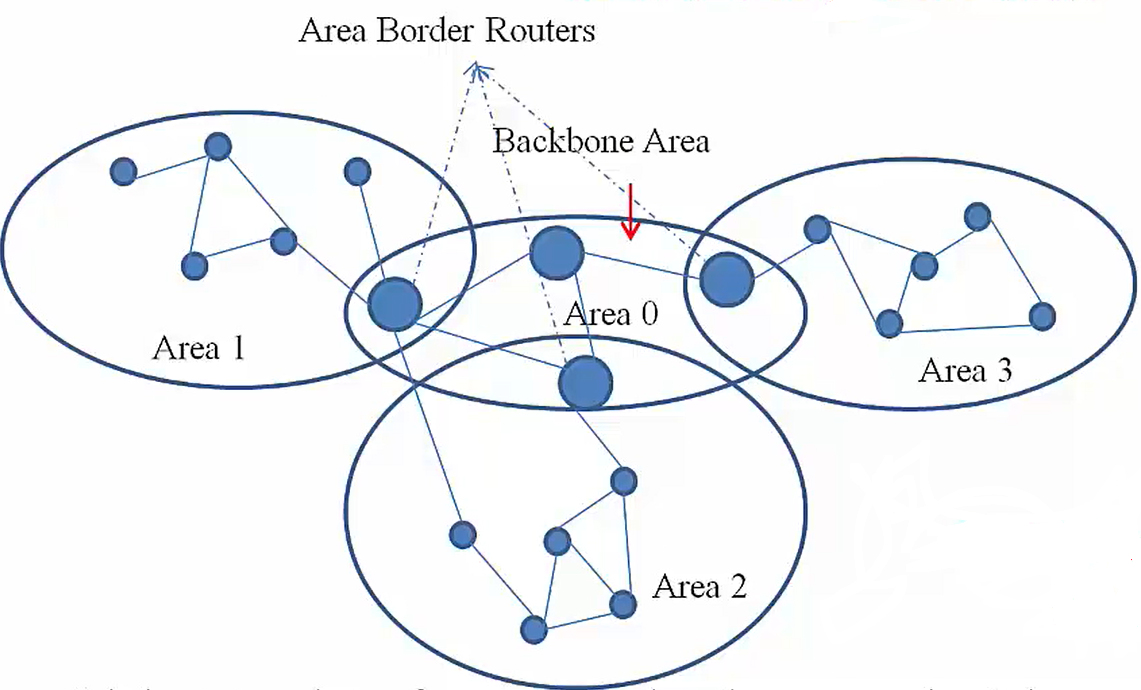
* Employ a reliable protocol to transfer information between neighbors.
* Avoid loops and minimize message exchanges. Need to detect duplicates.
* Packets need unique ids.
* For a given id, maintain state (send flags) to determine on which interface to send.



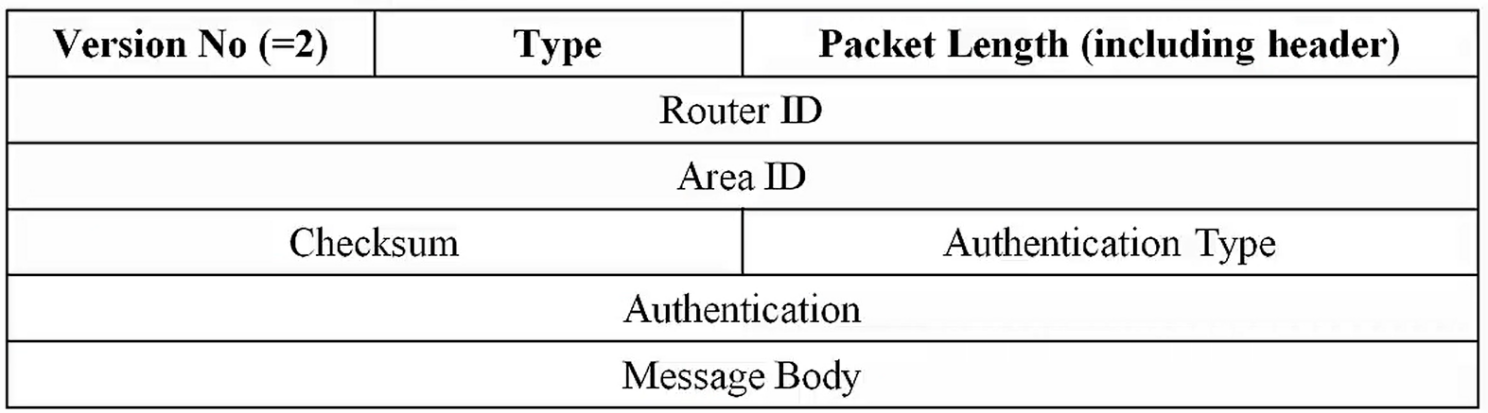
* New information should precede older information.
* Use sequence number (also uniquely identifies a packet)
* At a node, increment sequence number for each new message flooded.



* Corruption of sequence numbers
* Use checksums
* Each entry stored at anode is aged.
  1. **Link-State Protocol**
* From the protocol aspect, **what message** to send? => **LSP** (Link State packet)
* **What to do** when you receive an **LSP**? => **Action at a node**
* When to send LSP? => Updates
* **LSP (Link State packet)**
* The **packet id** of a node sending the packet. It is unique for each node.
* **Link-state of the node:** neighborhood information (list of neighbour and cost to each)
* **Sequence number:** Each packet contains a sequence number. This keep tracks how many times this packet has been sent.
* **Time to Live (TTL) or Age :** This keep track of life of a packet in a network.
* **Action at a node**
* Suppose a node X receives an LSPgenerated by node Y (Y need not be X’s neighbor)
* Did I (I.e.X) hear (receive the packet) from Y before?
* **NO:** Store the link state information. Start an ageing timer
* **YES:** Compare sequence number of this packet (Seq\_New) with stored information (Sqe\_Old). If Seq\_New>Seq\_Old, overwrite old link-state information, refresh ageing timer, forward to required neighbor. If Seq\_New<=Seq\_Old, discard received packet.
* **Updates**
* When to send the LSP information? As we know flooding leads to lots of traffic, we need to avoid flooding to the extent possible.
* **Triggered Updates:** A node floods the network whenever its link-state information changes.
* **Periodic Updates**: Need not be sent often, use long times (order of hours).
  1. **Route Calculation**
* Once a node has a LSP packet from every node, it has the complete graph information.
* Use Dijkstra’s algorithm to calculate shortest paths to nodes.
  1. **OSPF (Open Shortest Path First)**
* OSPF is a very widely used IGP or intra-domain routing protocol.
* It replaces more or less RIP as it really converges fast. RIP is operated in application layer and OSPF is operated in network layer. The LSP of OSPF is encapsulated within IP datagram with protocol number of 89.
* OSPF implements reliability itself via checksums and in-built ACKs.
* It has many features like it supports authentications (A router gets infroms from other routers if it authenticates other routers), additional hierarchy (I.e. areas) , load balancing.
* OSPF is a very complex type of protocol. It has many types of packets and lots of actions based on the packets.
* **High Level Overview:**
* Routing Areas: How OSPF handles scalability. Here entire domain is divided into areas.



* **Path of a packet**
* Source network to backbone area
* Cross the backbone area
* Backbone area to destination network.
* Link-State advertisement of a non-area boarder router don’t leave area.
* Area boarder router summerize area advertisemnts and advertise to other areas.
* **OSPF Common Header Format**



* **Version No.:** 2 represents IPV4 and 3 for IPV6.
* **Type :** There are many OSPF packets.

|  |  |
| --- | --- |
| **Type Value** | **Message Type** |
| 1 | Hello |
| 2 | Database Description |
| 3 | Link State Request |
| 4 | Link State Update |
| 5 | Link State Acknowledgement |

* **Packet Length:** Length of the OSPF packet
* **Router ID:** It is the generator of the OSPF packet. It is generally set the IP address of the router. As router has many interfaces, so many IPs. So the smallest IP add is set to it.
* **Area ID:** To handle the area area ID is set.
* **Checksum:**
* **Authentication Type:**

|  |  |
| --- | --- |
| **Authentication Type Value** | **Authentication Type** |
| 0 | No authentication |
| 1 | Simple Password authentication |
| 2 | Cryptography authentication |

* **Authentication:** This field carries some authentication information like password info.
* **Message Body:** This field varies depending upon the type of OSPF packet it is.
  1. **Distance Vector Vs. Link State**

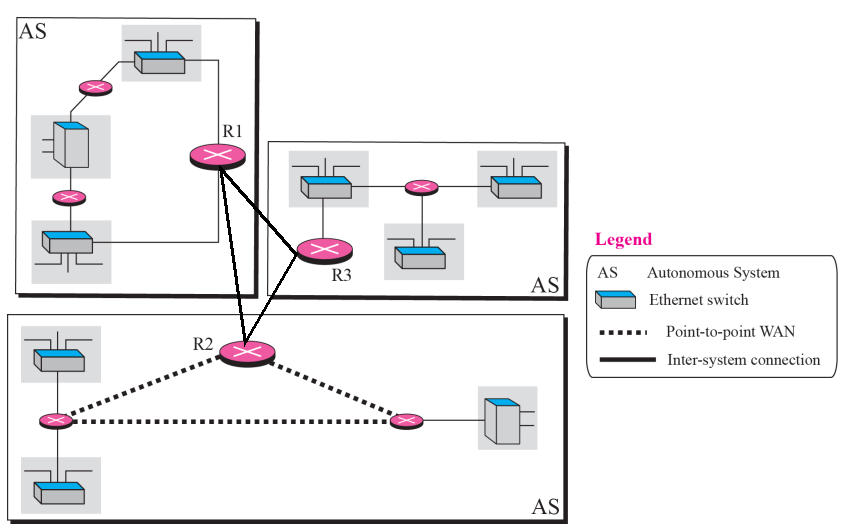
|  |  |
| --- | --- |
| **Distance Vector (DV)** | **Link State (LS)** |
| * Each node talks only with directly connected neighbors, but tells everything it has learned. (Sends the entire routing table) * Susceptible to network routing loops problem * Slow convergence * Donot know the network topology. * Easy to configure and administer. * Consumes a lot of Bandwidth. * Requires less memory and processing power of routers. * Has frequent and periodic updates. | * Each node talks to all nodes, but only state of directly connected nodes. (Sends only link-state information) * Less susceptible or no problem of network routing loops since each node has global information. * Fast convergence * Knows the entire network topology. * Difficult to configure and administer. * Consumes less BW than distance vector. * Requires more processing power and memory than distance vector. * Has event triggered updates. |

**Distance Vector.** Distance Vector routers compute the best path from information passed to them from neighbors

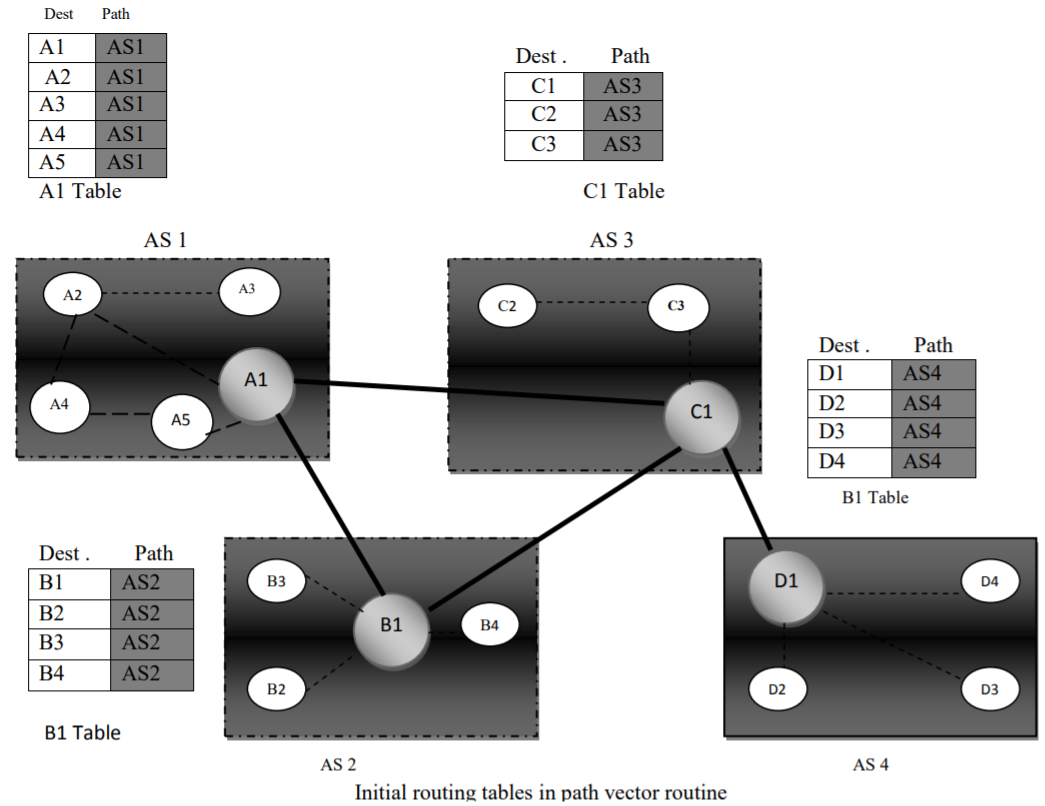
**Link State.** Link State routers each have a copy of the entire network map and compute best routes from this local map

1. **Exterior routing Protocol (or Inter Domain Routing)**

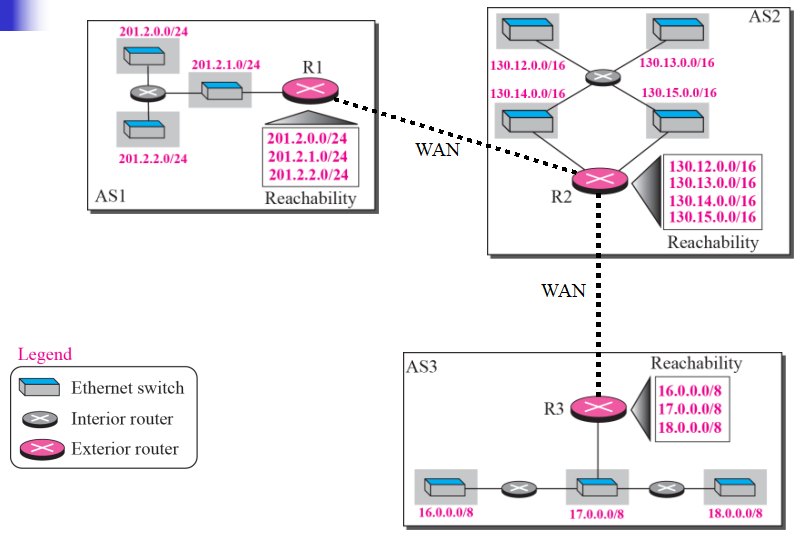
* An **internet** is a **combination of networks connected by routers**. When a datagram goes from a source to a destination, it will probably pass through many routers until it reaches the router attached to the destination network.
* Today, an **internet can be so large** that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems.
* **Autonomous system (AS)** is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is called **intra-domain routing.** Routing between autonomous systems is called **inter-domain routing.**



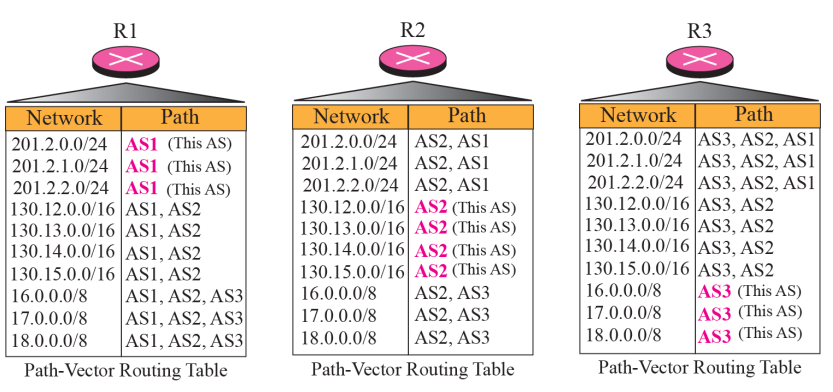
* Distance vector and link state routing are both interior routing protocols. They can be used inside an autonomous system. Both of these routing protocols become **intractable when the domain of operation becomes large**. Distance vector routing is subject to instability if there is more than a few 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.
* The difference between the distance vector routing and path vector routing can be compared to the difference between a national map and an international map. A national map can tell us the road to each city and the distance to be traveled if we choose a particular route; an international map can tell us which cities exist in each country and which countries should be passed before reaching that city.
  1. **Path Vector Routing**
* Path Vector Routing is a routing algorithm in unicast routing protocol of network layer, and it is useful for inter domain routing.
* The principle of path vector routing is similar to that of distance vector routing.
* It assumes that there is **one node in each autonomous system** that acts on behalf of the entire autonomous system is called **Speaker node**. The speaker node in an AS creates a routing cable and advertises to the speaker node in the neighbouring ASs. A speaker node advertises the path, not the metrics of the nodes, in its autonomous system or other autonomous systems.
* **Example-1**



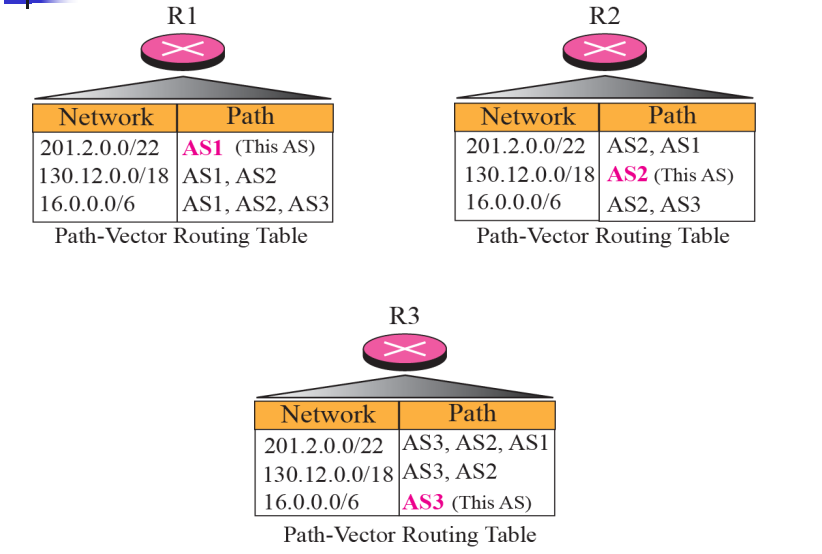
* **Example-2**



* **Stabilized table for three autonomous system**



* **Routing tables after aggregation**



* It is the initial table for each speaker node in a system made four ASs. Here Node A1 is the speaker node for AS1, B1 for AS2, C1 for AS3 and D1 for AS4, Node A1 creates an initial table that shows A1 to A5 and these are located in AS1, it can be reached through it. A speaker in an autonomous system shares its table with immediate neighbours ,here Node A1 share its table with nodes B1 and C1 , Node C1 share its table with nodes A1,B1 and D1 , Node B1 share its table with nodes A1 and C1 , Node D1 share its table with node C1 If router A1 receives a packet for nodes A3 , it knows that the path is in AS1,but if it receives a packet for D1,it knows that the packet should go from AS1,to AS2 and then to AS3 ,then the routing table shows that path completely on the other hand if the node D1 in AS4 receives a packet for node A2,it knows it should go through AS4,AS3,and AS1.
* **FUNCTIONS**
* **PREVENTION OF LOOP** : The creation of loop can be avoided in path vector routing. A router receives a message and checks to see if its AS is in the path list to the destination.
* **POLICY ROUTING:**When a router receives a messages it can check the path, if one of the autonomous system listed in the path against its policy, it can ignore its path and destination. It does not update its routing table with this path or it does not send the messages to its neighbors.
* **OPTIMUM PATH:** A path to a destination that is the best for the organization that runs the autonomous system. **Problem:** Each AS that is included in the path may use a different criteria for the metric

- The optimum path is the path that fits the organization

- Chose the one that had the smaller number of ASs

- Other criteria: security, safety, reliability, etc.

1. **Broader gateway Protocol (BGP)**

* **BGP** is an interdomain routing protocol using path vector routing. It first appeared in 1989 and has gone through four versions.
* **Extension of distance‐vector routing**

- Support flexible routing policies.

- Avoid count‐to‐infinity problem

* **Key idea:** **Path vector:** Advertise or send the entire path for each destination node.
* BGP interconnects **three different types of AS:**
* **Stub AS:** Only one connection to another AS (only a source or sink for data traffic), e.g. a corporate network
* **Multihomed AS:** More than one connection to other AS, but it is still only a source or sink for data traffic. e.g. a large corporate network with connections to multiple ASs, but does not allow traffic to pass thru (transient)
* **Transit AS** : Multihomed AS that also allows transient traffic, such as an Internet backbone.
* BGP supports classless addressing and CIDR.
* BGP uses the services of TCP on port 179.
* Path‐vector rou3ng

– Share entire path, not distance: faster convergence

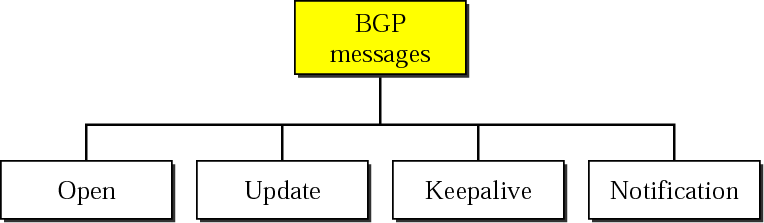
– More flexibility in selecting paths

* **BGP Functions**
* **Sharing:** A speaker in an AS shares its table with immediate neighbors.
* **Updating**: Adding the nodes that are not in its routing table and adding its own AS and the AS that sent the table. The routing table shows the path completely
* **Loop Prevention**: The creation of loop can be avoided in path vector routing. A router receives a message and checks to see if its AS is in the path list to the destination. Node can easily detect a loop.**Node can simply discard paths with loops.**
* **Policy Routing:**When a router receives a messages it can check the path, if one of the autonomous system listed in the path against its policy, it can ignore its path and destination. It does not update its routing table with this path or it does not send the messages to its neighbors.
* **Optimum Path:** A path to a destination that is the best for the organization that runs the autonomous system. **Problem:** Each AS that is included in the path may use a different criteria for the metric

- The optimum path is the path that fits the organization

- Chose the one that had the smaller number of ASs

- Other criteria: security, safety, reliability, etc.



* ssIn a network, flooding is the forwarding by a router of a packet from any node to every other node attached to the router except the node from which the packet arrived. Flooding is a way to distribute routing information updates quickly to every node in a large network. It is also sometimes used in multicast packets (from one source node to many specific nodes in a real or virtual network).

simple computer network routing algorithm in which every incoming packet is sent through every outgoing link except the one it arrived on.[1]

Flooding is used in bridging and in systems such as Usenet and peer-to-peer file sharing and as part of some routing protocols, including OSPF, DVMRP, and those used in ad-hoc wireless networks

There are several variants of flooding algorithms. Most work roughly as follows:

Each node acts as both a transmitter and a receiver.

Each node tries to forward every message to every one of its neighbors except the source node.

This results in every message eventually being delivered to all reachable parts of the network.

**Advantages**

**If a packet can be delivered, it will (probably multiple times).**

**Since flooding naturally utilizes every path through the network, it will also use the shortest path.**

**This algorithm is very simple to implement.**

**Disadvantages**

**Flooding can be costly in terms of wasted bandwidth. While a message may only have one destination it has to be sent to every host. In the case of a ping flood or a denial of service attack, it can be harmful to the reliability of a computer network.**

**Messages can become duplicated in the network further increasing the load on the networks bandwidth as well as requiring an increase in processing complexity to disregard duplicate messages.**

**Duplicate packets may circulate forever, unless certain precautions are taken:**

**Use a hop count or a time to live (TTL) count and include it with each packet. This value should take into account the number of nodes that a packet may have to pass through on the way to its destination.**

**Have each node keep track of every packet seen and only forward each packet once.**

**Enforce a network topology without loops.**

* 1. **Distance Vector Routing (DVR) Protocol**
* It sees an AS with all the routers and networks as a graph.
* In DVR framework, each node maintains a routing table called distance vector. The table contains three information:
* Destination Node
* Estimated cost to destination
* Next hop via which to reach destination.

****Distance vector routing:****

1. Distance Vector Routing is one of the dynamic routing algorithm.
2. It is suitable for packet switched network.
3. In distance vector routing, each router maintains a routing table.
4. It contains one entry for each router in the subnet.

This entry has two parts:

a. The first part shows the preferred outgoing line to be used to reach the destination.

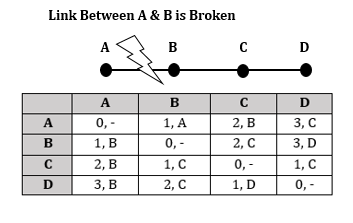
b. Second part gives an estimate of the time or distance to the destination.

1. In distance vector routing, a node tells its neighbor about its distance to every other node in the network.

****Count to infinity problem:****

1. One of the important issue in Distance Vector Routing is County of Infinity Problem.
2. Counting to infinity is just another name for a routing loop.
3. In distance vector routing, routing loops usually occur when an interface goes down.
4. It can also occur when two routers send updates to each other at the same time.

****Example:****



 Imagine a network with a graph as shown above in figure 4.8.

 As you see in this graph, there is only one link between A and the other parts of the network.

 Now imagine that the link between A and B is cut.

 At this time, B corrects its table.

 After a specific amount of time, routers exchange their tables, and so B receives C's routing table.

 Since C doesn't know what has happened to the link between A and B, it says that it has a link to A with the weight of 2 (1 for C to B, and 1 for B to A -- it doesn't know B has no link to A).

 B receives this table and thinks there is a separate link between C and A, so it corrects its table and changes infinity to 3 (1 for B to C, and 2 for C to A, as C said).

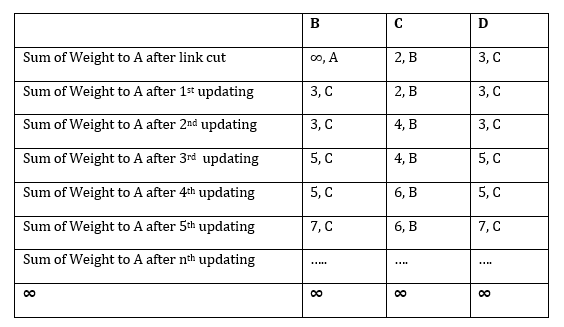
 Once again, routers exchange their tables.

 When C receives B's routing table, it sees that B has changed the weight of its link to A from 1 to 3, so C updates its table and changes the weight of the link to A to 4 (1 for C to B, and 3 for B to A, as B said).

 This process loops until all nodes find out that the weight of link to A is infinity.

 This situation is shown in the table below 4.2.

 In this way, Distance Vector Algorithms have a slow convergence rate.



 One way to solve this problem is for routers to send information only to the neighbors that are not exclusive links to the destination.

 For example, in this case, C shouldn't send any information to B about A, because B is the only way to A.

* 1. **Flooding**
  2. **Link State Routing Protocols**
* 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 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.
* 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 is arrived ie flooding) and discarded, if the packet is duplicate. If the sequence number is lower (than the highest one), it is rejected.
* **Some changes to improve the basic algorithm:**
* Once the router accumulates full set of link state packets, it can construct the entire subnet graph and Dijkshtra'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.
* Link state routing protocol is widely used in actual networking system.

**Difference between distance vector and link state routing protocols**

**Distance Vector.** Distance Vector routers compute the best path from information passed to them from neighbors

**Link State.** Link State routers each have a copy of the entire network map and compute best routes from this local map

IMG_256

* 1. **Routing**
* The interfaces or interconnecting devices are switche, Bridge, Router & gateway
* F
  1. **Delivery of a Packet**
* The network layer examines the handling of packets by the underlying networks. This handling is referred to as **delivery of a packet**.The delivery of packet (source to final destination) can be achieved by two methods:
* **Direct delivery**
* In this method, the source and destination of the packets are located on the same network.
* The sender can determine, if the delivery is direct. With the help of masking, the sender can extract the network address of the destination and compares this address with the addresses of the connected networks. If the match is found, then the delivery is direct.
* **Indirect delivery**
* In this method, the destination host is not on the same network as the deliverer. The packet is delivered directly.
* The packet moves from router to router until it reaches the same physical network as its final destination.

