OSI MODEL:

Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer

Application Layer : The Application layer is reponsible for providing an interface for the users to interact with application services of Network services
Ex: web browser etc..

i) Indentication of services is done using port numbers. Port is a logical communication channel port number is a 16 bit identification
   Total no of ports 0 – 65535
   Wellknown ports 0 – 1023
   Registered ports 1024 – 49151
   Epimeral ports 49152 – 65535

Ex : Internet

xyz.com we need type in webbrowsing tool internally it contains the http which allow to access the web pages.

Application layer is used to provide a interfaces

Protocol	TCP/UDP	Port Number
1) File Transfer protocol	TCP	20 - FTP data 21 - FTP control
2) Secure Shell (SSH)	TCP	22
3) Telnet	TCP	23
4) Simple Mail Transfer	TCP	25
5) Domain Name Space	TCP/UDP	53
6) DHCP	UDP	67 – Boot server 68 – Bootp client

Presentation Layer : The Presentation Layer defines a specific format for a data to be sent over the network
The major functions described at this layer is Encoding and Decoding
Ex : ASCll etc

Presentation Layer is used to define format to a data like
http://www.google.com is a clear text
but https://www.google.com is a secure (Encrypted format)

Session Layer : The Session Layer deals with sessions or Interaction between the applications
2) It is responsible for establishing,maintaining and termination of sessions.
3) Sessions layer organizes communication through Simplex,Half-Duplex or Full-Duplex
Simplex : Only one way communication like T.V,Radio etc..
Half-Duplex : Two way communication but not at the same time like waki Taki.
Full-Duplex : Communication both at the same time.

Transport Layer : It is responsible for the end-to-end transportation of data between the application.
The major funcions described at the transport layer are:
1) Identifying Service
2) Multiplexing and De-multiplexing
3) Segmentation
4) Sequencing & Reassembling
5) Error Correction
6) Flow Control

Transport deals with TCP and UDP .

Segmentation/Sequencing/Re-assembly : Segmentation/Sequencing/Re-assembly are used to divide large packet into small depending on the MTU size.

Error Control : If the it receives 4 ACK out of 5 ACK source will be specific amount of time .

Flow Control : How many packets can be sent at a time.

Network Layer : It is responsible for end-to-end transportation of data across multiple networks. Like Logical addressing & Path determination (Routing) are described at this layer.

Routing Protocols acts as the data carries and defines logical addressing

Routing Protocols are RIP,OSPF etc..

Datalink Layer : It is responsible for end-to-end delivery of data between the devices on a network segment.

It deals with the MAC address

MAC address are 12 digit Hexa-decimal identifier used to identify the devices uniquely on the network segment.

It also provides ERROR detection using CRC (Cyclic Redundancy Check) and FRAMING (Encapsulation)

Ex: Ethernet,Token Ring etc ...

Physical Layer : It transfer the data in bits format like 0's and 1's and conversion of bits over the media depends on type of the media used

Ex : Copper Media : Electric signals of different wavelength.
Fiber Media : Light pulses of different wavelength.
Wireless Media : Radio frequency waves.

IPv4 Addressing and IPv4 header

IPV4 Addressing modes are class full and classless
Class full Addressing
Class A , Class B , Class C , Class D , Class E
Classless Addressing

IPV4 Addressing modes are class full and classless

Class A

Class A : Class A is assigned to network that contains the large no of host and also higher order bit should be zero

Net id Host id

0	1st byte	2 nd byte	3 rd byte	4 th byte

The net id is 8 bits long
The host id is 24 bits long
In Class A the net id is from 0 to 127

0 0000000

0 1111111

The Remaining 3 bytes represents the host id in the Class A
The no of host id's in is Class A:2^24 - 2=16,777,214 16,777

Class B

Class B : Class B is assigned to network that contains the range from medium-sized to the large-sized and also the higher order bits of the first octet of IP addresses belonging to class B are always set to 10 .

$\longrightarrow$ Net id $\longleftarrow$ Host id

10	1st byte	2 nd byte	3 rd byte	4 th byte

The net id is 16 bits long
The host id is 16 bits long
The no of net id's in class B : 2 ^ 14 = 16384 network address .

10 000000

10 111111

The Remaining 2 bytes represents the host id in the Class B
The no of host id's in Class B :2 ^ 16 – 2 = 65534 host address

Class C

Class C : Class C is assigned to network that contains the small-sized networks and also the higher order bits of the first octet of IP addresses belonging to class C are always set to 11.

Net id Host id

11	1st byte	2 nd byte	3 rd byte	4 th byte

The net id is 24 bits long
The host id is 8 bits long
The no of net id's in class C : 2 ^ 21 = 2097152 network address.

110 00000

110 11111

The Remaining 1 bytes represents the host id in the Class C
The no of host id's in Class C : 2 ^ 8 – 2 = 254 host address

Class D

Class D : Class D is reserved for multi casting and the higher order bits of the first octet of IP addresses belonging to class D are always set to 1110.

1110	1st byte	2 nd byte	3 rd byte	4 th byte

The host id is 28 bits long

2 nd byte

1110 0000

1110 1111

Class D does not posses any sub-net mask. IP addresses belonging to class D ranges from 224.0.0.0 – 239.255.255.255.

Class E

Class E : IP addresses belonging to class E are reserved for experimental and research purposes. This class doesn't have any sub-net mask. The higher order bits of first octet of class E are always set to 1111.

1111	1st byte	2 nd byte	3 rd byte	4 th byte

The host id is 28 bits long

1111 0000

1111 1111

169.254.0.0 – 169.254.0.16 : Link local addresses
127.0.0.0 – 127.0.0.8 : Loop-back addresses
0.0.0.0 – 0.0.0.8 : used to communicate within the current network.

CLASS	Lea-ding bits	Net id Bits	Host id bits	No of networks	Address per network	Start address	End address
CLASS A	0	8	24	2^7 (128)	2^24 (16,777,216)	0.0.0.0	127.255.255.255
CLASS B	10	16	16	2^14) (16,38	2^16 (65,536)	128.0.0.0	191.255.255.255
CLASS C	110	24	8	2^21 (2,097,158)	2^8 (256)	192.0.0.0	223.255.255.255
CLASS D	1110	Not defined	Not defined	Not defined	Not defined	224.0.0.0	235.255.255.255
CLASS E	1111	Not defined	Not defined	Not defined	Not defined	240.0.0.0	255.255.255.255

IP Header

Version 4 bits	HLEN 4 bits	Service 8 bits	Total Length 16 bits	
IDENTIFICATION 16 bits			FLAGS 3bits	Fragmentation 13 bits
TIME TO LIVE 8 bits		PROTOCAL 8 bits	HEADER CHECKSUM 16 bits	
Source ip address (32 bits)				
Destination ip address (32 bits)				
Options 32 bits				

Version : 4 bits

The Version field indicates the format of the Internet header .
It indicates the version 4 .

IHL : 4 bits

Internet Header Length is the length of the Internet header in 32 bit words, and thus points to the beginning of the data.

Service Type : 8 bits

D : Minimize delay
T : Maximum Throughput
R : Maximize Reliability
C : Minimize Cost

			D	T	R	C	

Precedence Service Type don,t care

DTRC : The DTRC are used to defined the quality of the service by means of enabling and disabling.

Total length : This field represents the length of the datagram measures in octets accepts upto 65,536.

Flag bits : This field represents weather to fragment the of data is done depending on the MTU (Maximum transfer unit).

Fragmentation : This field refers to the breaking down of an ip datagram into smaller chunks . And reassembly refers to collecting all the fragments in order.
Time to live : A datagram has a limited lifetime in its travel through an Internet .
Protocol : An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP .
Check sum : A check sum is basically a value that is computed from data packet to check its integrity.
For Example : 4500 003c 1c46 4000 4006 b1c6 ac10 0a63 ac10 0a0c
If the check sum is be16 is equal to sum of all IP packet received

Class less Addressing

Private IP Addresses of Class less Addressing

The frist Class less IP range is 10.0.0.0 to 10.255.255.255.
The subnet masking for this class less is 255.0.0.0
The second Class less IP range is 172.16.0.0 to 172.31.255.255 .
The subnet masking for this class less is 255.255.0.0
The third Class less IP range is 192.168.0.0 to 192.168.255.255 .
The subnet masking for this class less is 255.255.255.0

The Rules for class less IP Addresses

1) The addresses in a block must be contiguous one after another
2) The no of addresses in a block must be a power of 2 (1,2,4,8 ...) .
3) The first addresses must be evenly divisible by the number of addresses

For Example : The range given is 205.16.37.32 upto 205.16.37.47

1) The addresses is continuous.
2) The is from 32 to 47 = 16 (2 ^4).
3) The first address is 32 is evenly divisible by 16 (no of address).

For Example : Finding no of address in a block .
We that one of th Address is 205.16.37.39/28

The binary representation of the given address is 11001101 00010000 00100101 00100111. If

we set 32 - 28 rightmost bits to 0, we get 11001101 000100000100101 0010000 or 205.16.37.32
is the starting address .

The binary representation of the given address is 11001101 00010000 00100101 00100111. If we set 32 - 28 rightmost bits to 1, we get 11001101 00010000 001001010010 1111 or 205.16.37.47 is the last address.

The private IP ranges example:

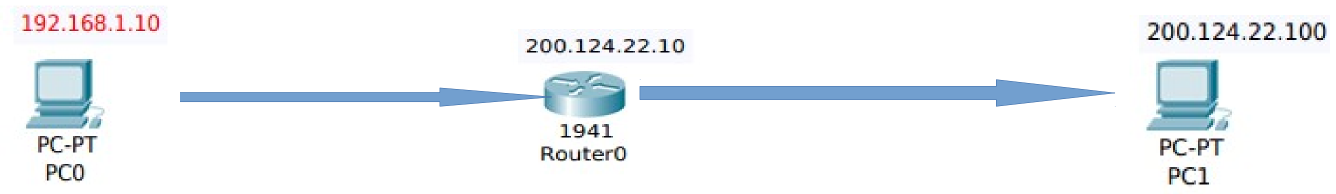
10.0.0.0 to 10.255.255.255
172.16.0.0 to 172.31.0.0
192.168.0.0 to 192.168.255.0

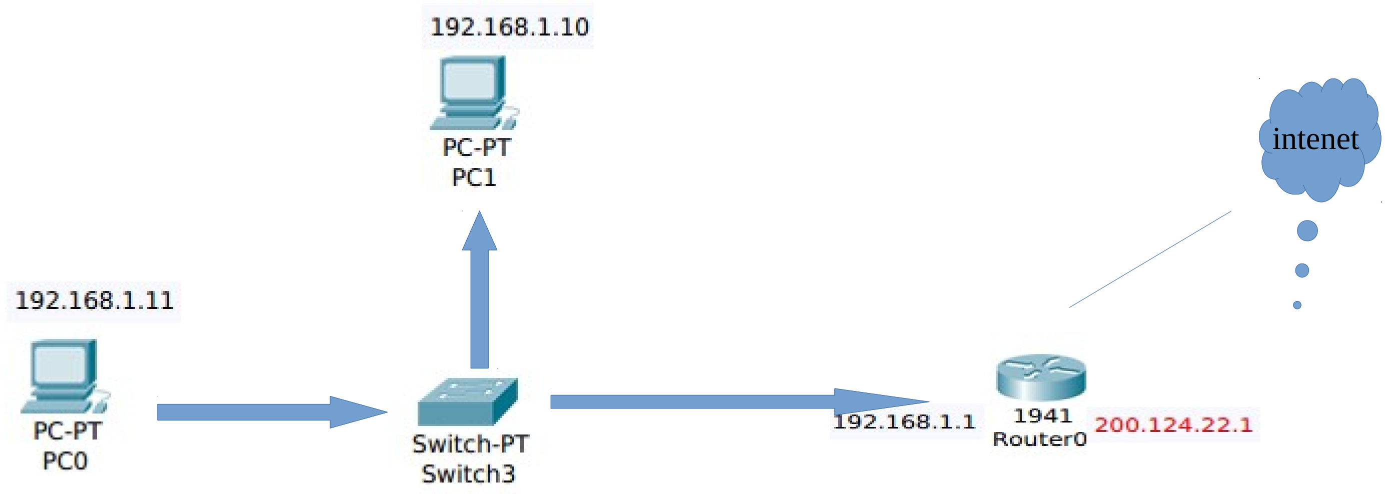
The public IP ranges example:

1.0.0.0 to 9.255.255.255
11.0.0.0 to 126.255.255.255
129.0.0.0 to 169.253.255.255

Network Address Translation

Private IP	Public IP
Class A : 10.0.0.0 – 10.255.255.255/8	1.0.0.0 – 9.255.255.255 11.0.0.0 – 126.255.255.255
class B : 172.16.0.0 – 172.31.255.255/12	128.0.0.0 – 172.15.255.255 172.32.0.0 – 191.167.255.255
class C : 192.168.0.0 – 192.168.255.255/16	192.0.0.0 – 192.167.255.255 192.169.0.0 – 223.255.255.255





Inside Local	Inside Global
192.168.1.10	200.124.22.1
192.168.1.11	200.124.22.2
192.168.1.100	200.124.22.3

Port Address Translation

Inside Local	Inside Global
192.168.1.10:50113	200.124.22.1:23112
192.168.1.11:51772	200.124.22.1:23551
192.168.1.100:51985	200.124.22.1:29007

Network Address Translation

NAT is a process in which one or more local IP address is translated into one or more global IP address & vice versa in order to provide Internet access to the local host.

Types of NAT

1) Static NAT : One of one mapping between local and global address. This is generally used for web hosting.
2) Dynamic NAT : Un registered IP address is translated into a registered (public) IP address from a pool IP address.
This is used when the no of users who wants to access the internet are fixed.

Port Address Translation

It is also known as NAT overload
1) Many local (Private) IP address can be translated to single registered IP address.
2) Port numbers are used to distinguish the traffic i.e. which traffic belongs to which IP address.
3) This is most frequently used as it is cost effectivness thousands of user can be connected to the Internet by using only one real global (Public) IP address.

Advantages NAT

1) **NAT** conserves legally registered IP address.
2) It provides privacy as the device IP address sending and receiving traffic will be hidden.
3) Eliminates address renumbering when a network evolves

Disadvantages of NAT

1) Translation results in switching path delays.
2) Certain applications will not function while NAT is enabled.
3) Complicates tunneling protocols such as IP sec.

Subnetting

1) Subnetting dividing the network into different domains.
For example : The given ip address is 192.168.100.255 and the subnet mask is 255.255.255.255
subnetting the network into the four sixty fours

i) First Subnetting 64 hosts

Network id : 192.168.100.0/26
Broadcast id : 192.168.100.63/26
Subnetmask id : 255.255.255.191

ii) Second Subnetting 64 hosts

Network id : 192.168.100.64/26
Broadcast id : 192.168.100.127/26

Subnetmask id : 255.255.255.191

iii) Third Subnetting 64 hosts

Network id : 192.168.100.128/26
Broadcast id : 192.168.100.191/26
Subnetmask id :255.255.255.191

iv) Fourth Subnetting 64 hosts

Network id : 192.168.100.192/26
Broadcast id : 192.168.100.255/26
Subnetmask id : 255.255.255.191
Here the Subnetmask is alwalys be 255.255.255.191 because we are creating subnetting
four equal parts left most 2 bits will be unchanged.

Supernetting

A supernet is created by combining several Internet Protocol (IP) networks or subnets into one network with a single classless interdomain routing (CIDR) prefix. The new combined network has the same routing prefix as the collection of the prefixes of the subnets. The procedure used to create a supernet is commonly called supernetting, route aggregation or route summarization.

For Example : Form the above example subnetting the router will maintain only one subnetmask 255.255.255.191 if we are dividing the subnorking into equal parts

Variable Length Subnetting Mask(VLSM)

The Variable Length Subnetting Mask (VLSM) amounts to "**subnetting subnets**," which means that **VLSM** allows network engineers to divide an IP address space into a hierarchy of **subnets** of different sizes, making it possible to create **subnets** with very different host counts without wasting large numbers of addresses.

For Example : If we want to create the variable length subnetting for 128,64,64

i) First Network for 128 hosts

Network id : 192.168.100.0/25
Broadcast id : 192.168.100.127/25
Subnetmask id : 255.255.255.128

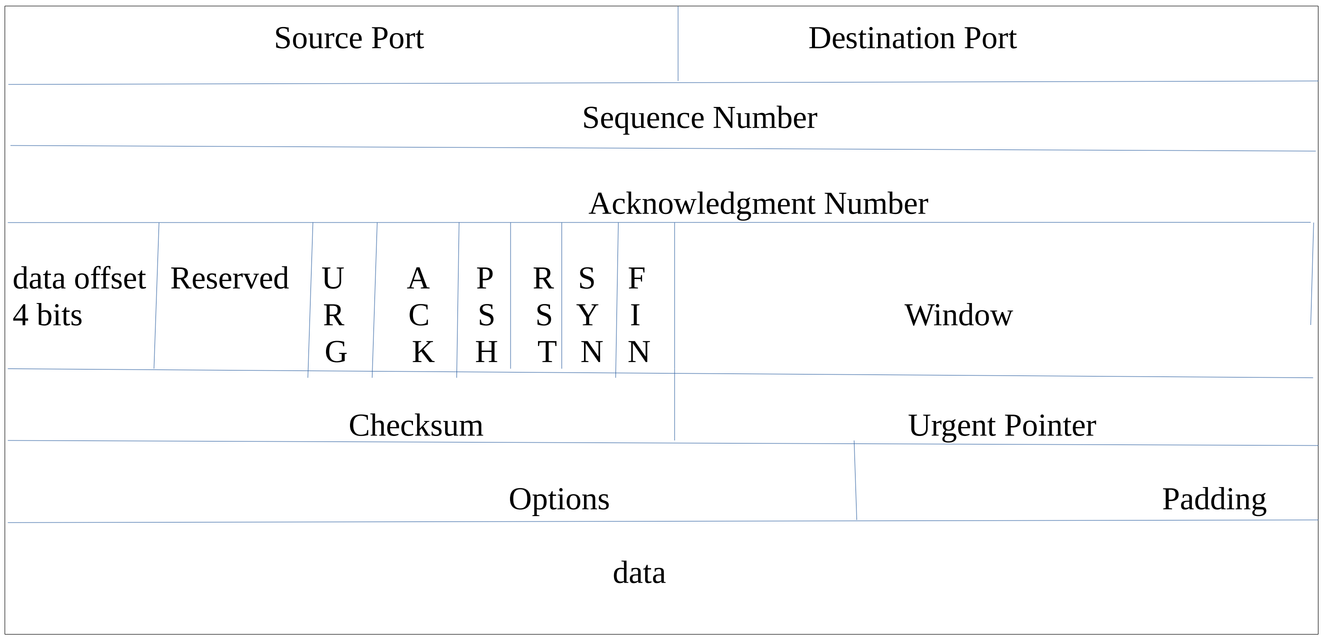
ii) Second Subnetting for 64 hosts

Network id : 192.168.100.128/26
Broadcast id :192.168.100.191/26
Subnetmask id : 255.255.255.191

iii) Third Subnetting for 64 hosts

Network id : 192.168.100.192/26
Broadcast id : 192.168.100.255/26
Subnetmask id : 255.255.255.191

Transmission Control Protocol



TCP Header Format

Source Port: 16 bits
The source port number.
Destination Port: 16 bits
The destination port number.

Sequence Number: 32 bits

The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

Acknowledgment Number: 32 bits

If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

Data Offset: 4 bits

The number of 32 bit words in the TCP Header. This indicates where the data begins. The TCP header (even one including options) is an integral number of 32 bits long.

Reserved: 6 bits

Reserved for future use. Must be zero.

Control Bits: 6 bits (from left to right):

URG: Urgent Pointer field significant

The URG flag is used to inform a receiving station that certain data within a segment is urgent and should be prioritized. If the URG flag is set, the receiving station evaluates the urgent pointer, a 16-bit field in the TCP header. This pointer indicates how much of the data in the segment, counting from the first byte, is urgent.

Admittedly, this is probably not the most illustrative example of the URG flag, but it was surprisingly difficult to find other uses of it in real-world captures.

ACK: Acknowledgment field significant

The internet's Transmission Control **Protocol** (**TCP**) is an example of an **acknowledgement**-based **protocol**. When computers communicate via **TCP**, received packets are acknowledged by sending back a packet with an **ACK** bit set.

PSH: Push Function

PSH or PUSH flag is an option provided by TCP that allows the sending application to start sending the data even when the buffer is not full (contains data less than MTU). The application needs to set the PSH flag to true for the socket and with that TCP starts pushing the data immediately.

At the receiving end, when TCP data is received with PSH set, it will immediately transfer the received data to the application.

RST: Reset the connection

In a stream of packets of a TCP connection, each packet contains a TCP header. Each of these headers contains a bit known as the "reset" (RST) flag. In most packets this bit is set to 0 and has no effect; however, if this bit is set to 1, it indicates to the receiving computer that the computer should immediately stop using the TCP connection; it should not send any more packets using the connection's identifying numbers, called ports, and discard any further packets it receives with headers indicating they belong to that connection. A TCP reset basically kills a TCP connection instantly.

When used as designed, this can be a useful tool. One common application is the scenario where a computer (computer A) crashes while a TCP connection is in progress. The computer on the other end (computer B) will continue to send TCP packets since it does not know that computer A has crashed. When computer A reboots, it will then receive packets from the old pre-crash connection. Computer A has no context for these packets and no way of knowing what to do with them, so it might send a TCP reset to computer B. This reset lets computer B know that the connection is no longer working. The user on computer B can now try another connection or take other action.

SYN: Synchronize sequence numbers

A **three-way handshake** is a method used in a TCP/IP network to create a connection between a local host/client and server. It is a three-step method that requires both the client and server to exchange SYN and ACK (**acknowledgment**) packets before actual data **communication** begins.
FIN: No more data from sender

Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to **initiate** a TCP connection. The FIN flag indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive.

Window: 16 bits
The number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.
Checksum: 16 bits
The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header and text. If a segment contains an odd number of header and text octets to be

checksummed, the last octet is padded on the right with zeros to form a 16 bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum,the checksum field itself is replaced with zeros.The checksum also covers a 96 bit pseudo header conceptually.prefixed to the TCP header. This pseudo header contains the Source Address, the Destination Address, the Protocol, and TCP length. This gives the TCP protection against misrouted segments. This information is carried in the Internet Protocol and is transferred across the TCP/Network interface in the arguments or results of calls by the TCP on the IP.

Source Address		
Destination Address		
zero	PTCL	TCP Length

The TCP Length is the TCP header length plus the data length in octets (this is not an explicitly transmitted quantity, but is computed), and it does not count the 12 octets of the pseudo header.
Urgent Pointer: 16 bits
This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field is only be interpreted in segments with the URG control bit set.

Options: variable

Options may occupy space at the end of the TCP header and are a multiple of 8 bits in length. All options are included in the checksum. An option may begin on any octet boundary. There are two cases for the format of an option:
Case 1: A single octet of option-kind.
Case 2: An octet of option-kind, an octet of option-length, and the actual option-data octets.
The option-length counts the two octets of option-kind and option-length as well as the option-data octets.
Note that the list of options may be shorter than the data offset field might imply. The content of the header beyond the End-of-Option option must be header padding (i.e., zero).

A TCP must implement all options

Currently defined options include (kind indicated in octal):

Kind	**Length**	**Meaning**
----	------	-------
0	-	End of option list.
1	-	No-Operation.
2	4	Maximum Segment Size.

Specific Option Definitions

End of Option List

```
        00000000
         Kind=0
```

This option code indicates the end of the option list. This might not coincide with the end of the TCP header according to the Data Offset field. This is used at the end of all options,not the end of each option, and need only be used if the end of the options would not otherwise coincide with the end of the TCP header.

No-Operation

```
        00000001
        Kind=1
```

This option code may be used between options, for example, to align the beginning of a subsequent option on a word boundary. There is no guarantee that senders will use this option, so receivers must be prepared to process options even if they do not begin on a word boundary.

Maximum Segment Size

```
        00000010  00000100  max seg size
         Kind=2   Length=4
```

Maximum Segment Size Option Data: 16 bits
If this option is present, then it communicates the maximum receive segment size at the TCP which sends this segment. This field must only be sent in the initial connection request (i.e., in segments with the SYN control bit set). If this option is not used, any segment size is allowed.

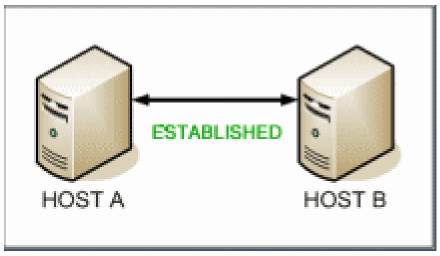
Padding: variable

The TCP header padding is used to ensure that the TCP header ends and data begins on a 32 bit boundary. The padding is composed of zeros.

TCP 3-Way Handshake (SYN,SYN-ACK,ACK)

The TCP three-way handshake in Transmission Control Protocol (also called the TCP-handshake; three message handshake and/or SYN-SYN-ACK) is the method used by TCP set up a TCP/IP connection over an Internet Protocol based Network. TCP's three way handshaking technique is often referred to as "SYN-SYN-ACK" (or more accurately SYN, *SYN-ACK*, ACK) because there are three messages transmitted by TCP to negotiate and start a TCP session between two computers. The TCP handshaking mechanism is designed so that two computers attempting to communicate can negotiate the parameters of the network TCP socket connection before transmitting data such as SSH and HTTP web browser requests.

This 3-way handshake process is also designed so that both ends can initiate and negotiate separate TCP socket connections at the same time. Being able to negotiate multiple TCP socket connections in both directions at the same time allows a single physical network interface, such as ethernet, to be multiplexed to transfer multiple streams of TCP data simultaneously.



Host A **sends** a TCP **SYN**chronize packet to Host B

Host B receives A's **SYN**

Host B **sends** a **SYN**chronize-ACKnowledgement

Host A receives B's **SYN-ACK**

Host A **sends** **ACK**nowledge

Host B receives **ACK**.
TCP socket connection is ESTABLISHED.

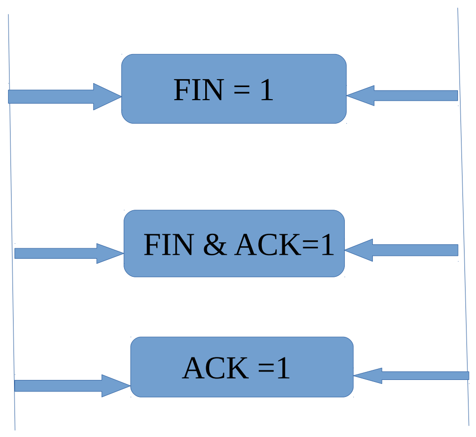
SYNchronize and ACKnowledge messages are indicated by a either the SYN bit, or the ACK bit inside the TCP header, and the SYN-ACK message has both the SYN and the ACK bits turned on (set to 1) in the TCP header.

TCP knows whether the network TCP socket connection is opening, synchronizing, established by using the **SYN**chronize and ACKnowledge messages when establishing a network TCP socket connection.

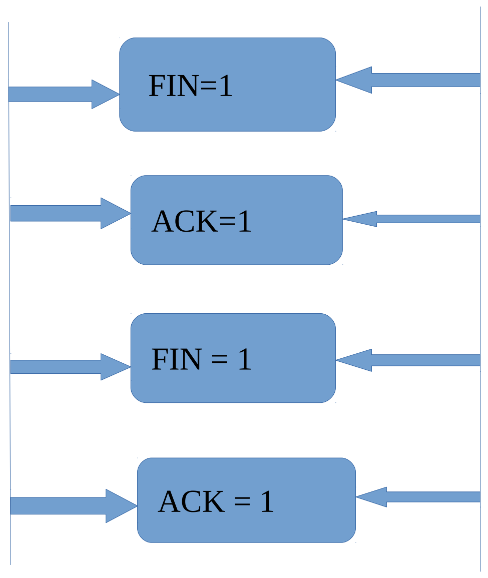
When the communication between two computers ends, another 3-way communication is performed to tear down the TCP socket connection. This setup and teardown of a TCP socket connection is part of what qualifies TCP a reliable protocol. TCP also acknowledges that data is successfully received and guarantees the data is reassenbled in the correct order.

Note that UDP is connectionless. That means UDP doesn't establish connections as TCP does, so UDP does not perform this 3-way handshake and for this reason, it is referred to as an unreliable protocol That doesn't mean UDP can't transfer data, it just doesn't negotiate how the conneciton will work, UDP just transmits and hopes for the best

3-way handshaking for connection termination or connection closer



4-Way handshaking for connection termination



Here the data transmission is from client to server and server to clinet
TCP uses 4-types of timer

1) Retransmission Timer – To retransmit lost segments, TCP uses retransmission timeout (RTO). When TCP sends a segment the timer starts and stops when the acknowledgment is received. If the timer expires timeout occurs and the segment is retransmitted. RTO (retransmission timeout is for 1 RTT) to calculate retransmission timeout we first need to calculate the RTT(round trip time).

RTT three types –

- **Measured RTT(RTTm) –** The measured round-trip time for a segment is the time required for the segment to reach the destination and be acknowledged, although the acknowledgment may include other segments.

- **Smoothed RTT(RTTs) –** It is the weighted average of RTTm. RTTm is likely to change and its fluctuation is so high that a single measurement cannot be used to calculate RTO.

  Initially -> No value
  After the first measurement -> RTTs=RTTm
  After each measurement -> RTTs= (1-t)*RTTs + t*RTTm
  Note: t=1/8 (default if not given)

- **Deviated RTT(RTTd) –** Most implementation do not use RTTs alone so RTT deviated is also calculated to find out RTO.

  Initially -> No value
  After the first measurement -> RTTd=RTTm/2
  After each measurement -> RTTd= (1-k)*RTTd + k*(RTTm-RTTs)
  Note: k=1/4 (default if not given)

Retransmission Timeout : RTO calculation – The value of RTO is based on the smoothed round-trip time and its deviation. Most implementations use the following formula to calculate the RTO:

Initial value -> Original (given in question)
After any measurement -> RTO=RTTs + 4*RTTd

2) Persistent Timer – To deal with a zero-window-size deadlock situation, TCP uses a persistence timer.When the sending TCP receives an acknowledgment with a window size of zero, it starts a persistence timer. When the persistence timer goes off, the sending TCP sends a special segment called a probe. This segment contains only 1 byte of new data. It has a sequence number, but its sequence number is never acknowledged; it is even ignored in calculating the sequence number for the rest of the data. The probe causes the receiving TCP to resend the acknowledgment which was lost.

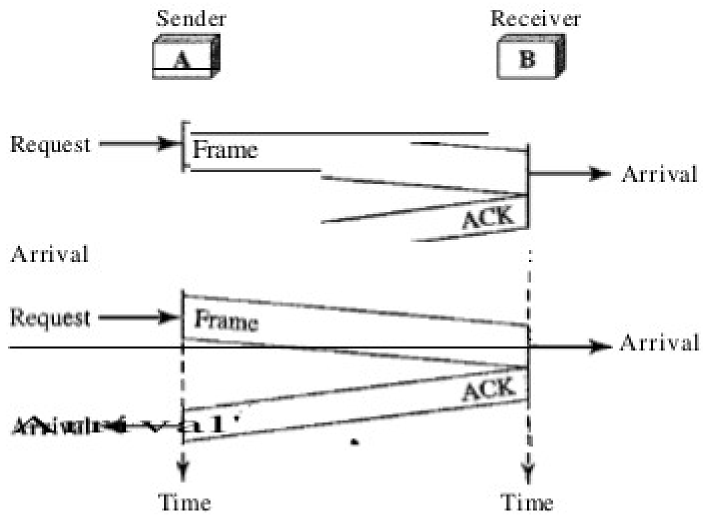
3) Keep Alive Timer – A keepalive timer is used to prevent a long idle connection between two TCPs. If a client opens a TCP connection to a server transfers some data and becomes silent the client will crash. In this case, the connection remains open forever. So a keepalive timer is used. Each time the server hears from a client, it resets this timer. The time-out is usually 2 hours. If the server does not hear from the client after 2 hours, it sends a probe segment.If there is no response after 10 probes, each of which is 75 s apart, it assumes that the client is down and terminates the connection.

4) Time Wait Timer – This timer is used during tcp connection termination. The timer starts after sending the last Ack for 2nd FIN and closing the connection.

Stop-and-Wait Protocol

If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use. Normally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service. To prevent the receiver from becoming over- whelmed with frames,we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender.

The protocol we discuss now is called the Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our previous protocol.



Stop-and-Wait Automatic Repeat Request
Our first protocol, called the Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol. Let us see how this protocol detects and corrects errors.

To detect and correct corrupted frames, we need to add redundancy bits to our data frame (see Chapter 10). When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is mani- fested by the silence of the receiver.

Lost frames are more difficult to handle than corrupted ones. In our previous proto- cols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.

The comlpted and lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since the protocol uses the stop-and-wait mecha- nism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network.

Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field. In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.

Sequence Numbers

As we discussed, the protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame.

One important consideration is the range of the sequence numbers. Since we want to minimize the frame size, we look for the smallest range that provides unambiguous communication. The sequence numbers of course can wrap around. For example, if we decide that the field is m bits long, the sequence numbers start from 0, go to 2 m - 1, and then are repeated.

Let us reason out the range of sequence numbers we need. Assume we have used x as a sequence number; we only need to use x + 1 after that. There is no need for x + 2. To show this, assume that the sender has sent the frame numbered x. Three things can happen.

1. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, causing the sender to send the next frame numbered x + 1.

2. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the frame (numbered x) after the time-out. Note that the frame here is a duplicate. The receiver can recognize this fact because it expects frame x + I but frame x was received.

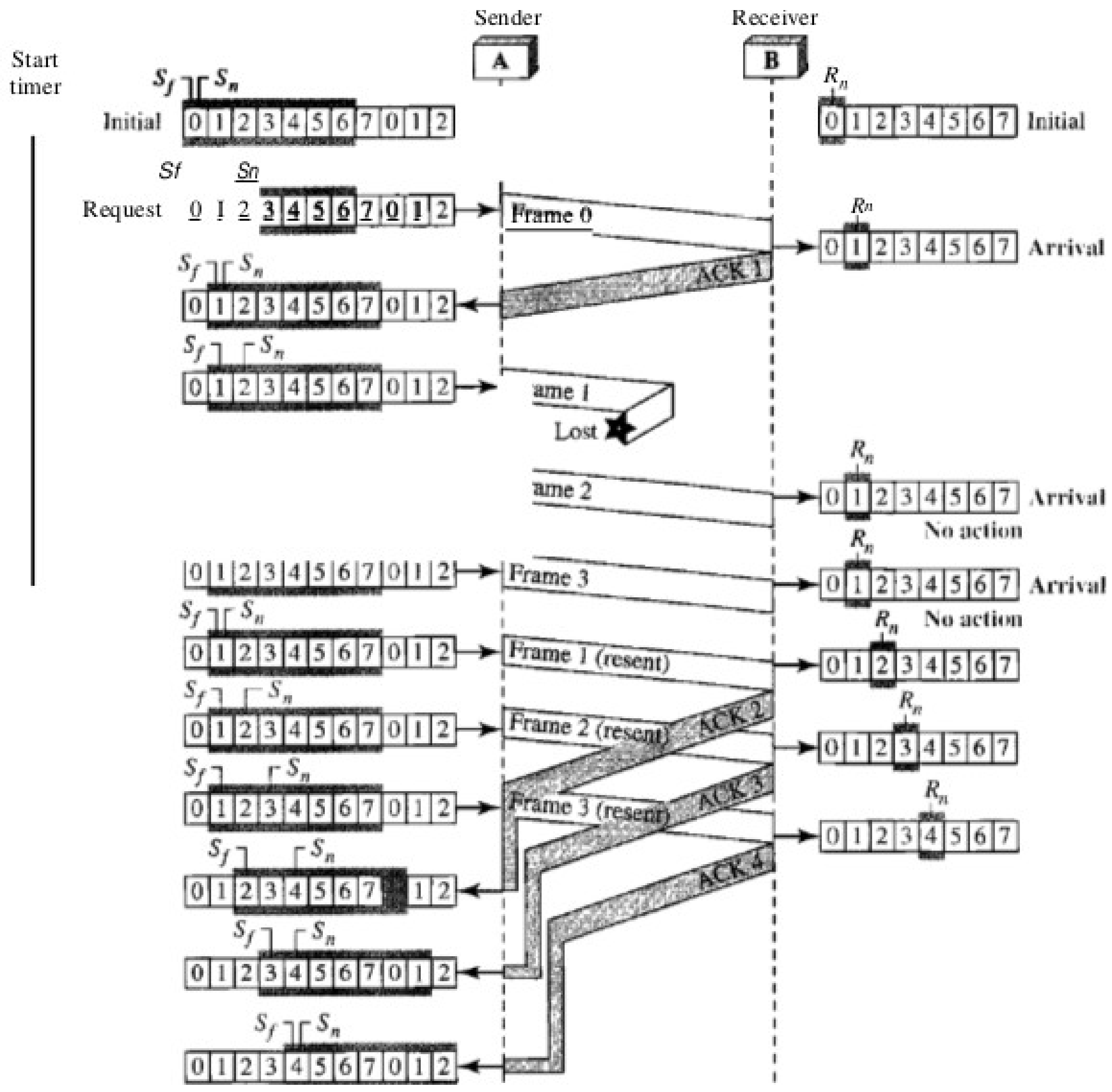
3. The frame is corrupted or never arrives at the receiver site; the sender resends the frame (numbered x) after the time-out. We can see that there is a need for sequence numbers x and x + I because the receiver needs to distinguish between case 1 and case 2. But there is no need for a frame to be numbered x + 2. In case 1, the frame can be numbered x again because frames x and x + 1 are acknowledged and there is no ambiguity at either site. In cases 2 and 3, the new frame is x + I, not x + 2. If only x and x + 1 are needed, we can let x = 0 and x + I == 1. This means that the sequence is 0, I, 0, I, 0, and so on.

Acknowledgment Numbers

Since the sequence numbers must be suitable for both data frames and ACK frames, we use this convention: The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver. For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next). If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected).

GO – BACK – N ARQ

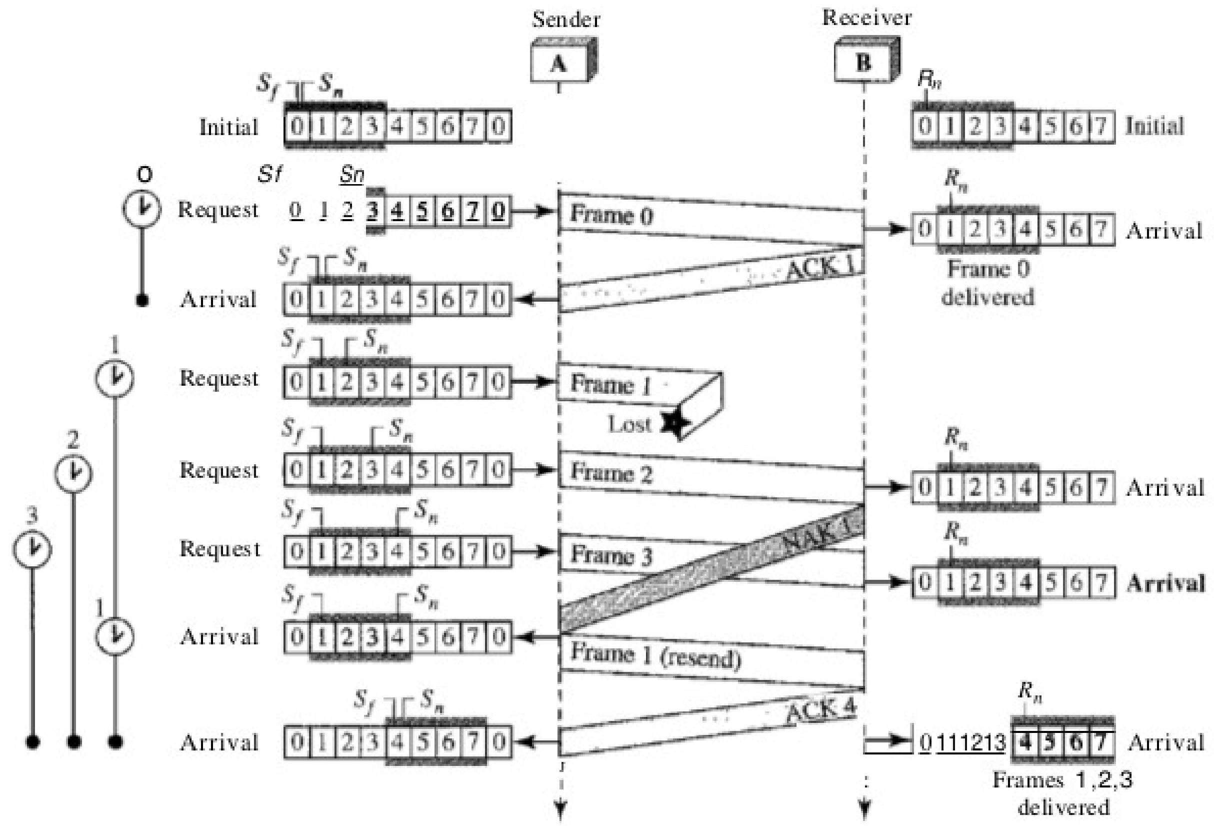
The reader may find that there is a similarity between Go-Back-N ARQ and Stop-and-Wait ARQ. We can say that the Stop-and-WaitARQ Protocol is actually a Go-Back-N ARQ in which there are only two sequence numbers and the send window size is 1. In other words, m = 1, 2 m - 1 = 1. In Go-Back-N ARQ, we said that the addition is modulo-2 m; in Stop-and-Wait ARQ it is 2, which is the same as 2 m when m = 1.



Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission. For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective RepeatARQ. It is more efficient for noisy links, but the processing at the receiver is more complex.

The Selective Repeat Protocol also uses two windows: a send window and a receive win-dow. However, there are differences between the windows in this protocol and the ones in Go-Back-N. First, the size of the send window is much smaller; it is 2 m- I . The reason for this will be discussed later. Second, the receive window is the same size as the send window. The send window maximum size can be 2 m - I . For example, if m = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the Go-Back-N Protocol). The smaller window size means less efficiency in filling the pipe, but the fact that there are fewer duplicate frames can compensate for this.
The protocol uses the same variables as we discussed for Go-Back-N

Selective Repeat Timing Diagram



Selective Repeat is part of the automatic repeat-request (ARQ). With selective repeat, the sender sends a number of frames specified by a window size even without the need to wait for individual ACK from the receiver as in Go-Back-N ARQ. The receiver may selectively reject a single frame, which may be retransmitted alone; this contrasts with other forms of ARQ, which must send every frame from that point again. The receiver accepts out-of-order frames and buffers them. The sender individually retransmits frames that have timed out.

User Datagram Packet (UDP)

1) It is a connection less communication protocol of the transport layer.
2) It is often used for broadcast based communication.
3) DNS and DHCP also make use of UDP for queries and responses in broadcast based communication.
4) Supports error checking but not error correction and retransmission.
5) Here the checksum involves a mathematical completion both “Header” and “data”.

This protocol provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed. Applications requiring ordered reliable delivery of streams of data should use the Transmission Control Protocol

Source Port	Destination Port
Length	Checksum
data octets ...	

Fields:

Source Port is an optional field, when meaningful, it indicates the port of the sending process, and may be assumed to be the port to which a reply should be addressed in the absence of any other information. If not used, a value of zero is inserted.

Destination Port has a meaning within the context of a particular internet destination address.

Length is the length in octets of this user datagram including this header and the data. (This means the minimum value of the length is eight.)

Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

Internet Control Message Protocol

1) IP protocol used for host-to-host datagram service in a system of interconnected networks ,
but two deficiencies:
- Lack of error control
- Lack of query mechanisms

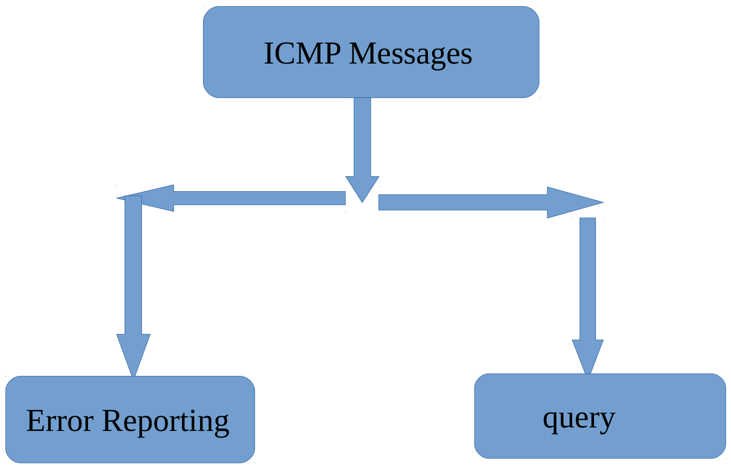
2) ICMP was designed to overcome these deficiencies
 Type field specifies the type of error (or) query message.
3) Code field specifies the reason for the message type.
4) In ICMP checksum is calculate over the entire message (header and data)

ICMP Packet Format

Type	code	checksum
Message specification information		

Types of ICMP Messages

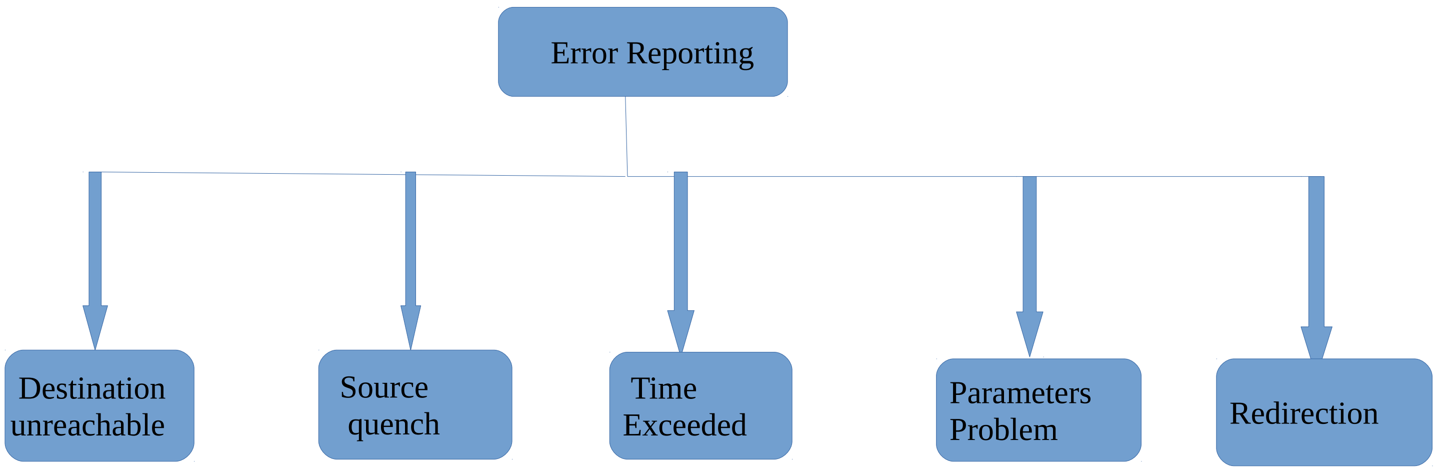
Two types of ICMP messages



ICMP error reporting

● Main responsibility of ICMP is to report errors.
● ICMP report error message to the original source.
● ICMP error message contain a data section that includes the IP header of the original datagram and first 8bytes of data in datagram (port number and sequence number).

Types of error



Destination unreachable

● When a router cannot route a datagram, this case router send ICMP error-report as destination unreachable to the source.
Type :3
code : 0 net unreachable.
code : 1 host unreachable.
code : 2 protocol unreachable.

code : 3 port unreachable.
code : 4 fragmentation needed and DF set.
code : 5 source route failed.

Source quench

● If a router or host discards a datagram due to congestion, it send source-quench message to the sender.
Type :4
code : 0
● The source quench message is a request to the host to cut back the flow it is sending traffic to the Internet destination.

Time exceeded

● Whenever router receive the datagram with time-to-live as zero and does not receive all fragments in set time, it discarded the datagram and sends a time_exceeded message to source. If a host reassembly the fragments cannot completed due to missing fragments with in its time limit.
Type :11
code: 0 time to live exceeded in transit.
code: 1 fragment reassembly time exceeded.

Parameters problem

● If any ambiguity is found in header of a datagram the datagram is discarded and parameter-problem message send back to source.
Type :12
code:0 indicates the error

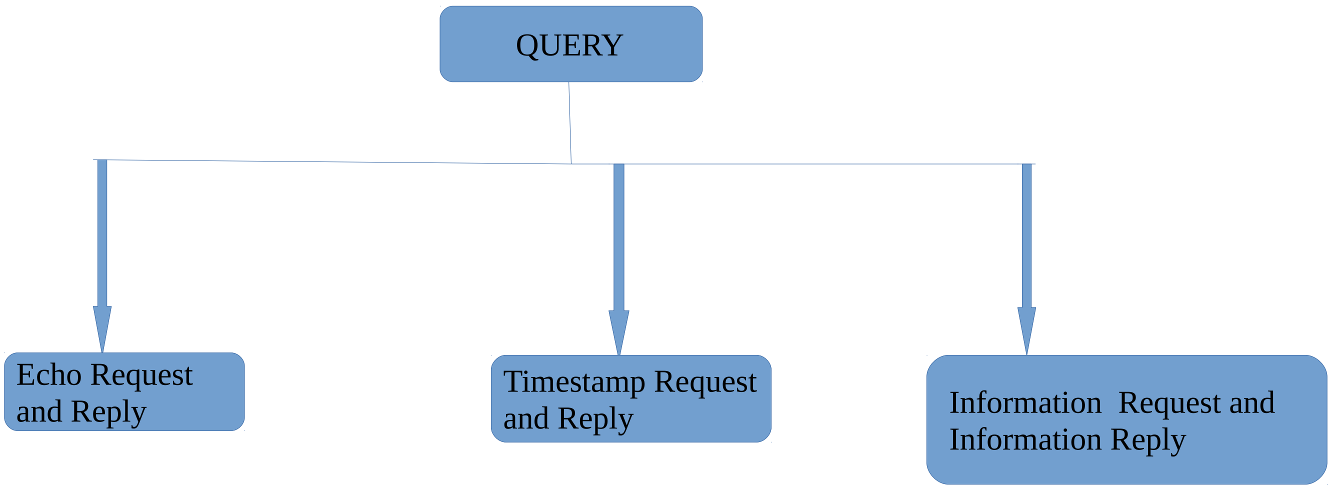
Redirection

● Default gateway redirect the datagram to another router while doing this router will send ICMP redirection message to host it will help to update the routing table information dynamically.
Type :5
Code : 0 Redirect datagrams for the Network.
Code : 1 Redirect datagrams for the host.
Code : 2 Redirect datagra**4) Time Wait Timer** ms for the type of service and Network.
Code : 3 Redirect datagrams for the type of service and host.

No ICMP generated

● Response to datagram carrying an ICMP error message.
● A datagram having multi-cast address.
● A datagram having special address like 127.0.0.0 or 0.0.0.0
● If fragment zero is not available then no time exceeded need be sent at all.

ICMP query reporting


QUERY
Echo Request and Reply
Timestamp Request and Reply
Information Request and Information Reply

Echo request and reply

● An echo-request message can be sent be sent by a host or router.
● An echo-reply send by host or router which receives echo request message.
Type : 8 (echo request)
Type : 0 (echo reply)
code :0
●
echo-request and echo-reply can test the reachability of host
Ex:ping command

Time stamp Request and reply

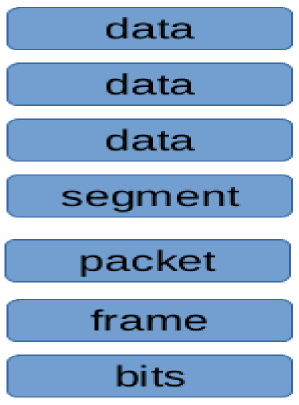
● Time stamp-request and time stamp-reply can be used to calculate the round trip time between source and destination machine, even their clocks are not synchronized
Type : 13 (time stamp request message)
Type : 14 (time stamp reply message)
Code : 0
sending time = received time stamp – original time stamp
received time stamp = packet return time – transmit time stamp
round-trip time = sending time + receiving time

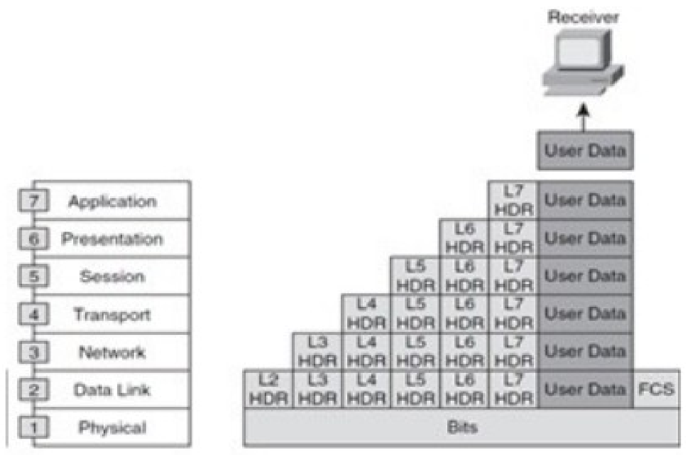
Information request And reply

● This message may be sent with the source network in the IP header source and destination address fields zero. The replying IP module should send the reply with the addresses fully specified.
Type: 15 information request message
Type: 16 information reply message
Code : 0

Decapsulation

● Data decapsulation is nothing but receiving the encapsulated data from source host. The data at the bottom of the layer are packaged several times to ensure security. these data are unpacked until they reach the network application awaiting the data.





Switching Protocols :

Virtual Local Area Network

A collision domain is simply defined as any physical segment where a collision can occur. Hubs can only operate at half-duplex, and thus all ports on a hub belong to the same collision domain.

Layer-2 switches can operate at full duplex. Each individual port on a switch belongs to its own collision domain. Thus, Layer-2 switches create more collision domains, which results in fewer collisions.

Like hubs though, Layer-2 switches belong to only one broadcast domain. A Layer-2 switch will forward both broadcasts and multicasts out every port but the originating port.

Only Layer-3 devices separate broadcast domains. Because of this, Layer-2 switches are poorly suited for large, scalable networks. The Layer-2 header provides no mechanism to differentiate one network from another, only one host from another.

Virtual LANs (VLANs)

By default, a switch will forward both broadcasts and multicasts out every port but the originating port. However, a switch can be logically segmented into separate broadcast domains, using Virtual LANs (or VLANs).

Each VLAN represents a unique broadcast domain:

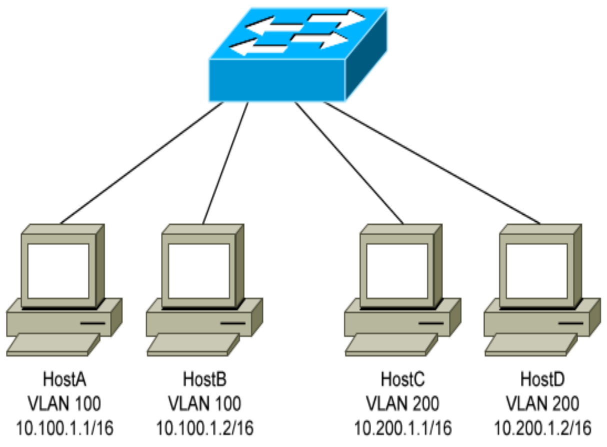
- Traffic between devices within the same VLAN is switched.
- Traffic between devices in different VLANs requires a Layer-3 device to communicate.

Broadcasts from one VLAN will not be forwarded to another VLAN. The logical separation provided by VLANs is not a Layer-3 function. VLAN
tags are inserted into the Layer-2 header.

Thus, a switch that supports VLANs is not necessarily a Layer-3 switch. However, a purely Layer-2 switch cannot route between VLANs.

Remember, though VLANs provide separation for Layer-3 broadcast domains, they are still a Layer-2 function. A VLAN often has a direct relationship with an IP subnet, though this is not a requirement.

VLAN Example
Consider the following example:



Four hosts are connected to a Layer-2 switch that supports VLANs:

- HostA and HostB belong to VLAN 100
- HostC and HostD belong to VLAN 200

Because HostA and HostB belong to the same VLAN, they belong to the same broadcast domain as well. The switch will be able to forward frames between the two hosts without the need of a Layer-3 device, such as a router.

Likewise, HostC and HostD belong to the same VLAN, and thus the same broadcast domain. They also will be able to communicate without an intervening Layer-3 device.

However, HostA and HostB will not be able to communicate with HostC and HostD. They are members of separate VLANs, and belong in different broadcast domains. Thus, a Layer-3 device is required for those hosts to communicate.

A broadcast sent from a host in VLAN 100 will be received by all other hosts in that same VLAN. However, that broadcast will not be forwarded to any other VLAN, such as VLAN 200.

On Cisco switches, all interfaces belong to VLAN 1 by default. VLAN 1 is also considered the Management VLAN, and should be dedicated for system traffic such as CDP, STP, VTP, and DTP.

Advantages of VLANs

VLANs provide the several benefits:
• Broadcast Control – eliminates unnecessary broadcast traffic, mproving network performance and scalability.
• Security – logically separates users and departments, allowing administrators to implement access-lists to control traffic between VLANs.
• Flexibility – removes the physical boundaries of a network, allowing a user or device to exist anywhere.
VLANs are very common in LAN and campus networks. For example, user networks are often separated from server networks using VLANs.
VLANs can span across WANs as well, though there are only limited scenarios where this is necessary or recommended.

VLAN Membership

VLAN membership can be configured one of two ways:

- Statically
- Dynamically

Statically assigning a VLAN involves manually assigning an individual or group of ports to a VLAN. Any host connected to that port (or ports) immediately becomes a member of that VLAN. This is transparent to the host - it is unaware that it belongs to a VLAN.

VLANs can be assigned dynamically based on the MAC address of the host. This allows a host to remain in the same VLAN, regardless of which switch port it is connected to.

Dynamic VLAN assignment requires a separate database to maintain the MAC-address-to-VLAN relationship. Cisco developed the VLAN Membership Policy Server (VMPS) to provide this functionality.

In more sophisticated systems, a user's network account can be used to determine VLAN membership, instead of a host's MAC address.

Static VLAN assignment is far more common than dynamic, and will be the focus of this guide.

Creating VLANs

By default, all interfaces belong to VLAN 1. To assign an interface to a different VLAN, that VLAN must first be created:

```
Switch(config)# vlan 100
Switch(config-vlan)# name SERVERS
```

The first command creates VLAN 100, and enters VLAN configuration mode. The second command assigns the name SERVERS to this VLAN.

Note that naming a VLAN is not required.

The standard range of VLAN numbers is 1 – 1005, with VLANs 1002-1005 reserved for legacy Token Ring and FDDI purposes.

A switch operating in VTP transparent mode can additionally use the VLAN range of 1006 – 4094. These are known as extended-range VLANs. VTP is covered in great detail later in this guide.

To remove an individual VLAN:

```
Switch(config)# no vlan 100
```

Note that VLAN 1 cannot be removed. To remove a group of VLANs:

```
Switch(config)# no vlan 150-200
```

To view all created VLANs, including the interfaces assigned to each VLAN:

```
Switch# show vlan

VLAN        NAMES                   STATUS       PORT
1           default                 active       gi1/1-24
100         SERVERS                 active
1002        fddi-default            suspended
1003        token-ring-default      suspended
1004        fddinet-default         suspended
1005        trnet-default           suspended
```

Note that no interfaces have been assigned to the newly created VLAN 100 yet.

Statically Assigning VLANs

To statically assign an interface into a specific **VLAN:**

```
Switch(config)# interface gi1/10
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 100
```

The first command enters interface configuration mode. The second command indicates that this is an access port, as opposed to a trunk port. This will be explained in detail shortly.

The third command assigns this access port to VLAN 100. Note that the VLAN number is specified, and not the VLAN name.

The show vlan command should now reflect the new VLAN assignment

For switches running in VTP server or client mode, the list of VLANs are stored in a database file named vlan.dat. The vlan.dat file is usually stored in flash, though on some switch models it is stored in NVRAM. The VLAN database will be maintained even if the switch is rebooted.

For switches running in VTP transparent mode, the list of VLANs is stored in the startup-config file in NVRAM. VTP is covered extensively later in this guide.

Regardless of VTP mode, the VLAN assignment for every switch interface is stored in the switch's startup-config

VLAN Port Types

A VLAN-enabled switch supports two types of ports:

- Access ports
- Trunk ports

An access port is a member of only a single VLAN. Access ports are most often used to connect host devices, such as computers and printers. By default on Cisco switches, all switch ports are access ports.

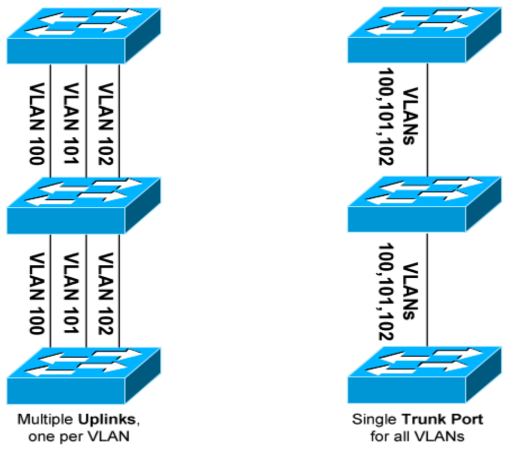
Any host connected to an access port immediately becomes a member of the VLAN configured on that port. This is transparent to the host - it is unaware that it belongs to a VLAN.

It is possible for a VLAN to span more than one switch. There are two methods of connecting a VLAN across multiple switches:

- Create uplink access ports between the switches, one for each VLAN.
- Create a **trunk connection** between the switches.

A **trunk port** is not a member of a single VLAN. The traffic from any or all VLANs can traverse trunk links to reach other switches.

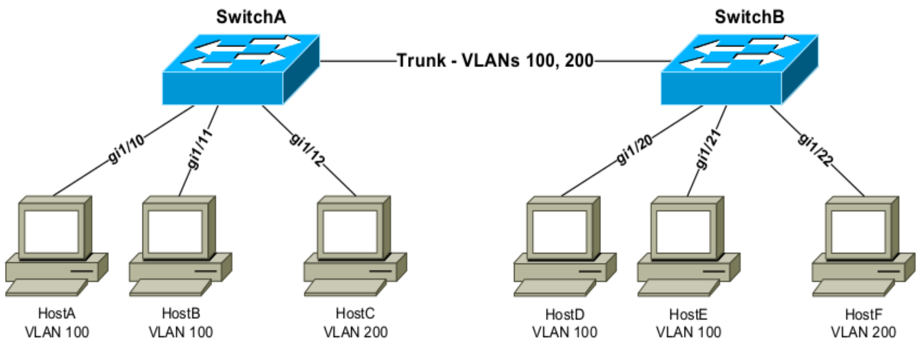
Uplinking access ports quickly becomes unfeasible in large switching environments. The following illustrates the advantage of using trunk ports:



VLAN Frame-Tagging

When VLANs span multiple switches, a mechanism is required to identify which VLAN a frame belongs to. This is accomplished through frame tagging, which places a VLAN ID in each frame.

Tagging only occurs when a frame is sent out a trunk port. Traffic sent out access ports is never tagged. Consider the following example:



If HostA sends a frame to HostB, no frame tagging will occur:
- The frame never leaves the SwitchA.
- The frame stays within its own VLAN.
- The frame is simply switched to HostB.

If HostA sends a frame to HostC, which is in a separate VLAN:
- The frame again never leaves the switch.
- Frame tagging will still not occur.
- Because HostC is in a different VLAN, the frame must be routed.

If HostA sends a frame to HostD, which is on a separate switch:
- The frame is sent out the trunk port to SwitchB.
- The frame must be tagged as it is sent out the trunk port.
- The frame is tagged with its **VLAN ID - VLAN 100** in this example.
- When SwitchB receives the frame, it will only forward it out ports belonging to VLAN 100 – gi1/20 and gi1/21.
- If SwitchB has HostD’s MAC address in its table, it will forward the frame only out the appropriate port – gi1/20.
- The VLAN tag is stripped from the frame before being forwarded to the host.

Frame Tagging Protocols

Cisco switches support two frame tagging protocols:
- Inter-Switch Link (ISL)
- IEEE 802.1Q

The tagging protocol can be manually specified on a trunk port, or dynamically negotiated using Cisco’s proprietary Dynamic Trunking Protocol (DTP).

Inter-Switch Link (ISL)

Inter-Switch Link (ISL) is Cisco's proprietary frame tagging protocol. ISL supports several technologies:

- Ethernet
- Token Ring
- FDDI
- ATM

ISL encapsulates a frame with an additional header (26 bytes) and trailer (4 bytes). Thus, ISL increases the size of a frame by 30 bytes.

The header contains several fields, including a 15-bit VLAN ID. The trailer contains an additional 4-byte CRC to verify data integrity.

Normally, the maximum possible size of an Ethernet frame is 1518 bytes. This is known as the Maximum Transmission Unit (MTU). Most Ethernet devices use a default MTU of 1514 bytes.

ISL increases the frame size by another 30 bytes. Thus, most switches will disregard ISL-tagged frames as being oversized, and drop the frame. An oversized frame is usually referred to as a giant. Somewhat endearingly, a slightly oversized frame is known as a baby giant.

Cisco switches are specifically engineered to support these giant ISL tagged frames. Note that this is a key reason why ISL is Cisco-proprietary.

ISL supports a maximum of 1000 VLANs on a trunk port. ISL is also almost entirely deprecated - most modern Cisco switches no longer support it.

IEEE 802.1Q

IEEE 802.1Q, otherwise referred to as dot1Q, is an industry-standard frame-tagging protocol.

802.1Q is supported by nearly all switch manufacturers, including Cisco. Because 802.1Q is an open standard, switches from different vendors can be trunked together.

Recall that ISL encapsulates a frame with an additional header and trailer. In contrast, 802.1Q embeds a 4-byte VLAN tag directly into the Layer-2 frame header. Because the Layer-2 header is modified, 802.1Q must recalculate the frame's CRC value.

The VLAN tag includes a 12-bit VLAN ID. This tag increases the size of an Ethernet frame, from its default of 1514 bytes to 1518 bytes. Nearly all modern switches support the 802.1Q tag and the slight increase in frame size.

802.1Q supports a maximum of 4096 VLANs on a trunk port.

Configuring Trunk Links

To manually configure an interface as a trunk port:

```
Switch(config)# interface gi2/24
Switch(config-if)# switchport mode trunk
```

For a switch that supports both ISL and 802.1Q, the tagging or encapsulation protocol must be configured first:

```
Switch(config)# interface gi2/24
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk

Switch(config)# interface gi2/24
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
```

Important note: Both sides of the trunk must be configured with the same tagging protocol. Otherwise, a trunk connection will not form.

If the switch only supports 802.1Q, the switchport trunk encapsulation command will not be available.

Configuring Trunk Links (continued)

The switch can negotiate the tagging protocol, using **DTP:**

```
Switch(config)# interface gi2/24
Switch(config-if)# switchport trunk encapsulation negotiate
Switch(config-if)# switchport mode trunk
```

The tagging protocol that is supported by both switches will be used. If the switches support both ISL and 802.1Q, ISL will be the preferred protocol.

By default, all active VLANs are allowed to traverse a trunk link. While this is convenient, a good security practice is to allow only necessary VLANs over a trunk.

To explicitly allow a subset of **VLANs on a trunk port:**

```
Switch(config)# interface gi2/24
Switch(config-if)# switchport trunk allowed vlan 3,9,11-15
```

The above command will force the trunk link to only forward traffic from **VLANS 3, 9, and 11 – 15.**
To remove a VLAN from the allowed list:

```
Switch(config)# interface gi2/24
Switch(config-if)# switchport trunk allowed vlan remove 12
```

To add a specific **VLAN** back into the allowed list:

```
Switch(config)# interface gi2/24
Switch(config-if)# switchport trunk allowed vlan add 25
```

Important Note: It is common to restrict the allowed VLANs on a trunk link, and then add to the allowed list as new VLANs are created. However, don't forget to use the add parameter. If add is omitted, the command will replace the list of allowed VLANs on the trunk link,

To allow all **VLANs** except for a specific range:

Switch(config-if)# switchport trunk allowed vlan except 50-99

To allow all **VLANs** again:

Switch(config-if)# switchport trunk allowed vlan all

Native VLANs

Recall that a trunk port tags frames with a VLAN ID. But what happens if a t runk port receives an untagged frame?

The native VLAN determines the VLAN that untagged traffic belongs to. By default on all trunking ports, the native VLAN is VLAN 1. The native VLAN can be changed on a per trunk port basis:

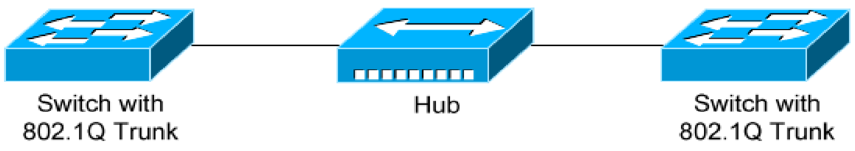
```
Switch(config)# interface gi2/24
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk native vlan 42
```

Only one native **VLAN** can be assigned to a trunk port. All untagged traffic received on this port will become a member of the native VLAN. Additionally, frames belonging to the native **VLAN** are not tagged when being sent out a trunk port.

Native VLANS are only supported on **802.1Q** trunk ports. ISL does not support untagged frames, and will always tag frames from all VLANs.

The native VLAN must be configured **identically on both sides of the 802.1Q trunk,** otherwise the switches will not form a trunk connection.

The original intent of native **VLANs** was for legacy compatibility with hubs. Consider the following deprecated example:

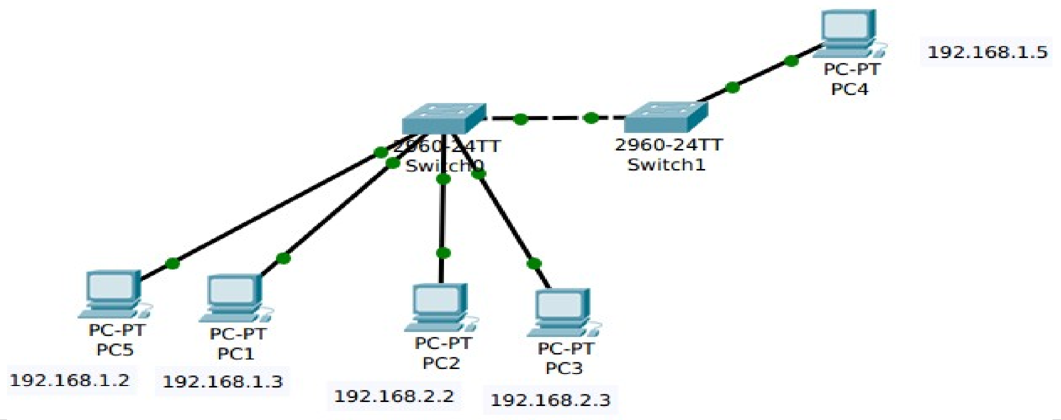


The hub has no knowledge of VLANs or 802.1Q. Traffic from hosts connected to the hub will be forwarded to the switches untagged, which in turn will place the untagged traffic into the native VLAN.

Native VLANs pose a security risk, allowing an attacker to hop to another VLAN by double-tagging a frame. This can be mitigated by changing the native VLAN to an unused or disabled VLAN. A better solution is to force trunk ports to tag native VLAN traffic - globally or on a per-trunk basis:

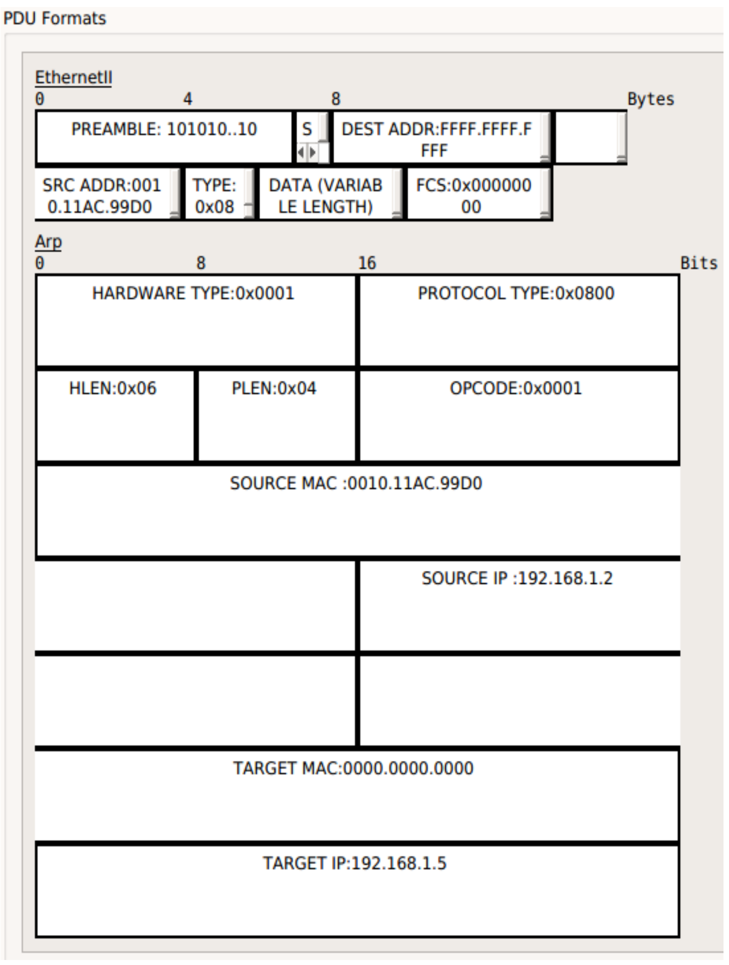
```
Switch(config)# vlan dot1q tag native
Switch(config)# interface gi2/24
Switch(config-if)# switchport trunk native vlan tag
```

i) VLANS is done by doing single broadcast network to multiple broadcast network.
ii) By default, a switch will forward both broadcasts and multicasts out every port but the originating port. However,a switch can be logically segmented into separate broadcasts domains,using virtual LANS.



ping for 192.168.1.2 to 192.168.1.5

The ARP Packet for 192.168.1.2 to switch **The ARP for switch to 192.168.1.5**



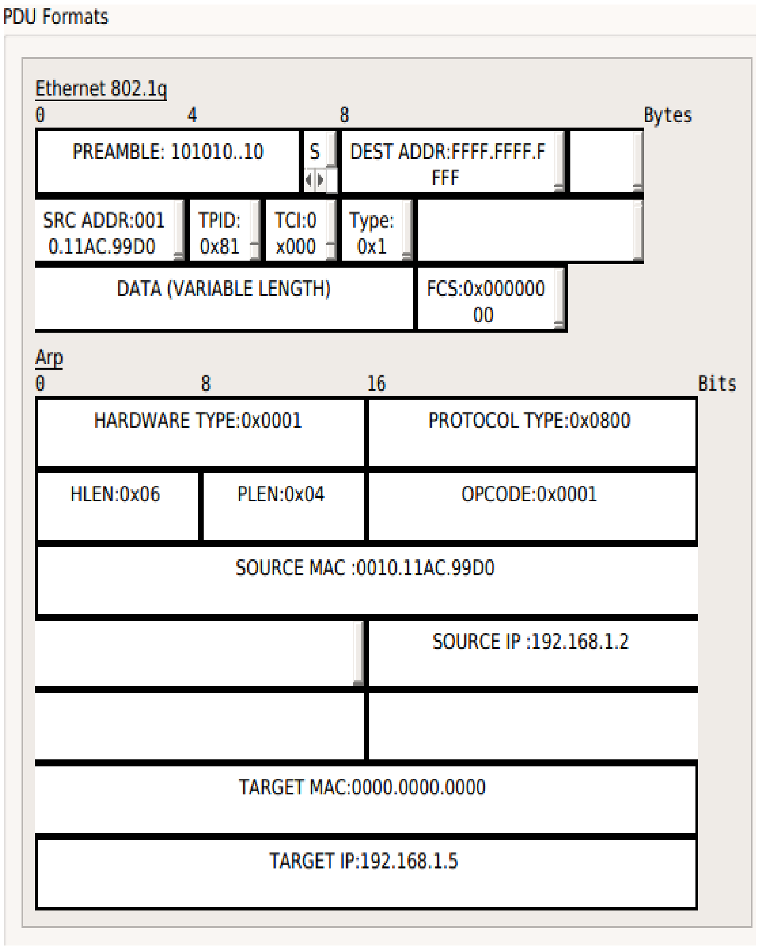
PDU Formats

EthernetII
0 4 8 Bytes

PREAMBLE: 101010..10	S	DEST ADDR:FFFF.FFFF.FFFF	
SRC ADDR:0010.11AC.99D0	TYPE: 0x08	DATA (VARIABLE LENGTH)	FCS:0x00000000

Arp
0 8 16 Bits

HARDWARE TYPE:0x0001		PROTOCOL TYPE:0x0800
HLEN:0x06	PLEN:0x04	OPCODE:0x0001
SOURCE MAC :0010.11AC.99D0		
		SOURCE IP :192.168.1.2
TARGET MAC:0000.0000.0000		
TARGET IP:192.168.1.5		



PDU Formats

Ethernet 802.1q
0 4 8 Bytes

PREAMBLE: 101010..10	S	DEST ADDR:FFFF.FFFF.FFFF		
SRC ADDR:0010.11AC.99D0	TPID: 0x81	TCI:0x000	Type: 0x1	
DATA (VARIABLE LENGTH)	FCS:0x00000000			

Arp
0 8 16 Bits

HARDWARE TYPE:0x0001		PROTOCOL TYPE:0x0800
HLEN:0x06	PLEN:0x04	OPCODE:0x0001
SOURCE MAC :0010.11AC.99D0		
		SOURCE IP :192.168.1.2
TARGET MAC:0000.0000.0000		
TARGET IP:192.168.1.5		

The host 192.168.2.2 will not communicate with the 192.168.1.5 because both are different **VLANS**

VLAN Trunking Protocol (VTP)

Maintaining a consistent VLAN database can be difficult in a large switching environment.

Cisco’s proprietary VLAN Trunking Protocol (VTP) simplifies this management - updates to the VLAN database are propagated to all switches using VTP advertisements.

VTP requires that all participating switches join a VTP domain. Switches must belong to the same domain to share VLAN information, and a switch can only belong to a single domain.

VTP Versions

There are three versions of VTP. VTP version 1 supports the standard 1 – 1005 VLAN range. VTP version 1 is also default on Catalyst switches.

VTP version 2 introduces some additional features:

- Token Ring support
- VLAN consistency checks
- Domain-independent transparent pass through

VTPv1 and **v2** are **not compatible**. The **VTP** version is dictated by the **VTP** server, discussed in detail shortly. If the **VTP** server is configured for **VTPv2**, all other switches in the **VTP** domain will change to v2 as well.

Until recently, **VTP Version 3** was supported on only limited Cisco switch platforms. VTPv3 was built to be flexible, and can forward both VLAN and other database information, such as Multiple Spanning Tree (MST) protocol.

Other enhancements provided by VTPv3 include:

- Support for the extended **1006-4094 VLAN** range.
- Support for private **VLANs.**
- Improved **VTP authentication**.
- Protection from accidental database overwrites, by using **VTP primary and secondary servers.**
- Ability to enable **VTP** on a per-port basis.

VTP Modes

A switch using VTP must operate in one of th

- **Server**
- **Client**
- **Transparent**

VTP servers are responsible for creating, deleting, or modifying entries in the **VLAN** database. Each **VTP** domain must have at least one **VTP** server, and this is the **default mode** for Cisco switches.

VTP Server Mode

1) **Default Mode.**
2) **Creates,Modifies and Deletes VLANS.**
3) **Synchronizes VLAN Configuration.**
4) **Sends and Forwards advertisements.**
5) **Saves Configuration in NVRAM.**

Servers advertise the VLAN database to all other switches in the VTP domain, including other servers. VTP servers can only advertise the standard 1-1005 VLAN range, and advertisements are only sent out **trunk** ports.

VTP clients cannot modify the VLAN database, and rely on advertisements from other switches to update VLAN information. A client will also forward VTP advertisements out every trunk port.

VTP Client Mode

1) **Can not Add,Modifies and Deletes VLANS Configurations.**
2) **Doesn't store its VLAN Configuration information in the NVRAM.Instead learns it from the Server every time it boots up**
3) **Forwards advertisements.**
4) **Do not Saves Configuration in NVRAM.**
5) **Synchronizes save in NVRAM.**

Remember: switches must be in the same VTP Domain to share and accept updates to the VLAN database. Only servers can change the VLAN database.

A VTP transparent switch maintains its own local VLAN database, and does not directly participate in the VTP domain. A transparent switch will never accept VLAN database information from another switch, even a server. Also, a transparent switch will never advertise its local VLAN database to another switch.

VTP Transparent Mode

1) **Can Add,Modifies and Deletes VLANS Configurations.**
2) **Does not Synchronize VLAN Configurations.**
3) **Forwards Advertisements.**
4) **Save Configuration in NVRAM.**

Transparent switches will pass through advertisements from other switches in the VTP domain. The VTP version dictates how the pass through is handled:
• **VTP version 1 –** the transparent switch will only pass through advertisements from the same VTP domain.
• **VTP version 2 –** the transparent switch will pass through advertisements from any VTP domain

VTP Advertisements – Revision Number

Recall that updates to the VLAN database are propagated using VTP advertisements. VTP advertisements are always sent out trunk ports, on **VLAN 1.**

VTP advertisements are marked with a 32-bit configuration revision number, to identify the most current VLAN database revision. Any change to the VLAN database increments the configuration revision number by 1. Thus, a higher number represents a newer database revision.

A **switch** will only accept an advertisement if the revision number is higher than the current VLAN database. Advertisements with a lower revision number are ignored.

Important note: While only VTP servers can change the VLAN database, VTP clients can advertise updates, to other clients and even to a server! As long as the revision number is higher, the switch will accept the update.

This can result in a newly-introduced switch advertising a blank or incorrect VLAN database to all other switches in the domain. Switch ports would then lose their VLAN memberships, resulting in a significant network outage.

This can be avoided when implementing a new switch into the VTP domain. Best practice is to configure a new switch as a VTP client, and reset its revision number to zero before deploying into a production network.

There are two methods of resetting the revision number to zero on a switch:

1. **Change the VTP domain name, and then change it back to the original name.**
2. **Change the VTP mode to transparent, and then change it back to either server or client.Transparent switches always a revision number of 0.**

VTP has fallen out of favor, due to the risk of an unintentional overwrite of the **VLAN** database. Until very recently, Cisco did not support **VTP** on the Nexus platform of switches.

VTPv3 directly addresses this risk through the use of **VTP** primary and secondary servers. Only the primary server is allowed to update the **VLAN** database on other switches. Only one primary server is allowed per domain.

VTP Advertisements – Message Types
Three message types exist for VTP advertisements:

- **Summary Advertisement**
- **Subset Advertisement**
- **Advertisement Request**

Both **VTP** servers and clients will send out a summary advertisement every 300 seconds. **Summary advertisements** contain the following information about the VTP domain:

- **VTP version**
- **Domain name**
- **Configuration revision number**
- **Time stamp**
- **MD5 digest**

Summary advertisements are also sent when a change occurs to the VLAN database. The summary is then followed with a subset advertisement, which actually contains the full, updated VLAN database.

A **subset advertisement** will contain the following information:

- **VTP version**
- **Domain name**

- **Configuration revision number**
- **VLAN IDs for each VLAN in the database**
- **VLAN-specific information, such as the VLAN name and MTU**

Important note: Switches will only accept summary and subset advertisements if the domain name and MD5 digest match. Otherwise, the advertisements are ignored.

If a switch receives a summary advertisement with a revision number higher than its own, it will send out an advertisement request. VTP servers will then respond with an updated summary and subset advertisement so that the switch can synchronize to the most current VLAN database.

A switch that is reset or newly joined to the VTP domain will also send out an advertisement request.

Configuring VTP

By default, a switch is in VTP server mode, and joined to a blank domain labeled NULL.

To change the VTP domain name:

```
Switch(config)# vtp domain MYDOMAIN
```

Note that the domain name is case sensitive. To configure the **VTP** mode:

```
Switch(config)# vtp mode server
Switch(config)# vtp mode client
Switch(config)# vtp mode transparent
```

The VTP domain can be secured using a password:

```
Switch(config)# vtp password P@SSWORD!
```

The password is also case sensitive. All switches participating in the VTP domain must be configured with the same password. The password is hashed into a **16-byte MD5 digest.**

Cisco switches use VTP version 1 by default, which is not compatible with **VTPv2.** The **VTP** version is dictated by the **VTP server**, and if the server is configured for **VTPv2,** all other switches in the VTP domain will change to v2 as well.

```
Switch(config)# vtp version 2
```

To view status information about VTP:

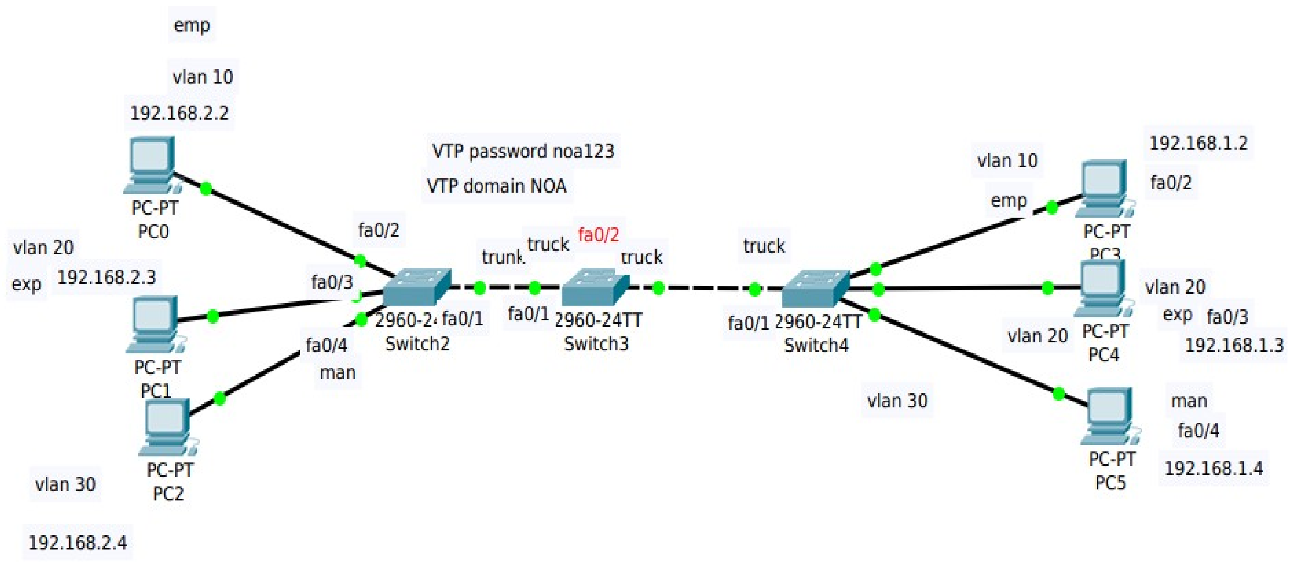
```
Switch# show vtp status
```

```
VTP Version                      : 2
Configuration Revision           : 42
Maximum VLANs supported locally  : 1005
Number of existing VLANs         : 7
VTP Operating Mode               : Server
VTP Domain Name                  : MYDOMAIN
VTP Pruning Mode                 : Disabled
VTP V2 Mode                      : Enabled
VTP Traps Generation             : Disabled
MD5 digest                       : 0x42 0x51 0x69 0xBA 0xBE 0xFA 0xCE 0x34
Configuration last modified by 0.0.0.0 at 6-22-14 4:07:52
```

To view VTP statistical information and error counters:

```
Switch# show vtp counters
```

vlan and vtp



VTP Summary Advertisement

Version:1	CODE : 1	Followers : 1	MGT Domain Length : 0x03
Management domain name : noa			
Configuration Revision no :7			
Update timer :0.0.0.0			
Update timestamp :3-1-93 00:11:01			
MD5 DIGEST			

SNAP:Source Network Access Protocol

DNAP:0xaa	SSAP : 0xaa	Control byte :3
OUI : 0x00000c		PID :0x2003

Ethernet 802.1q

Premeable:101010...10	start delimiter	Dest ADDR : 0100.0CCC.CCCC.CCCC				
SRC ADDR:000A.F3E0.5901	TPID:0x8100	TCI:0X0001	Type:0x01	DATA (Variable Length)	FCS	

Following are the fields in an 802.1Q VLAN tag.

• **TPID (Tag Protocol Identifier, 16 bits): TPID (Tag Protocol Identifier)** is globally and always have a value of 0x8100 to signify an 802.1Q tag.

• **Priority (3 bits):** The Priority field is used by 802.1Q to implement Layer 2 quality of service (QoS).

• **CFI (Canonical Format Identifier, 1 bit):** The CFI (Canonical Format Identifier) bit is used for compatibility purposes between Ethernet and Token Ring.

• **VLAN ID (12 bits):** The VID field is used to distinguish between VLANs on the link.

Dest ADDR : It is the multicast address of the VTP

Vtp Subset Advertisement

Version:1	Code : 2	Sequence num :1	MGT DOMAIN :0x03
Management domain name:NOA			
Revision no : 7			

VTP all the VLAN Information

VLAN INFO LEN:	STATUS:0	VLAN TYPE:1 ethernet	VLAN name Length : 3
VLAN ID : 1			MTU Size Revision Num : 19
802.1q INDEX ID			
VLAN NAME : emp			

SNAP

DSAP:0Xaa	SSAP:0xaa	Control Byte :3
OUI : 0x00000C		PID :0x2003

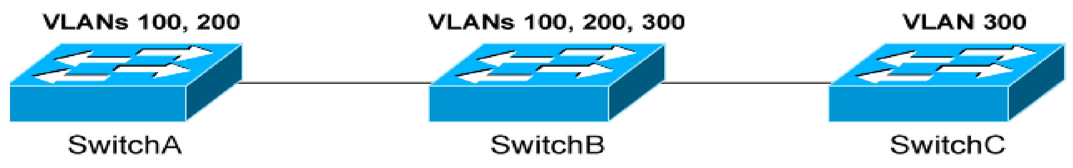
Ethernet 802.1q

Preamble:101010...10	Start Delimiter	Dest ADDR :0100.0CCC.CCCC.CCC				
SRC ADDR:000A.F3E0.5901	TPID:0x8100	TCI:0x0001	TYPE:0x01	DATA(Variable)	FCS	

VTP Purning

Recall that Layer-2 switches belong to only one broadcast domain. A Layer-2 switch will thus forward both broadcasts and multicasts out every port in the same VLAN but the originating port. This includes sending out broadcasts out trunk ports to other switches, which will in turn flood that broadcast out all ports in the same VLAN.

VTP pruning eliminates unnecessary broadcast or multicast traffic throughout the switching infrastructure. Consider the following example:



Assume that a host is connected to SwitchB, in VLAN 300. If the host sends out a broadcast, SwitchB will forward the broadcast out every port in VLAN 300, including the trunk ports to SwitchA and SwitchC. Both SwitchA and SwitchC will then forward that broadcast out every port in VLAN 300.

However, SwitchA does not have any ports in VLAN 300, and will drop the broadcast. Thus, sending the broadcast to SwitchA is a waste of bandwidth.

VTP pruning allows a switch to learn which VLANs are active on its neighbors. Thus, broadcasts are only sent out the necessary trunk ports where those VLANs exist. In the preceding example, pruning would prevent VLAN 300 broadcasts from being sent to SwitchA, and would prevent VLAN 100 and 200 broadcasts from being sent to SwitchC.

VTP pruning is disabled by default on IOS switches. VTP pruning must be enabled on a server, and will be applied globally to the entire VTP domain:

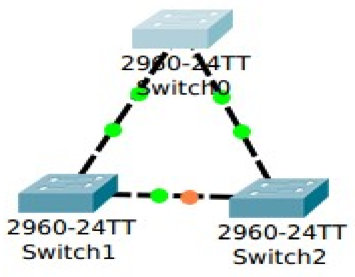
```
Switch(config)# vtp pruning
```

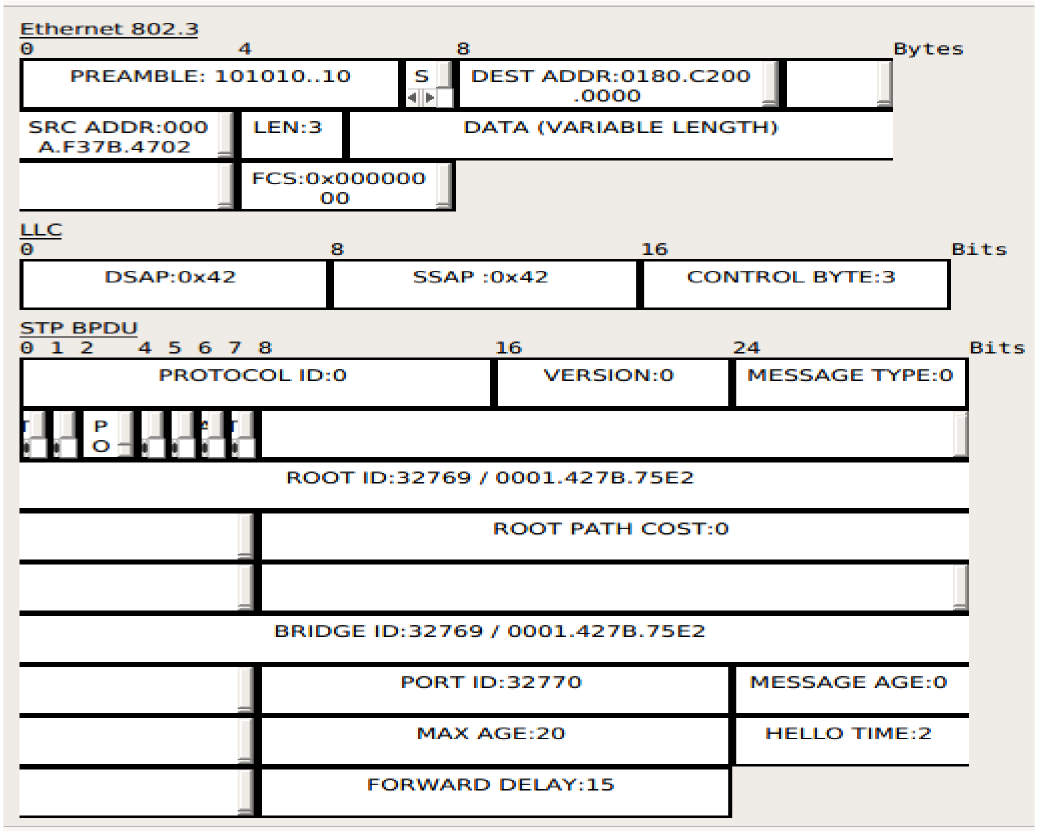
Both VLAN 1 and the system VLANs 1002-1005 are never eligible for pruning. To manually specify which VLANs are pruning eligible on a trunk:

```
Switch(config)# interface gi2/24
Switch(config-if)# switchport trunk pruning vlan 2-10
Switch(config-if)# switchport trunk pruning vlan add 42
Switch(config-if)# switchport trunk pruning vlan remove 5
Switch(config-if)# switchport trunk pruning vlan except 100-200
Switch(config-if)# switchport trunk pruning vlan none
```

Spanning Tree Protocol

1) Spanning Tree Protocol Stop the loops which occurs when you have multiple links between switchs.
2) STP stops avoiding Broadcast Storms,Multiple Frame Copies & Database instability.
3) STP is a open standard IEEE 802.1D.
4) STP is enabled by default on all cisco catalyst switches.




Ethernet 802.3
0 4 8 Bytes
PREAMBLE: 101010..10
S
DEST ADDR:0180.C200 .0000
SRC ADDR:000 A.F37B.4702
LEN:3
DATA (VARIABLE LENGTH)
FCS:0x000000 00

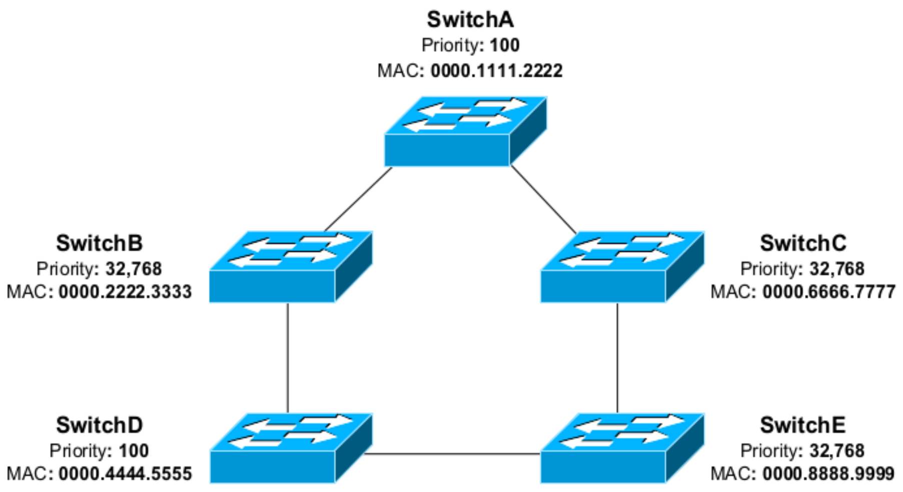
LLC
0 8 16 Bits
DSAP:0x42
SSAP :0x42
CONTROL BYTE:3

STP BPDU
0 1 2 4 5 6 7 8 16 24 Bits
PROTOCOL ID:0
VERSION:0
MESSAGE TYPE:0
P O
ROOT ID:32769 / 0001.427B.75E2
ROOT PATH COST:0
BRIDGE ID:32769 / 0001.427B.75E2
PORT ID:32770
MESSAGE AGE:0
MAX AGE:20
HELLO TIME:2
FORWARD DELAY:15

How STP works:

1) Selecting the Root Bridge.
2) Selecting the Root Port.
3) Selectiong Designated Port & Non Designated Port.

Consider the following example:



i) Selecting the Root Bridge :

1) The bridge with the best (Lowest) bridge ID
   Bridge ID = Priority + MAC address of the switch.
2) Out of all the switches in the network,one is elected as a root bridge that becomes the focal point in the network .
3) Every LAN will have only one Root bridge all the remaining switches will be considered as Non-root bridge.
4) They keep on sending the hello message we call them as BPDU messages every switch will exchange there own bridge ID information by using some bridge protocol
data unit messages for every 2 secs.

Bridge ID = Priority value should be least value is considered as best.

What if all the bridges having the same Priority vlaue?
Ans) Bridge ID = Priority + MAC address of the switch.

Initially every switch will advertise BPDU root bridge and bridge ID as it s own MAC address.

There will be only one root bridge all the remaining bridges will be consider as non root bridges.

ii) Selecting the Root port:

1) Shortest path to the Root bridge.
2) Every Non-root bridge looks the best way to go Root-bridge
   i) Least cost (speed)
   ii) The Lowest forwarding switch ID (priority + MAC)
   iii) Lowest forwarding physical port number
3) For every non-root bridge there is only one root port

The Designated Port

STP Port Cost:

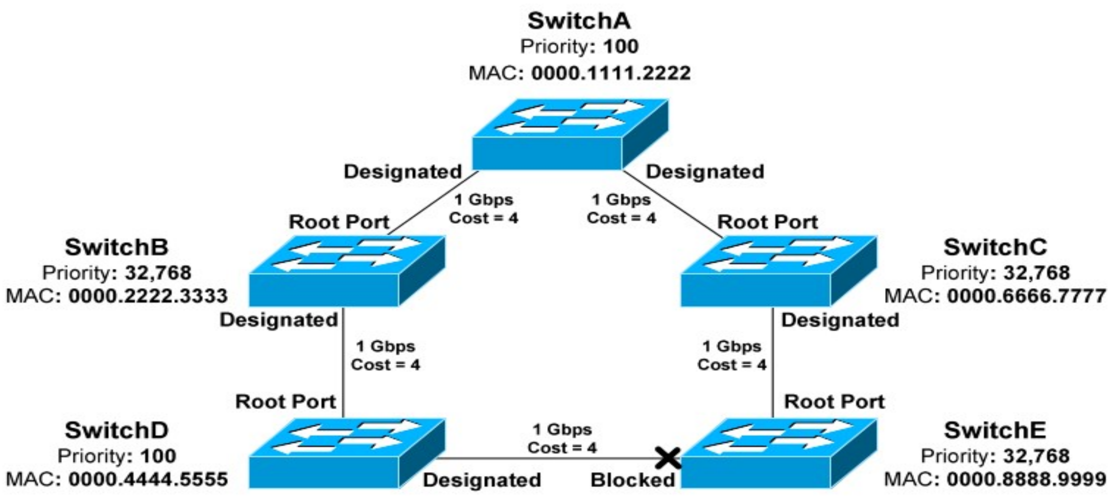
Link Speed (Bandwidth)	Port Cost
10 mbps	100
100 mbps	19
1 gbps	4
10 gbps	2

If there are 20 switches among them 1 is root bridge and the remaining all are non root bridges

The root bridge will always be in the forwarding state and does not have blocking state The root port which was selected also be in the forwarding state

Each 1Gbps link has a path cost of 4. SwitchA has a cumulative path cost of 0, because it is the Root

Bridge. Thus, when SwitchA sends out BPDU’s, it advertises a root path cost of 0.



SwitchB has two paths to the Root Bridge:
`
- A direct connection to SwitchA, with a path cost of 4.
- Another path through SwitchD, with a path cost of 16.

The lowest cumulative path cost is considered superior, thus the port directly connecting to SwitchA will become the root port. A BPDU advertising a higher path cost is often referred to as an inferior BPDU.
SwitchD also has two paths to the Root Bridge:
- A path through SwitchB, with a path cost of 8.
- A path through SwitchE, with a path cost of 12.
- The port to SwitchB is preferred, and will become the root port.

Recall that the Root Bridge will advertise BPDU’s with a path cost of 0. As the downstream switches receive these BPDU’s, they will add the path cost of the receiving port, and then advertise the cumulative cost to neighbors.

For example, SwitchC will receive a BPDU with a path cost of 0 from SwitchA, which is the Root Bridge. SwitchC will add the path cost of its receiving port, and thus SwitchC’s cumulative path cost will be 4.

SwitchC will advertise a path cost of 4 to SwitchE, which will add the path cost of its receiving port. SwitchE’s cumulative path cost will thus be 8. Path cost can be artificially adjusted on a per-port basis:

Path cost can be artificially adjusted on a per-port basis:
```
SwitchD(config)# int gi2/22
SwitchD(config-if)# spanning-tree vlan 101 cost 42
```

The third step in the STP convergence process is to identify designated ports. A single designated port is identified for each network segment. This port is responsible for forwarding BPDUs and frames to that segment.

If two ports are eligible to become the designated port, then there is a loop. One of the ports will be placed in a blocking state to eliminate the loop.

Similar to a root port, the designated port is determined by the lowest cumulative path cost leading the Root Bridge. A designated port will never be placed in a blocking state, unless there is a change to the switching topology and a more preferred designated port is elected.
Note: A port can never be both a designated port and a root port.

Ports on the Root Bridge are never placed in a blocking state. Thus, the two ports off of SwitchA will automatically become designated ports.

Remember, every network segment must have one designated port, regardless if a root port already exists on that segment. Thus, the network segments between SwitchB and SwitchD, and between

SwitchC and SwitchE, both require a designated port. The ports on SwitchD and Switch E have already been identified as root ports, thus the ports on Switch B and C will become the designated ports.

Because two ports on this segment are eligible to become the designated port, STP recognizes that a loop exists. One of the ports must be elected as the designated port, and the other must be placed in a blocking state.

Normally, whichever switch has the lowest cumulative path cost will have its port become designated. The switch with the highest path cost will have its port blocked.

In the above example, there is a tie in cumulative path cost. Both SwitchD and Switch have a path cost of 12 to reach the Root Bridge on that segment.

The lowest Bridge ID is used as the tiebreaker. SwitchD has a priority of 100, and SwitchE has the default priority of 32,768.

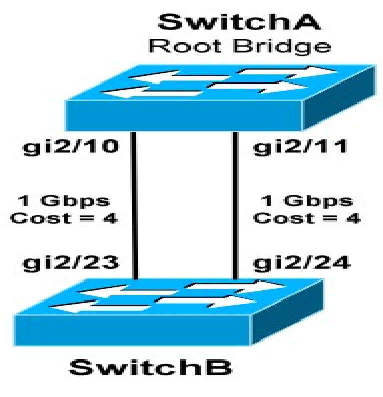
Thus, the port on SwitchD will become the designated port. The port on SwitchE will be placed in a blocking state.

As with electing the Root Bridge, if there is a tie in priority, the lowest \ MAC address is used as the tie breaker.

Remember: Any port not elected as a root or designated port will be placed in a blocking state.

Port ID

When electing root and designated ports, it is possible to have a tie in both
path cost and Bridge ID. Consider the following example:



The bandwidth of both links is equal, thus both ports on SwitchB have an equal path cost to the Root Bridge. Which port will become the root port then? Normally, the lowest Bridge ID is used as the tiebreaker, but that is not possible in this circumstance.

Port ID is used as the final tiebreaker, and consists of two components:

- 4-bit port priority
- 12-bit port number, derived from the physical port number

By default, the port priority of an interface is 128, and a lower priority is preferred. If there is a tie in priority, the lowest port number is preferred.

The sender port ID determines the tie break, and not the local port ID. In the above example, SwitchB must decide whether gi2/23 or gi2/24 becomes the root port. SwitchB will observe BPDU’s from SwitchA, which will contain the port ID’s for gi2/10 and gi2/11.

If priorities are equal, the sender Port ID from gi2/10 is preferred, due to the lower port number. Thus, gi2/23 on SwitchB will become the root port.

The port number is a fixed value, but port priority can be changed on a per- interface basis:

```
Switch(config)# int gi2/11
Switch(config-if)# spanning-tree vlan 101 port-priority 32
```

Remember: Port ID is the last tiebreaker STP will consider. STP determines root and designated ports using the following criteria, in order:

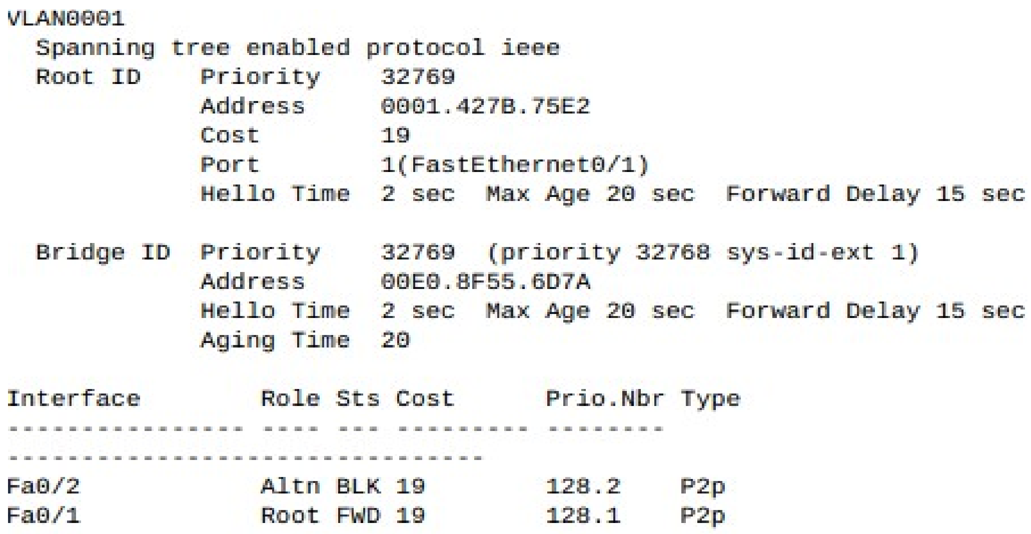
- **Lowest path cost to the Root Bridge**
- **Lowest bridge ID**
- **Lowest sender port ID**

Bridge Protocol Data Unit (BPDU):

i) All Switches exchange information through the BPDU's
ii) Hello = BPDU'S are sent for every 2 sec
ii) Max age (dead) = 20 sec
iv) Forward Delay (listening/learning time) = 15 sec
v) A BPDU contains information regarding ports,switches,port priority and address.
vi) Every root bridge by default is selected as the root bridge.

Sw1 will say i am the root bridge
Sw2 will say i am the root bridge

Sw2 will compare its own bridge id and neighbour bridge id and say i am not the root
Once they select the root bridge the root bridge will sent the BPDU messages
vii) When ever if you power on switch for the first then the switch will go into the listening state – 15 sec where it will listen to the BPDU messages and decides the root bridge and decides the root port and forwarding ports that will happen in the frame of 30 seconds.



```
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0001.427B.75E2
             Cost        19
             Port        1(FastEthernet0/1)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     00E0.8F55.6D7A
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- --------
------------------------------
Fa0/2            Altn BLK 19        128.2    P2p
Fa0/1            Root FWD 19        128.1    P2p
```

Spanning Tree Protocol Port States

- **i) Blocking 20 sec or No Limit**
- **ii) Listening 15 sec**
- **iii) Learning 15 sec**
- **iv) Forwarding NO Limit**
- **v) Disable No Limit**

i) Blocking

The Switch Ports will go into a blocking state at the time of election process, when a switch receives a **BPDU** on a port that indicates a better path to the **Root Switch (Root bridge),** and if a port is not a **Root port** or a **Designated port.**

A port in the blocking state does not participate in frame forwarding and also discards frames received from the attached network segment. During blocking state, the port is only listening to and processing **BPDUs** on its interfaces. After 20 Sw2 will compare its own bridge id and neighbour bridge id and say i am not the root

Once they select the root bridge the root bridge will sent the BPDU messages

vii) When ever if you power on switch for the first then the switch will go into the listening state – 15 sec where it will listen to the BPDU messages and decides the root bridge and decides the root port and forwarding ports that will happen in the frame of 30 seconds.seconds, the switch port changes from the blocking state to the listening state.

ii) Listening 15 sec

After blocking state, a **Root port** or a **Designated port** will move to a listening state. All other ports will remain in a blocked state. During the listening state the port discards frames received from the attached network segment and it also discards frames switched from another port for forwarding. At this state, the port receives **BPDUs** from the network segment and directs them to the switch system module for processing. After 15 seconds, the switch port moves from the listening state to the learning state.

Iii) Learning 15 sec

A port changes to learning state after listening state. During the learning state, the port is listening for and processing **BPDUs**. In the listening state, the port begins to process user frames and start updating the MAC address table. But the user frames are not forwarded to the destination. After 15 seconds, the switch port moves from the learning state to the forwarding state.

iv) Forwarding

A port in the forwarding state forwards frames across the attached network segment. In a forwarding state, the port will process **BPDUs,** update its MAC Address table with frames that it receives, and forward user traffic through the port. Forwarding State is the normal state. Data and configuration messages are passed through the port, when it is in forwarding state.

V) Disable

A port in the disabled state does not participate in frame forwarding or the operation of **STP** because a port in the disabled state is considered non-operational

If there is any problem in forwarding port the blocking port will become forwarding it waits 20 secs,15 Lisening State,15 Learning State

Timers

Important **Spanning tree protocol** timers are hello timer, forward delay timer and max age timer and their default values are listed below**.**

i) Hello timer

The hello timer is the time interval between each **Bridge Protocol Data Unit (BPDUs)** that is sent on a port. Defaut **Spanning Tree Protocol (STP)** hello timer is 2 seconds. You can adjust **Spanning Tree Protocol (STP)** hello timer to any value between 1 and 10 sec.

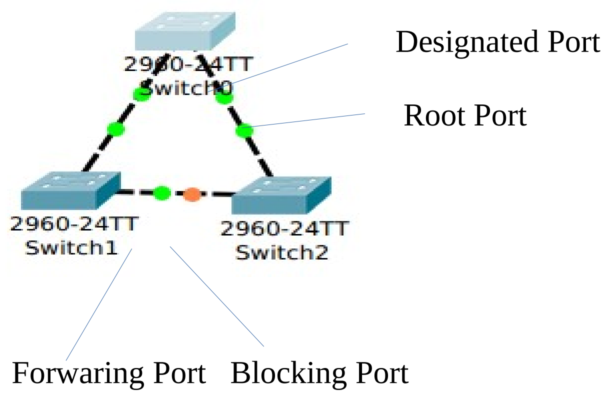
ii) Forward delay timer

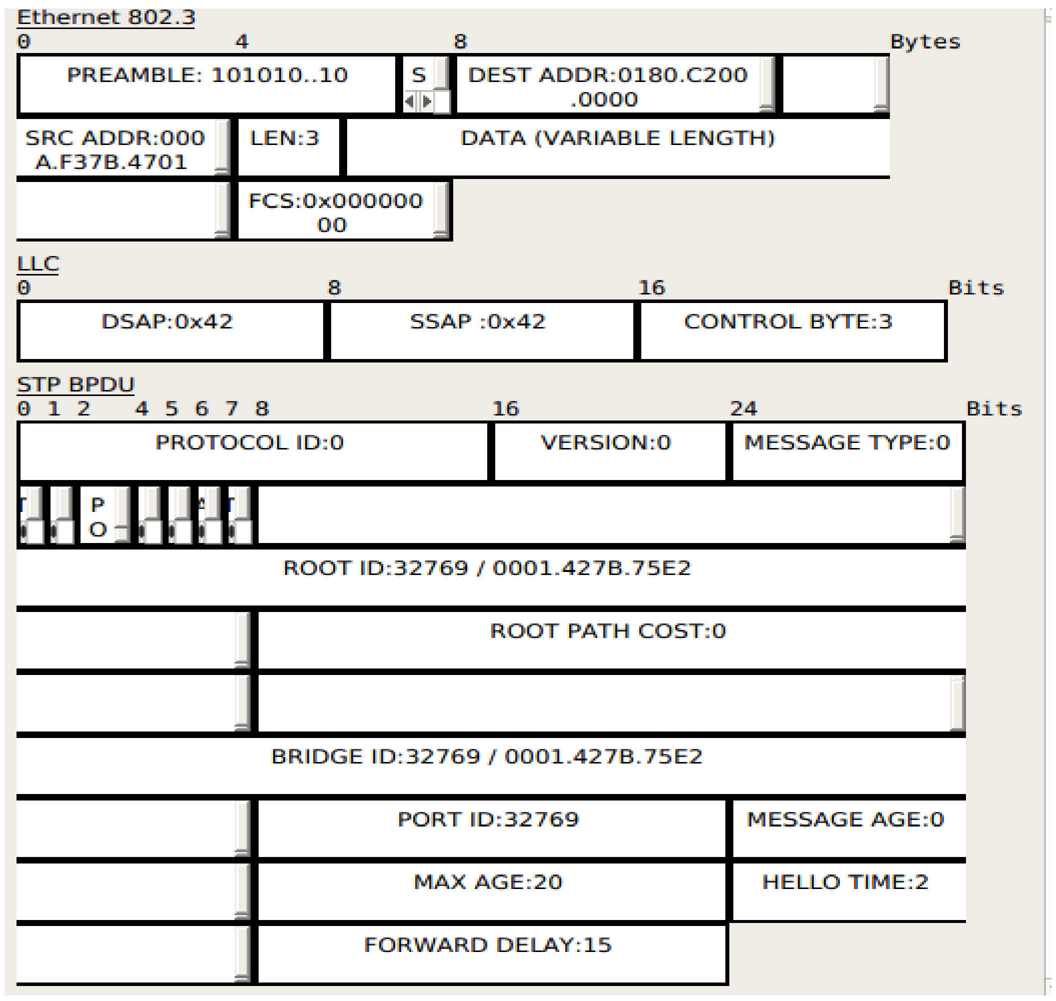
The forward delay timer is the time interval that is spent in the listening and learning state. Default **Spanning Tree Protocol (STP)** forward delay timer is 15 seconds. You can adjust the **Spanning Tree Protocol (STP)** forward delay timer to any value between 4 and 30 seconds.

Iii) Max age timer

The max age timer controls the maximum length of time interval that a **Spanning Tree Protocol (STP)** Switch port saves its configuration **Bridge Protocol Data Unit (BPDU)** information. Default max age timer is 20 seconds. You can tune the **Spanning Tree Protocol (STP)** max age timer to any value between 6 and 40 sec.

Bridge Protocol Data Units





```
Ethernet 802.3
0                 4                8                              Bytes
| PREAMBLE: 101010..10 | S | DEST ADDR:0180.C200 .0000 |        |
| SRC ADDR:000 A.F37B.4701 | LEN:3 | DATA (VARIABLE LENGTH) |
|                          | FCS:0x00000000 |

LLC
0                        8                        16               Bits
| DSAP:0x42              | SSAP :0x42             | CONTROL BYTE:3 |

STP BPDU
0 1 2   4 5 6 7 8                  16              24              Bits
| PROTOCOL ID:0                    | VERSION:0     | MESSAGE TYPE:0 |
| | | P O | | | | |                                                |
|                 ROOT ID:32769 / 0001.427B.75E2                    |
|                 | ROOT PATH COST:0                                |
|                 |                                                 |
|                 BRIDGE ID:32769 / 0001.427B.75E2                  |
|                 | PORT ID:32769           | MESSAGE AGE:0         |
|                 | MAX AGE:20              | HELLO TIME:2          |
|                 | FORWARD DELAY:15        |
```

Protocol Identifier : A two-octet field that identifies the Spanning Tree Protocol (STP).The field is set to a value of 0000 0000 0000 0000.
Protocol Version Identifier : A one-octet field that identifies the current version of the Protocol being used, which was specified in the previous field. For the STP, this field is set to a value of 0000 0000
MESSAGE TYPE : A one-octet field that identifies the type of BPDU being sent. This is set to a value of 0000 0000 for a configuration BPDU.
Flags : A one-octet field that contains flags used in response to a topology change notification

BPDU.Bit 1 is the topology change flag.It is used by the root to tell all switches to speed up their aging timers for their filtering databases.

The Flags field includes one of the following:

- 0... .... Topology change (TC) bit, which signals a topology change

Topology change acknowledgment (TCA) bit, which is set to acknowledge receipt of a configuration message with the TC bit set

.0.. .... Agreement : NO
..1. .... Forwarding : YES
...1 .... Learning : YES
.... 11.. Port Role : Designated (3)
.... ..0. Proposol : NO
....1 Topology Change : YES

Root ID :

The Root ID field indicates the root bridge by listing its 2-byte priority followed by its 6-byte ID.

Root Path Cost :

The Root Path Cost field indicates the cost of the path from the bridge sending the configuration message to the root bridge.

Bridge ID :

The Bridge ID field indicates the priority and ID of the bridge sending the message.

Port ID :

The Port ID field indicates the port number (IEEE or Cisco Spanning-Tree Protocol BPDU) or the ring and bridge number (IBM Spanning-Tree Protocol BPDU) from which the configuration message was sent. This field allows loops created by multiple attached bridges to be detected and corrected

Message Age :

The Message Age field indicates the amount of time that has elapsed since the root sent the configuration message on which the current configuration message is based.

Maximum Age :

The Maximum Age field indicates when the current configuration message should be deleted.

Hello Time :

The Hello Time field indicates the time between root bridge configuration messages.

Forward Delay :

The Forward Delay field indicates the length of time that bridges should wait before transitioning to a new state after a topology change. If a bridge transitions too soon, it is possible that not all network links will be ready to change their state and loops can result.

Rapid Spanning Tree Protocol

1) 802.1Q is a Standard way of speeding STP convergence.
2) Inbuilt features of portfast,uplink fast,backbonefast.
3) Path calculations remains same as a STP.

RSTP Port States

Discarding (Disable,Blocking,Listening)
Learning
Forwarding

Discarding State

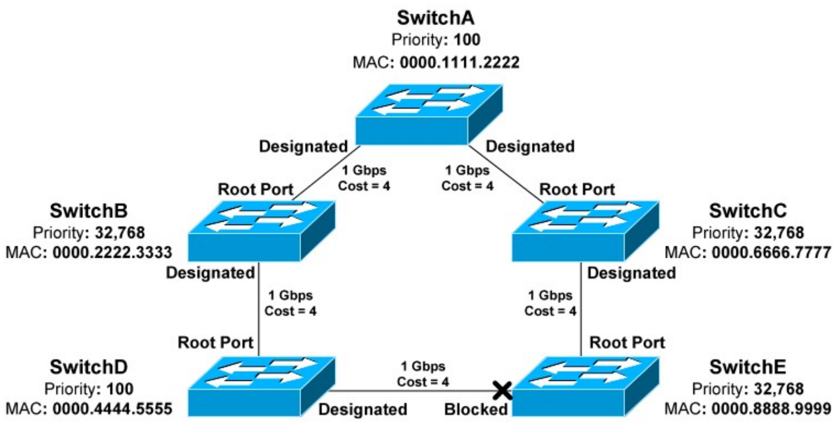
Frames are dropped no addresses are learned (link down/blocking/during synchronization)

Learning State

Frames are dropped but addresses are learned -> Maintain MAC Table

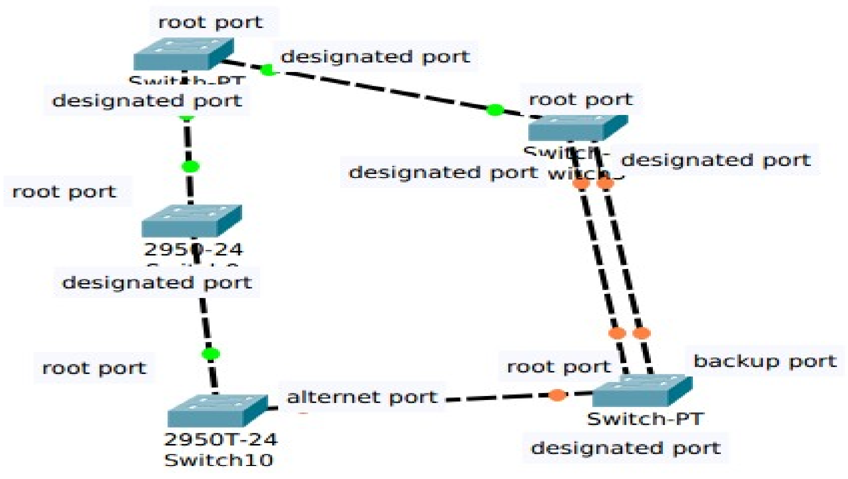
Forwarding

Frames are forwarded



RSTP Port Rules

1) Alternate Port
2) Root Port
3) Designated Port
4) Backup port
5) Edge port



Root Port : Port on each switch that has the best path cost to the root bridge . A Switch can only have one root port.

Designated Port : Non-root port that represents the best path cost for each network segment to the root bridge

Backup port : The backup port applies only when a single switch has two links to the same segment (collision domain).
2) To have two links to the same collisions domain,the switch must be attached to a hub.
3) A backup to the designated port.
4) Multiple links attached to the same network segment.
5) Activates if parimary designated fails.

Edge Port : Equivalent to portfast in STP.
Connected only to an end user
Maintained edge status as long as no BPDU received (with BPDU filter)

BPDU Difference in STP

1) In regular STP,BPDUs are originated by the root and relayed by each switch
2) In RSTP,each switch originates BPDUs,wheather or not it receives a BPDUs on its root port. PVST is done by Rapid PSTP + on catalyst switches
3) Hello = 2 sec   Dead = 6 sec

If any thing goes wrong all the switches will send a topology to all switches and do synchronization by sending proposol and agreement to make spanning tree protocol much faster than normal spanning tree protocol.

Improving STP Convergence

In many environments, a 30 second outage for every topology change is unacceptable. Cisco developed three proprietary features that improve STP convergence time:
• PortFast
• UplinkFast
• BackboneFast
Each feature will be covered in detail in the following sections.

PortFast

By default, all ports on a switch participate in the STP topology. This includes any port that connects to a host, such as a workstation. In most circumstances, a host represents no risk of a loop.

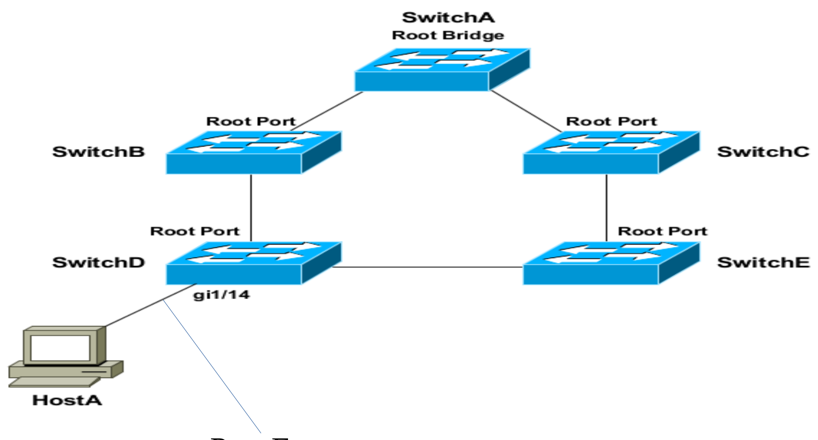
The host port will transition through the normal STP states, including waiting two forward delay times. Thus, a host will be without network connectivity for a minimum of 30 seconds when first powered on.

This is not ideal for a couple reasons:
• Users will be annoyed by the brief outage.
• A host will often request an IP address through DHCP during bootup. If the switch port is not forwarding quickly enough, the DHCP request may fail.
• Devices that boot from network may fail as well.

PortFast allows a switch port to bypass the usual progression of STP states. The port will instead transition from a blocking to a forwarding state immediately, eliminating the typical 30 second delay.

PortFast should only be enabled on ports connected to a host. If enabled on a port connecting to a switch or hub, any loop may result in a broadcast storm.

Note: PortFast does not disable STP on a port - it merely accelerates STP convergence. If a PortFast-enabled port receives a BPDU, it will transition through the normal process of STP states.



Port Fast
(normally each port goes to Listening and Learning state for 30 sec
to remove we can enable at that perticular port **PORTFAST** and its **disable STP**)

PortFast provides an additional benefit. Remember that a switch will generate a TCN if a port transitions to a forwarding or blocked state. This is true even if the port connects to a host device, such as a workstation.

Thus, powering on or off a workstation will cause TCNs to reach the Root Bridge, which will send out configuration BPDUs in response. Because the switching topology did not technically change, no outage will occur.

However, all switches will reduce the CAM aging timer to 15 seconds, thus purging MAC addresses from the table very quickly. This will increase frame flooding and reduce the efficiency and performance.

PortFast eliminates this unnecessary BPDU traffic and frame flooding. A TCN will not be generated for state changes on a PortFast-enabled port.
Portfast is disabled by default. To enable **PortFast** on a switch port:

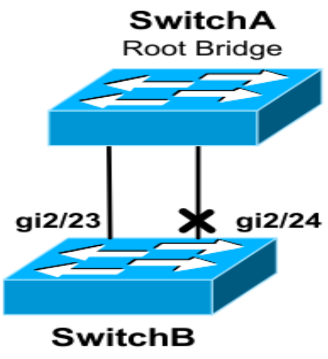
```
SwitchD(config)# int gi1/14
SwitchD(config-if)# spanning-tree portfast
```

PortFast can also be globally enabled for all interfaces:

```
SwitchD(config)# spanning-tree portfast default
```

UplinkFast

Often, a switch will have multiple uplinks to another upstream switch:



If the links are not bundled using an EtherChannel, at least one of the ports will transition to a blocking state to eliminate the loop. In the above example, port gi2/24 was placed into a blocking state on SwitchB.

Normally, if the root port fails on the local switch, STP will need to perform a recalculation to transition the other port out of a blocking state. At a minimum, this process will take 30 seconds.

UplinkFast allows a blocking port to be held in a standby state. If the root port fails, the blocking port can immediately transition to a forwarding state. Thus, UplinkFast improves convergence time for direct failures in the STP topology.

If multiple ports are in a blocking state, whichever port has the lowest root path cost will transition to forwarding.

UplinkFast is disabled by default, and must be enabled globally for all VLANs on the switch:

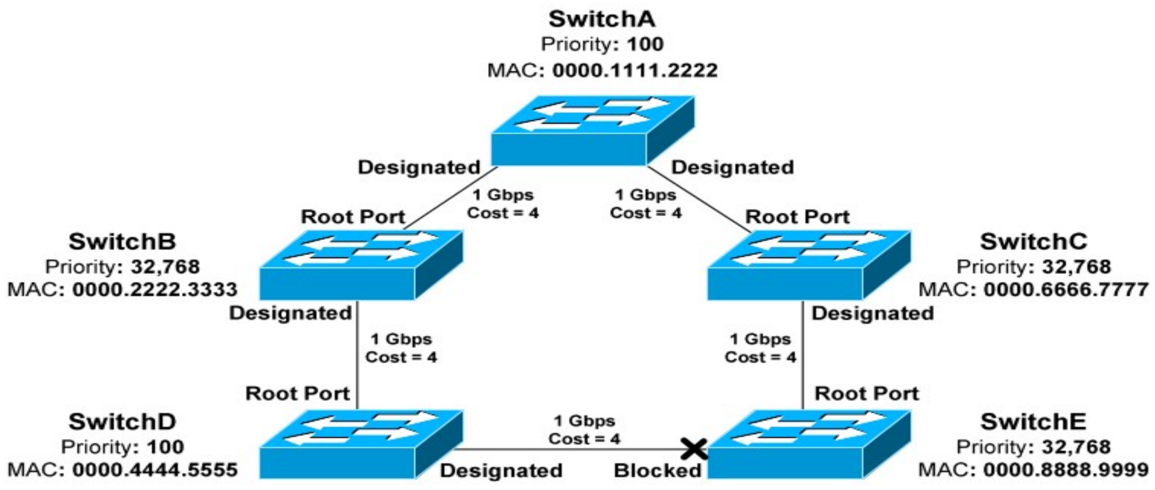
```
Switch(config)# spanning-tree uplinkfast
```

UplinkFast functions by tracking all possible links to the Root Bridge. Thus, UplinkFast is not supported on the Root Bridge. In fact, enabling this feature will automatically increase a switch’s bridge priority to 49,152.

UplinkFast is intended for the furthest downstream switches in the STP topology.

BackboneFast

UplinkFast provides faster convergence if a directly-connected port fails. In contrast, BackboneFast provides improved convergence if there is an indirect failure in the STP topology.



If the link between SwitchB and SwitchA fails, SwitchD will eventually recalculate a path through SwitchE to reach the Root Bridge. However, SwitchD must wait the max age timer before purging SwitchB's superior BPDU information. By default, this is 20 seconds.

BackboneFast allows a switch to bypass the max age timer. The switch will accept SwitchE's inferior BPDU's immediately. The blocked port on SwitchE must still transition to a forwarding state. Thus, BackboneFast essentially reduces total convergence time from 50 seconds to 30 seconds for an indirect failure.

This is accomplished by sending out Root Link Queries (RLQs). The Root Bridge will respond to these queries with a RLQ Reply:

- If a RLQ Reply is received on a root port, the switch knows that the root path is stable.
- If a RLQ Reply is received on a non-root port, the switch knows that the root path has failed. The max age timer is immediately expired to allow a new root port to be elected.

BackboneFast is a global command, and should be enabled on every switch:

```
Switch(config)# spanning-tree backbonefast
```

Protecting STP

STP is vulnerable to attack for two reasons:

- STP builds the topology by accepting BPDUs from neighboring switches.
- The Root Bridge is always determined by the lowest Bridge ID.

A switch with a low priority can be maliciously or inadvertently installed on the network, and then elected as the Root Bridge. STP will reconverge, often resulting in instability or a suboptimal topology.

Cisco implemented three mechanisms to protect the STP topology:

- **Root Guard**
- **BPDU Guard**

- **BPDU Filtering**

All three mechanisms are configured on a per-port basis, and are disabled by default.

Root Guard

Root Guard prevents an unauthorized switch from advertising itself as a Root Bridge. If a BPDU superior to the Root Bridge is received on a port with Root Guard enabled, the port is placed in a root-inconsistent state.

In this state, the port is essentially in a blocking state, and will not forward frames. The port can still listen for BPDUs.

Root Guard is enabled on a per-port basis, and is disabled by default:

```
Switch(config)# interface gi1/14
Switch(config-if)# spanning-tree guard root
```

To view all ports that have been placed in a root-inconsistent state:

```
Switch# show spanning-tree inconsistentports
```

Root Guard can automatically recover. As soon as superior BPDUs are no longer received, the port will transition normally through STP states.

BPDU Guard

Recall that PortFast allows a switch port to bypass the usual progression of **STP states**. However, PortFast does not disable STP on a port - it merely accelerates **STP convergence.** However, a PortFast-enabled port will still accept **BPDUs.**

PortFast should only be enabled on ports connected to a host. If enabled on a port connecting to a switch, any loop may result in a broadcast storm.

To prevent such a scenario, BPDU Guard can be used in conjunction with PortFast. Under normal circumstances, a port with Port Fast enabled should never receive a BPDU, as it is intended only for hosts.

BPDU Guard will place a port in an errdisable state if a BPDU is received, regardless if the BPDU is superior or inferior. The STP topology will not be impacted by another switch that is inadvertently connected to that port.

BPDU Guard should be enabled on any port with PortFast enabled. It is disabled by default, and can be enabled on a per-interface basis:

```
Switch(config)# interface gi1/14
Switch(config-if)# spanning-tree bpduguard enable
```

If **BPDU Guard** is enabled **globally**, it will only apply to PortFast ports:

```
Switch(config)# spanning-tree portfast bpduguard default
```

An interface can be manually recovered from an errdisable state by performing a shutdown and then no shutdown:

```
Switch(config)# interface gi1/14
Switch(config-if)# shutdown
Switch(config-if)# no shutdown
```

BPDUs will still be sent out ports enabled with **BPDU Guard.**

BPDU Filtering

BPDU Filtering prevents BPDUs from being sent out a port, and must be enabled in conjunction with PortFast.
If a BPDU is received on a port, **BPDU Filtering** will react one of two ways, depending on how it was configured.

- If filtering is enabled globally, a received **BPDU** will disable PortFast on the port. The port will then transition normally through the STP process.
- If filtering is enabled on a per-interface basis, a received BPDU is ignored.

Great care must be taken when manually enabling BPDU Filtering on a port. Because the port will ignore a received BPDU, STP is essentially disabled. The port will neither be err-disabled nor progress through the STP process, and thus the port is susceptible to loops.
If **BPDU Filtering** is enabled **globally**, it will only apply to PortFast ports:

```
Switch(config)# spanning-tree portfast bpdufilter default
```

To enable BPDU Filtering on a per-interface basis:

```
Switch(config)# interface gi1/15
Switch(config-if)# spanning-tree bpdufilter enable
```

Loop Guard

STP relies on the exchange of BPDUs to maintain a loop free environment.

If a software or hardware failure causes a switch to stop receiving BPDUs, a switch will eventually discard that BPDU information, after the max age timer has expired.

This may result in the switch incorrectly transitioning a blocking port to a forwarding state, thus creating a loop.

UDLD addresses only one of the possible causes of this scenario – a unidirectional link. Other issues may prevent BPDUs from being received or processed, such as the CPU on a switch being at max utilization.

Loop Guard provides a more comprehensive solution – if a blocking port stops receiving BPDUs on a VLAN, it is moved into a loop-inconsistent state for that VLAN.

A port in a loop-inconsistent state cannot forward traffic for the affected VLANs, and is essentially in a pseudo-errdisable state.

However, Loop Guard can automatically recover. As soon as BPDUs are received again, the port will transition normally through STP states.
Loop Guard can be enabled globally:

```
Switch(config)# spanning-tree loopguard default
```

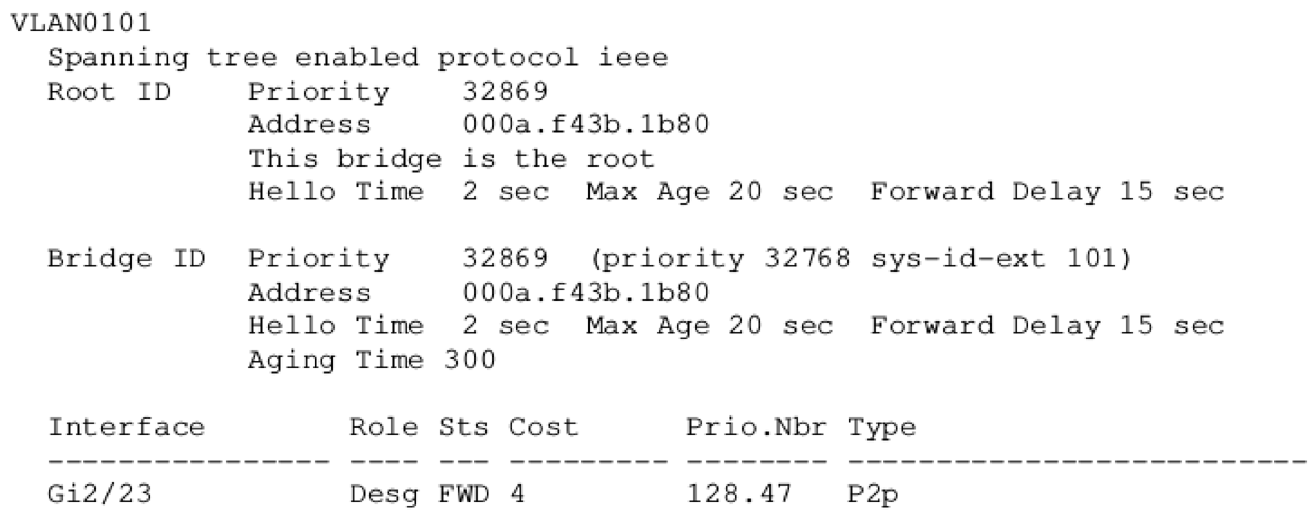
Loop Guard can also be enabled on a per-interface basis:

```
Switch(config)# interface gi2/23
Switch(config-if)# spanning-tree guard loop
```

Loop Guard should only be enabled on trunk ports, or ports that connect to other switches. Loop Guard should never be enabled on a port connecting to a host, as an access port should never receive a BPDU.

Troubleshooting STP

Switch# **show spanning-tree**

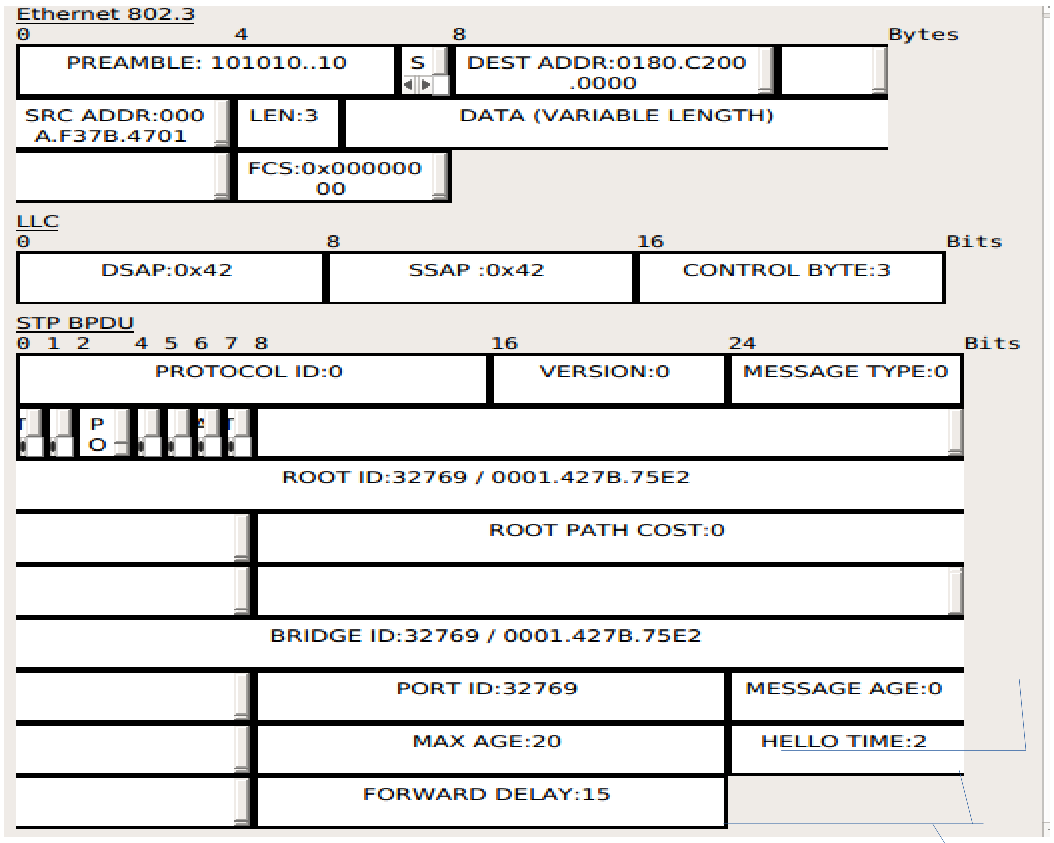


```
VLAN0101
  Spanning tree enabled protocol ieee
  Root ID    Priority    32869
             Address     000a.f43b.1b80
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32869  (priority 32768 sys-id-ext 101)
             Address     000a.f43b.1b80
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

  Interface        Role Sts Cost      Prio.Nbr Type
  ---------------- ---- --- --------- -------- --------------------------
  Gi2/23           Desg FWD 4         128.47   P2p
```

To view STP information specific to an interface:

Switch# **show spanning-tree interface gi2/24**

RSTP BPDUS



Verision Length
1 byte

Protocol Identifier : A two-octet field that identifies the Spanning Tree Protocol (STP).The field is set to a value of 0000 0000 0000 0000.

Protocol Version Identifier : A one-octet field that identifies the current version of the Protocol being used, which was specified in the previous field. For the STP, this field is set to a value of 0000 0000

MESSAGE TYPE : A one-octet field that identifies the type of BPDU being sent. This is set to a value of 0000 0000 for a configuration BPDU.

Flags : A one-octet field that contains flags used in response to a topology change notification BPDU.Bit 1 is the topology change flag.It is used by the root to tell all switches to speed up their aging timers for their filtering databases.

The Flags field includes one of the following:

- 0... .... Topology change (TC) bit, which signals a topology change

Topology change acknowledgment (TCA) bit, which is set to acknowledge receipt of a configuration message with the TC bit set

.0.. .... Agreement : NO

..1. .... Forwarding : YES

...1 .... Learning : YES

.... 11.. Port Role : Designated (3)

.... ..0. Proposol : NO

....1 Topology Change : YES

Root ID :

The Root ID field indicates the root bridge by listing its 2-byte priority followed by its 6-byte ID.

Root Path Cost :

The Root Path Cost field indicates the cost of the path from the bridge sending the configuration message to the root bridge.

Bridge ID :

The Bridge ID field indicates the priority and ID of the bridge sending the message.

Port ID :

The Port ID field indicates the port number (IEEE or Cisco Spanning-Tree Protocol BPDU) or the ring and bridge number (IBM Spanning-Tree Protocol BPDU) from which the configuration message was sent. This field allows loops created by multiple attached bridges to be detected and corrected

Message Age :

The Message Age field indicates the amount of time that has elapsed since the root sent the configuration message on which the current configuration message is based.

Maximum Age :

The Maximum Age field indicates when the current configuration message should be deleted.

Hello Time :

The Hello Time field indicates the time between root bridge configuration messages.

Forward Delay :

The Forward Delay field indicates the length of time that bridges should wait before transitioning to a new state after a topology change. If a bridge transitions too soon, it is possible that not all network links will be ready to change their state and loops can result.

Mutiple Spanning Tree Protocol

Earlier in this guide, three versions of 802.1D STP were described:

- CST utilizes a single STP instance for all VLANs.
- PVST and PVST+ employ a separate STP instance for each VLAN.

PVST and **PVST+** are more efficient, and allow **STP** to load balance **VLANs** across links. This comes at a cost – maintaining a separate **STP** instance for each **VLAN** adds overhead to the CPU and memory on a switch.

Multiple Spanning Tree (MST), defined in IEEE 802.1s, allows a group of VLANs to be mapped to an STP instance.

Each MST instance (MSTI) builds its own RSTP topology database, including electing its own Root Bridge. A VLAN can only be assigned to one instance.

MST further separates the STP topology into regions. All switches in a region must be configured with identical MST parameters:

- **32-byte configuration name**
- **16-bit revision number**
- **VLAN-to-instance mapping database**

If two switches are configured with different MST parameters, they belong to different MST regions.

For most Cisco platforms, a region can contain a maximum of **16 MST instances**, numbered **0 through 15**. By default, **all VLANs** belong to **instance 0**.

The **Internal Spanning Tree (IST)** is responsible for maintaining the topology for the entire region and all of the **MSTIs**. Only the IST can send and receive **BPDUs**, and encapsulates the **MSTI** information within a **BPDU** as an **MST** record (M-record).

The IST is always mapped to **instance 0.**

MST is compatible with all other implementations of **STP.** An **MST** region is obfuscated from non-**MST** switches, which will see the entire **MST** region as a **single 802.1D** or **RSTP switch.**

To enable **MST globally** on a switch:

```
Switch(config)# spanning-tree mode mst
```

Changes to **MST** parameters must be made from **MST** configuration mode:

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)#
```

To assign the **MST configuration name** and **revision number:**

```
Switch(config-mst)# name MYMSTNAME
 Switch(config-mst)# revision 2
```

To map VLANs to a specific MST instances:

```
Switch(config-mst)# instance 2 vlan 1-100
Switch(config-mst)# instance 3 vlan 101-200
```

Remember: A maximum of 16 MST instances are supported, numbered 0 to 15. The MST configuration name, revision number, and VLAN-to-instance mapping must be identical on all switches in the same region.

To view the changes to the configuration:

```
                Switch(config-mst)# show pending
Pending MST configuration
Name [MYMSTNAME]
Revision 2
                Instance                  Vlans mapped
                 --------               ------------------------
                   0                       201-4094
                   2                        1-100
                   3                       101-200
```

All other MST parameters are configured identically to 802.1D STP, with two exceptions:

- **The mst parameter must be used on all commands**
- **All commands reference the MST instance instead of a VLAN.**

Thus, to configure a switch as the Root Bridge for **MST instance 2:**

```
Switch(config)# spanning-tree mst 2 root primary
```

Multiple Spanning Tree Instance Ports

- **Root:** Provides the minimum cost path from the Bridge to the MSTI Regional Root.
- **Designated:** Provides the least cost path from the attached LANs through the Bridge to the

Regional Root.

- **Master:** Provides connectivity from the Region to a CIST Root that lies outside the Region. The Bridge Port that is the CIST Root port for the CIST Regional Root is the Master port for all MSTI.
- **Alternate or Backup:** Provides connectivity if other Bridges, Bridges ports or LANs fail or are erased

```
Frame 914 (167 bytes on wire, 167 bytes captured)
IEEE 802.3 Ethernet
   Destination: 01:80:c2:00:00:00 (Spanning-tree-(for-bridges)_00)
   Source: 00:01:f4:31:03:18 (Enterasys_31:03:18)
   Length: 153
Logical-Link Control
   DSAP: Spanning Tree BPDU (0x42)
   IG Bit: Individual
   SSAP: Spanning Tree BPDU (0x42)
   CR Bit: Command
   Control field: U, func = UI (0x03)
Spanning Tree Protocol
   Protocol Identifier: Spanning Tree Protocol (0x0000)
   Protocol Version Identifier: Multiple Spanning Tree (3)
   BPDU Type: Rapid/Multiple Spanning Tree (0x02)
   BPDU flags: 0x7c (Agreement, Forwarding, Learning, Port Role: Designated)
   Root Identifier: 4096 / 00:01:f4:7e:a0:63
   Port identifier: 0x8371
   Message Age: 0
   Max Age: 20
   Hello Time: 2
   Forward Delay: 15
   Version 1 Length: 0
   MST Extension, Length: 122
      MST Config ID format selector: 0
      MST Config name: a_sample
      MST Config revision: 0
MST Config digest: D0A1A0221648D311DA8FFD31AF847AED
CIST Internal Root Pth Cost: 0
CIST Bridge Identifier: 4096 / 00:01:f4:7e:a0:63
CIST Remaining hops: 20
MSTID 1, Regional Root Identifier 4096 / 00:01:f4:7e:a0:63
   MSTI flags: 0x7c (Agreement, Forwarding, Learning, Port Role: Designated)
   MSTID 1, priority 4096 Rot Identifier 00:01:f4:7e:a0:63
   Internal root path cost: 0
   Bridge Identifier Priority: 1
   Port identifier priority: 8
   Remaining hops: 20
MSTID 2, Regional Root Identifier 4096 / 00:01:f4:7e:a0:C7
   MSTI flags: 0x7c (Agreement, Forwarding, Learning, Port Role: Designated)
   MSTID 2, priority 4096 Rot Identifier 00:01:f4:7e:a0:C7
   Internal root path cost: 0
   Bridge Identifier Priority: 2
   Port identifier priority: 8
   Remaining hops: 19
MSTID 3, Regional Root Identifier 4096 / 00:01:f4:7e:a0:C7
   MSTI flags: 0x7c (Agreement, Forwarding, Learning, Port Role: Designated)
   MSTID 3, priority 4096 Rot Identifier 00:01:f4:7e:a0:C7
   Internal root path cost: 0
   Bridge Identifier Priority: 2
   Port identifier priority: 8
   Remaining hops: 19
```

Dynamic Trunking Protocol

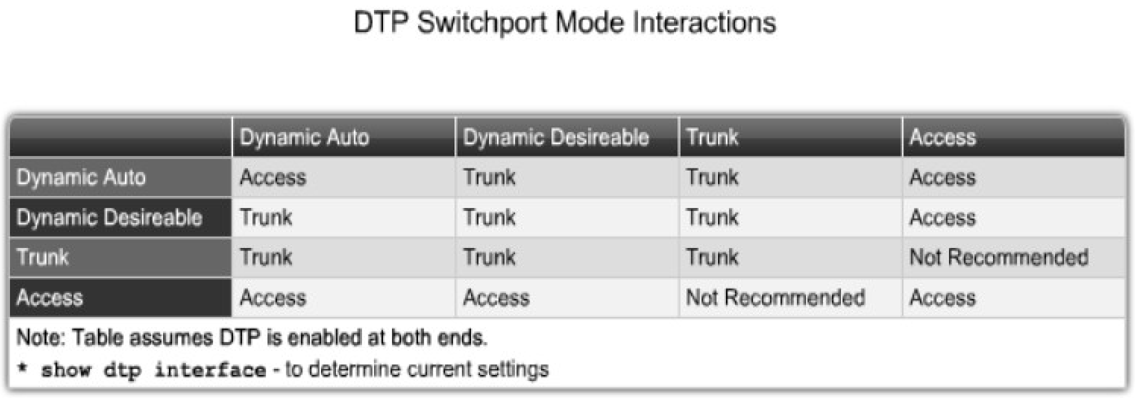
Dynamic Trunking Protocol (DTP) is a Cisco proprietary protocol. Switches from other vendors do not support DTP. DTP is automatically enabled on a switch port when certain trunking modes are configured on the switch port.

DTP manages trunk negotiation only if the port on the other switch is configured in a trunk mode that supports DTP. DTP supports both ISL and 802.1Q trunk**s.**

Port Modes

A switch port on a switch supports a number of trunking modes. The trunking mode defines
how the port negotiates using DTP to set up a trunk link with its peer port.
Port modes are:
1) Access: forces the link into access, even if the neighbor doesn't agree. Notice* this mode is not a trunk mode it’s a port mode where we connect a switch to end devices (PC, servers...)
2) Trunk: forces the link into permanent trunking, even if the neighbor doesn't agree.
3) Dynamic Auto: causes the port to passively be willing to convert to trunking. The port will not trunk unless the neighbor is set to on or desirable. This is the default mode. Note that auto-auto (both ends default) links will not become trunks.
4) Dynamic desirable: causes the port to actively attempt to become a trunk, subject to
neighbor agreement
5) Non-negotiate: Disables the sending of DTP frames on the port. For use when the DTP frames confuse the neighboring (non-Cisco) 802.1Q switch. It does not set the port to trunking mode automatically. Trunking is determined based off whether the on or off keywords have been entered on the device. This command generates an error if dynamic modes are configured.

The switch port periodically sends DTP frames


DTP Switchport Mode Interactions

	Dynamic Auto	Dynamic Desireable	Trunk	Access
Dynamic Auto	Access	Trunk	Trunk	Access
Dynamic Desireable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	Trunk	Not Recommended
Access	Access	Access	Not Recommended	Access

Note: Table assumes DTP is enabled at both ends.

* **show dtp interface** - to determine current settings

Routing Information Protocol

Introduction:

RIP is a routing protocol based on the Bellman-Ford (or distance vector) algorithm. This algorithm has been used for routing computations in computer networks since the early days of the ARPANET. the network will be organized as a collection of Autonomous Systems (AS), each of which will, in general, be administered by a single entity. Each AS will have its own routing technology, which may differ among AS's. The routing protocol used within an AS is referred to as an Interior Gateway Protocol (IGP). A separate protocol, called an Exterior Gateway Protocol (EGP), is used to transfer routing information among the AS's. RIP was designed to work as an IGP in moderate-size AS's. It is not intended for use in more complex environments.

RIP Versions1 :

1) Open Standard Protocol.
2) Classful Routing Protocol.
3) Updates are broadcast via 255.255.255.255.
4) Metric : Hop Count.
5) Load Balancing of 4 equal Paths.
6) Max Hop Counts : 15 Max router : 16.
7) used for small organizations
8) Exchange entire routing table for every 30 second.
9) Administrative distance is 120

Timers

Update Timer :30 sec

Time between consecutive updates

Invalid Timer : 180 secondary

Time a router waits to hear updates

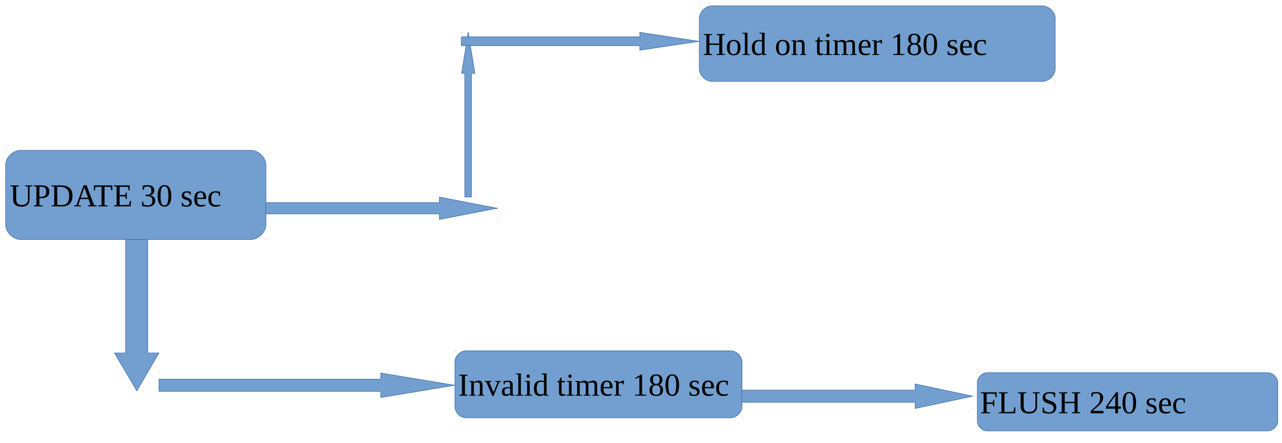
The route is marked unreachable if there is no update during this interval

Flush Timer : 240 sec

Time before the invalid route is plugged from the routing table

Hold On Timer : 180 sec

Stabilizes routing information and helps preventing routing loops during periods when the topology is converging on new information



30 + 150

180 + 60 = 240

Periodic Timer :(25 to 35 sec)

It controls the advertisement of regular update message it counts down to zero (0),when 0 is reached an update message is sent and timer is round only set once again

Expiration : (180 sec)

It governs validity of a route. If no update is received within allocated 180 sec route is considered to be expired and **HOP COUNT** is set to **16.**

Garbage Collection :(120 sec)

When the information about a route becomes invalid router doesn't purge that route from table, instead continuous to advertisement the route with cost = 6.When the count is 0, route is purged from the table.

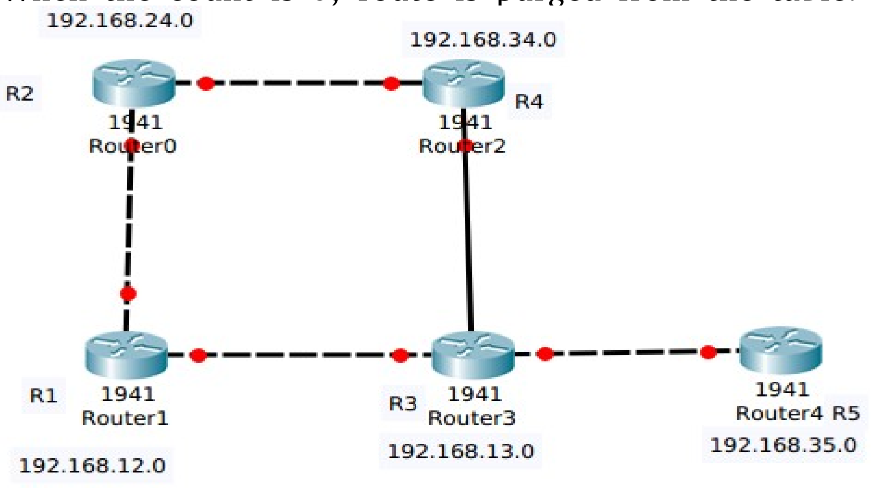


FIG :1

R3 has just sent out an update to **R1** and **R4** and **link** between **R3** and **R5**

goes down R3 still waits the **Periodictimer update** timer expires to notify R1.

Split Horizon

Split Horizon rules dicate that routing information should never be sent out an interface.In figure-1,**R3** is providing update about **192.168.35.0** to **R1** and **R4** upto this point **R1** and **R4** should advertise this information to **R2** and back to **R3**.This process of advertising back to the route to originator is called **"REVERSE ROUTE"**

Common sense dicates that **R1** and **R4** should not advertise back to R3 sinse it already knows about **192.168.35.0** and off course it waste resources. **Split Horizon** prevents this reverse route
For Ex:Let us assume that **Split Horizon** is not in effect and network 192.168.35.0 is unreachable even if the route invalidation timer is in effect,R3 must wait for next periodic update but unexpecdetly,R4 update arrives early. Now R3 has 192.168.35.0 reachable via R4 in two hops and a routing loop has occurred.

Count To Inifinity

The Primary purpose of **Split Horizon** is to avoid loop between neighbours.But Split Horizon alone cannot avoid loop in a network
FIG 1: For sake of simplicity,we are not considering multiple path to the distination.
Now assumming that the network is stable,we have the following routing table entries for **192.168.35.0**

Router	**Network**	**via**	**Hop Count**
R3	**192.168.35.0**	**Directly connected**	**0**
R4	**192.168.35.0**	**R3**	**1**
R2	**192.168.35.0**	**R4 and R1**	**2**
R1	**192.168.35.0**	**R3**	**1**

Let us take a closer look how routing information is propogated
The network 192.168.35.0 has failed R3 sends an update to R1 and R4 that 192.168.35.0 unreachable.

Router	**Network**	**via**	**Hop Count**
R3	**192.168.35.0**	**Unreachable**	**NULL**
R4	**192.168.35.0**	**Unreachable**	**NULL**
R1	**192.168.35.0**	**Unreachable**	**NULL**

Since R2 hasn't yet informed of the update that **192.168.35.0** is now not available (and unfortunately) it sends an update to **R1**.How did the hop count increased? Actually **R2** learned **192.168.35.0** from **R1** and **R4**,therefore information learned from **R4** is a different entity than that of **R1.R1** will advertise **192.168.35.0** via **R4** to **R1** and vice versa and **Spilt Horizon** rule is not void on this condition.In a stable network **R1** and **R4** will not install these advertisements from **R2**,Since they already have best path through **R3** in the loop.

Router	**Network**	**via**	**Hop Count**
R4	**192.168.35.0**	**R2**	**3**
R2	**192.168.35.0**	**R4 and R1**	**2**
R1	**192.168.35.0**	**R3**	**3**

R1 send this update to R3.R3 now can reach 192.168.35.0 via R1 in four hops.The routing increases up to infinity 16.

R3 install this route and advertise to R4.R4 now think that the metric towards 192.168.35.0 from R3 increased ;nonetheless,it is the only route and must be installed.The routing table of R4 looks like

Router	**Network**	**via**	**Hop Count**
R4	**192.168.35.0**	**R3**	**5**

R4 address to **R2,R2** to **R1** to **R3** and goes on.For each update the hop count keeps on increasing this phenomeon is called to **Count to Infinity**.To elevate this problem distance vector provide define **infinity**
For Example:**RIP** defines **infinity** as **hop count** reaches **16** and such routing information is discarded immediately.Although routing loop will be eliminated,but the convergence will be slow as each router must wait for hop count to reach **infinity**.

Split Horizon with Poison Reverse (Route Poisoning)

Uses to **"infinity"** value and advertise the route back to originator on the interface through which it was learned from as **"Unreachable",**The analogy is simple:**"bad information is better than no information".**The count to infinity problem can be avoided since each neighbour as **unreachable.**There is no need to wait for hop count to reach **inifinity.**

RIP packet format

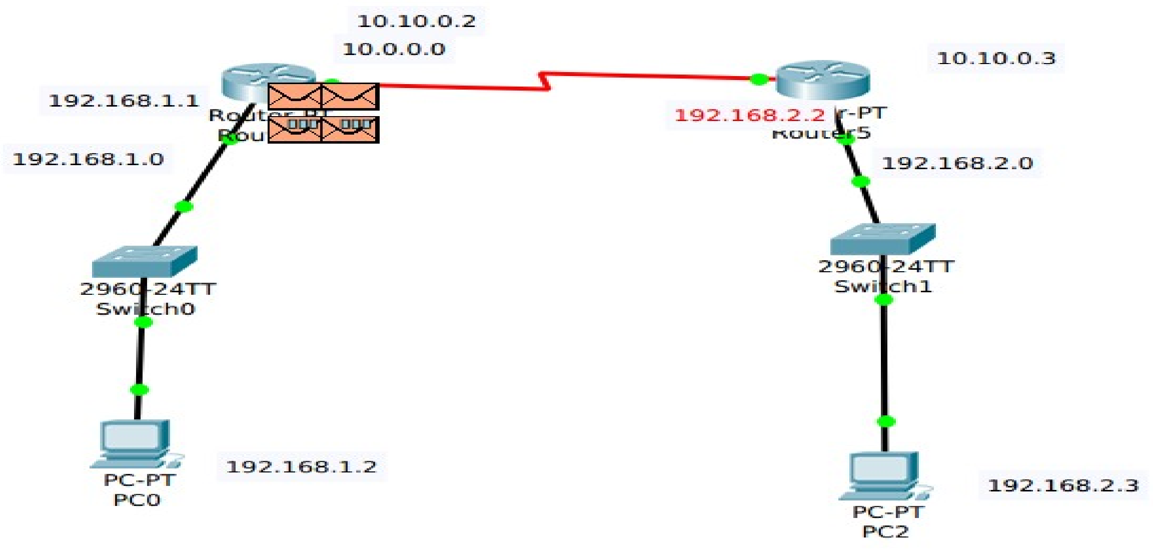
Command	Version	Reserved
Family		All 0'S
Network Address		
All 0's		
All 0's		
Distance		

RIP is a UDP-based protocol. Each router that uses RIP has a routing process that sends and receives datagrams on UDP port number 520, the RIP-1/RIP-2 port. All communications intended for another routers's RIP process are sent to the RIP port. All routing update messages are sent from the RIP port. Unsolicited routing update messages have both the source and destination port equal to the RIP port. Update messages sent in response to a request are sent to the port from which the request came. Specific queries may be sent from ports other

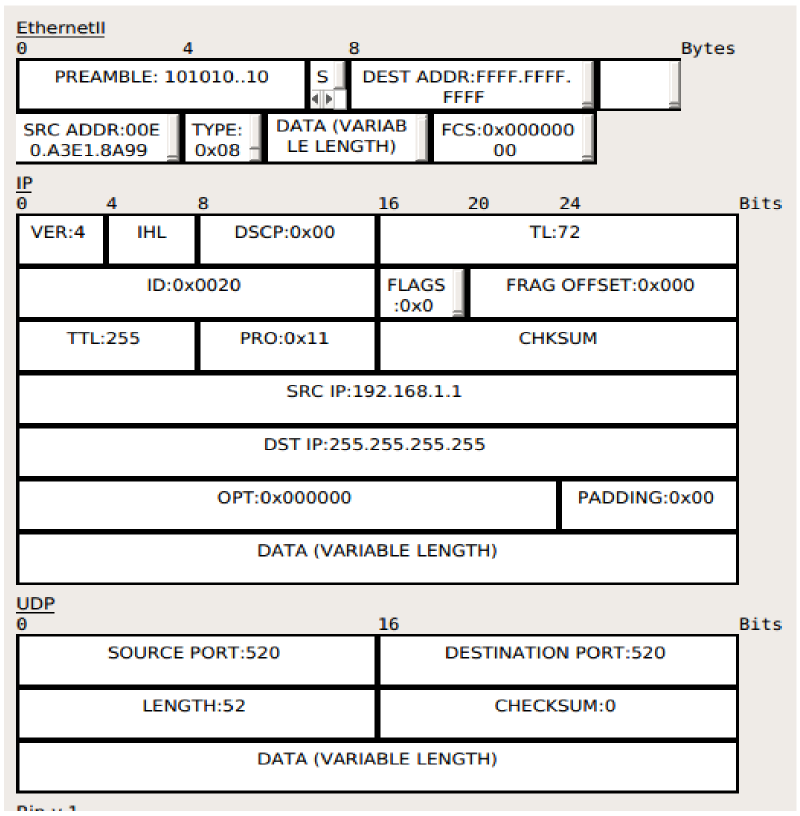
Command Indicates whether the packet is a request or a response. The request asks that a router send all or part of its routing table. The response can be an unsolicited regular routing update or a reply to a request. Responses contain routing table entries. Multiple RIP packets are used to convey information from large routing tables.

- **Version number** - Specifies the RIP version used. This field can signal different potentially incompatible versions.
- **Zero** - it was added solely to provide backward compatibility with prestandard varieties of RIP. Its name comes from its defaulted value: zero.
- **Address-family identifier (AFI)** - Specifies the address family used. RIP is designed to carry routing information for several different protocols. Each entry has an address-family identifier to indicate the type of address being specified. The AFI for IP is 2.
- **Address** - Specifies the IP address for the entry.

Metric - Indicates how many internetwork hops (routers) have been traversed in the trip to the destination. This value is between 1 and 15 for a valid route, or 16 for an unreachable route.



RIP verision 1



EthernetII

0 4 8 Bytes

PREAMBLE: 101010..10 | S | DEST ADDR:FFFF.FFFF.FFFF |

SRC ADDR:00E0.A3E1.8A99 | TYPE: 0x08 | DATA (VARIABLE LENGTH) | FCS:0x00000000

IP

0 4 8 16 20 24 Bits

VER:4 | IHL | DSCP:0x00 | TL:72

ID:0x0020 | FLAGS :0x0 | FRAG OFFSET:0x000

TTL:255 | PRO:0x11 | CHKSUM

SRC IP:192.168.1.1

DST IP:255.255.255.255

OPT:0x000000 | PADDING:0x00

DATA (VARIABLE LENGTH)

UDP

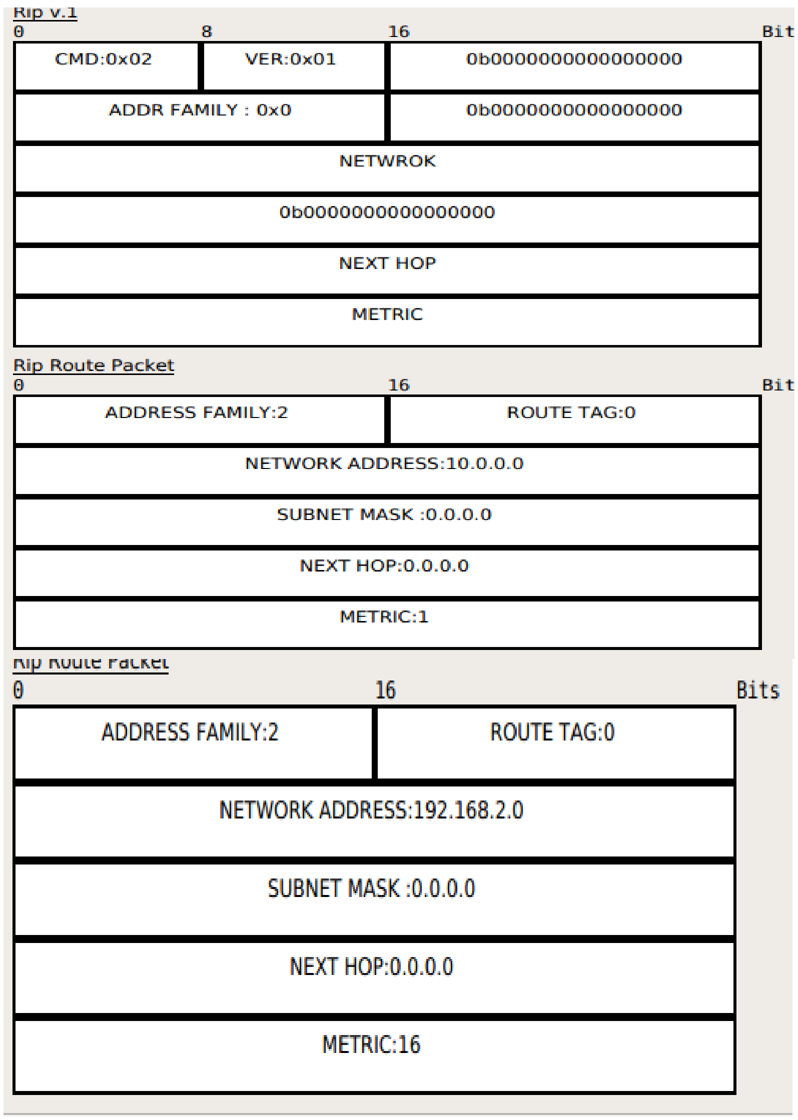
0 16 Bits

SOURCE PORT:520 | DESTINATION PORT:520

LENGTH:52 | CHECKSUM:0

DATA (VARIABLE LENGTH)

Rip v 1



Rip v.1

0 8 16 Bit

CMD:0x02 | VER:0x01 | 0b0000000000000000

ADDR FAMILY : 0x0 | 0b0000000000000000

NETWROK

0b0000000000000000

NEXT HOP

METRIC

Rip Route Packet

0 16 Bit

ADDRESS FAMILY:2 | ROUTE TAG:0

NETWORK ADDRESS:10.0.0.0

SUBNET MASK :0.0.0.0

NEXT HOP:0.0.0.0

METRIC:1

Rip Route Packet

0 16 Bits

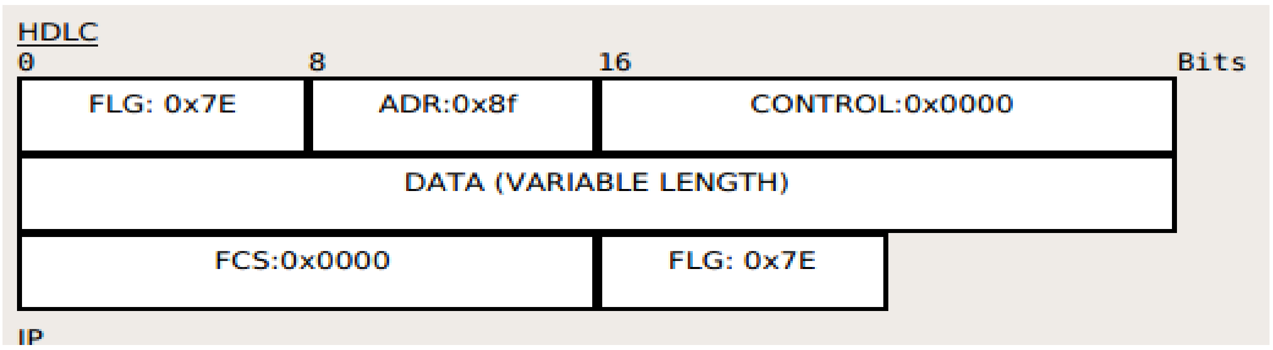
ADDRESS FAMILY:2 | ROUTE TAG:0

NETWORK ADDRESS:192.168.2.0

SUBNET MASK :0.0.0.0

NEXT HOP:0.0.0.0

METRIC:16



HDLC

0 8 16 Bits

FLG: 0x7E | ADR:0x8f | CONTROL:0x0000

DATA (VARIABLE LENGTH)

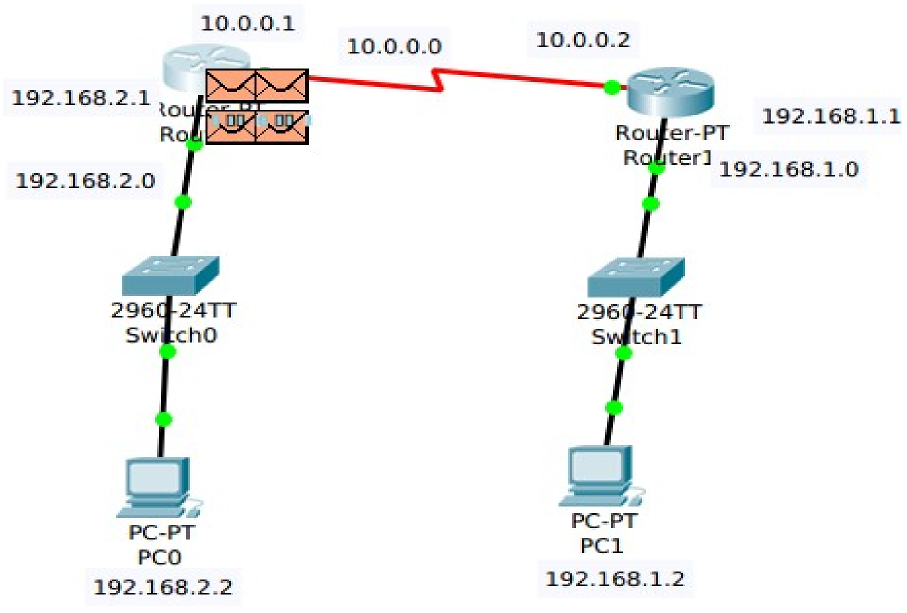
FCS:0x0000 | FLG: 0x7E

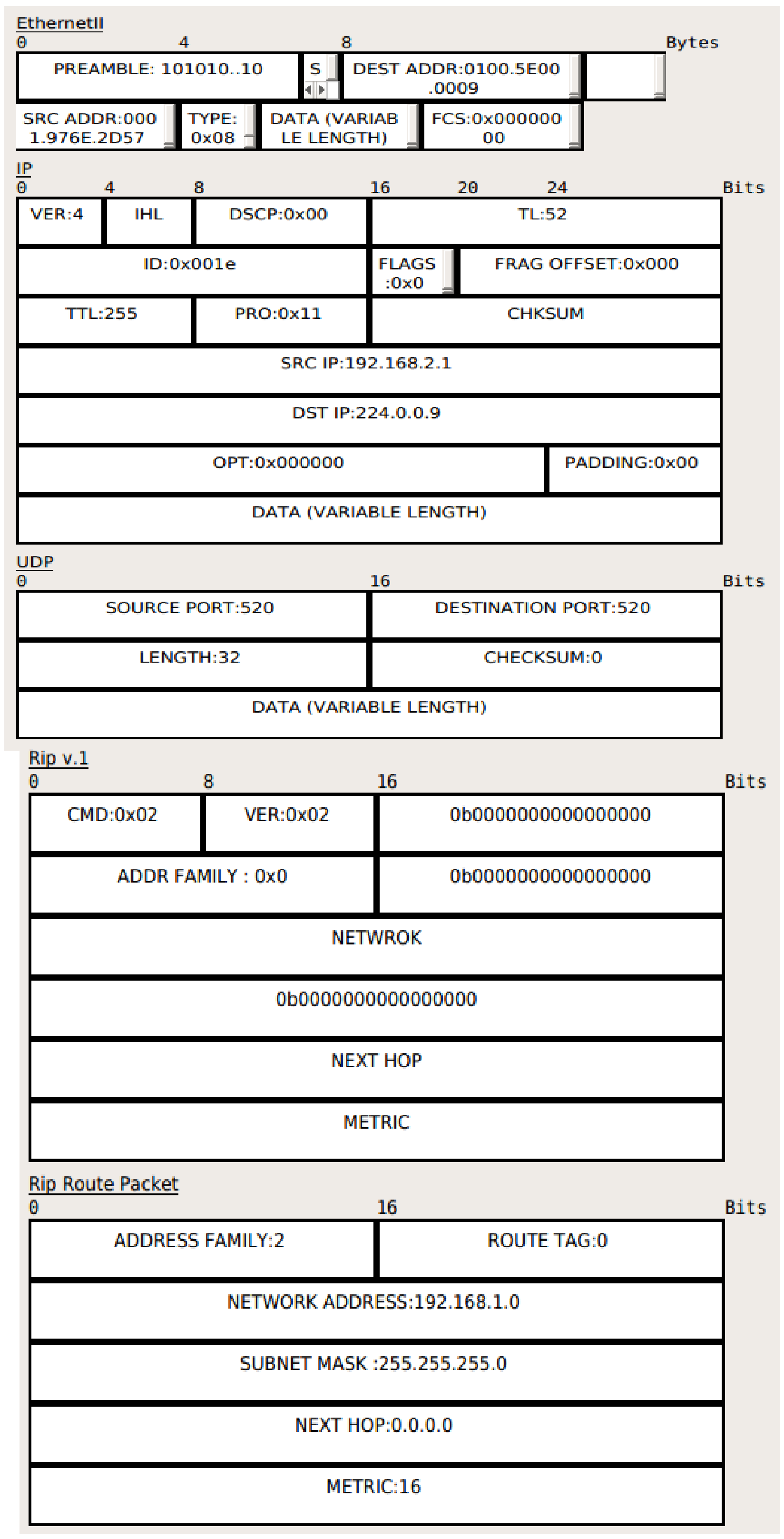
IP

RIP Version 2

1) Open Standard Protocols
2) Classless,Athentication
3) Updates are multicast via 224.0.0.9
4) Load balancing of 4 equal paths
5) Max Hop Counts :15 Max routers :16
6) Used for small organizations

7) Exchange entire routing table for every 30 seconds
8) Administrative distance is 120





EthernetII

0 4 8 Bytes

PREAMBLE: 101010..10	S	DEST ADDR:0100.5E00 .0009	
SRC ADDR:0001.976E.2D57	TYPE: 0x08	DATA (VARIABLE LENGTH)	FCS:0x00000000

IP

0 4 8 16 20 24 Bits

VER:4	IHL	DSCP:0x00	TL:52	
ID:0x001e			FLAGS :0x0	FRAG OFFSET:0x000
TTL:255		PRO:0x11	CHKSUM	
SRC IP:192.168.2.1				
DST IP:224.0.0.9				
OPT:0x000000				PADDING:0x00
DATA (VARIABLE LENGTH)				

UDP

0 16 Bits

SOURCE PORT:520	DESTINATION PORT:520
LENGTH:32	CHECKSUM:0
DATA (VARIABLE LENGTH)	

Rip v.1

0 8 16 Bits

CMD:0x02	VER:0x02	0b0000000000000000
ADDR FAMILY : 0x0		0b0000000000000000
NETWROK		
0b0000000000000000		
NEXT HOP		
METRIC		

Rip Route Packet

0 16 Bits

ADDRESS FAMILY:2	ROUTE TAG:0
NETWORK ADDRESS:192.168.1.0	
SUBNET MASK :255.255.255.0	
NEXT HOP:0.0.0.0	
METRIC:16	

Command - Indicates whether the packet is a request or a response. The request asks that a router send all or a part of its routing table. The response can be an unsolicited regular routing update or a reply to a request. Responses contain routing table entries. Multiple RIP packets are used to convey information from large routing tables.

- Version - Specifies the RIP version used. In a RIP packet implementing any of the RIP 2 fields or using authentication, this value is set to 2.
- Unused - Has a value set to zero.
- Address-family identifier (AFI) - Specifies the address family used. RIPv2's AFI field functions identically to AFI field, with one exception: If the AFI for the first entry in the message is 0xFFFF, the remainder of the entry contains authentication information. Currently, the only authentication type is simple password.
- Route tag - Provides a method for distinguishing between internal routes (learned by RIP) and external routes (learned from other protocols).
- IP address - Specifies the IP address for the entry.
- Subnet mask - Contains the subnet mask for the entry. If this field is zero, no subnet mask has been specified for the entry.
- Next hop - Indicates the IP address of the next hop to which packets for the entry should be forwarded.
- Metric - Indicates how many internetwork hops (routers) have been traversed in the trip to the destination. This value is between 1 and 15 for a valid route, or 16 for an unreachable route.

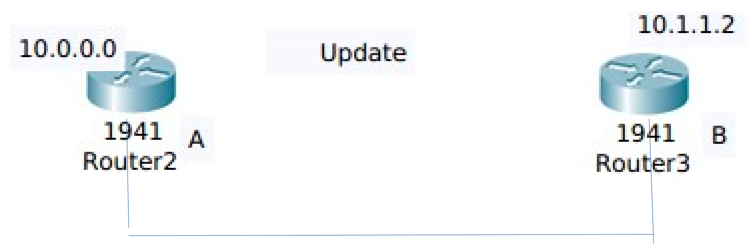
Configuration of RIP

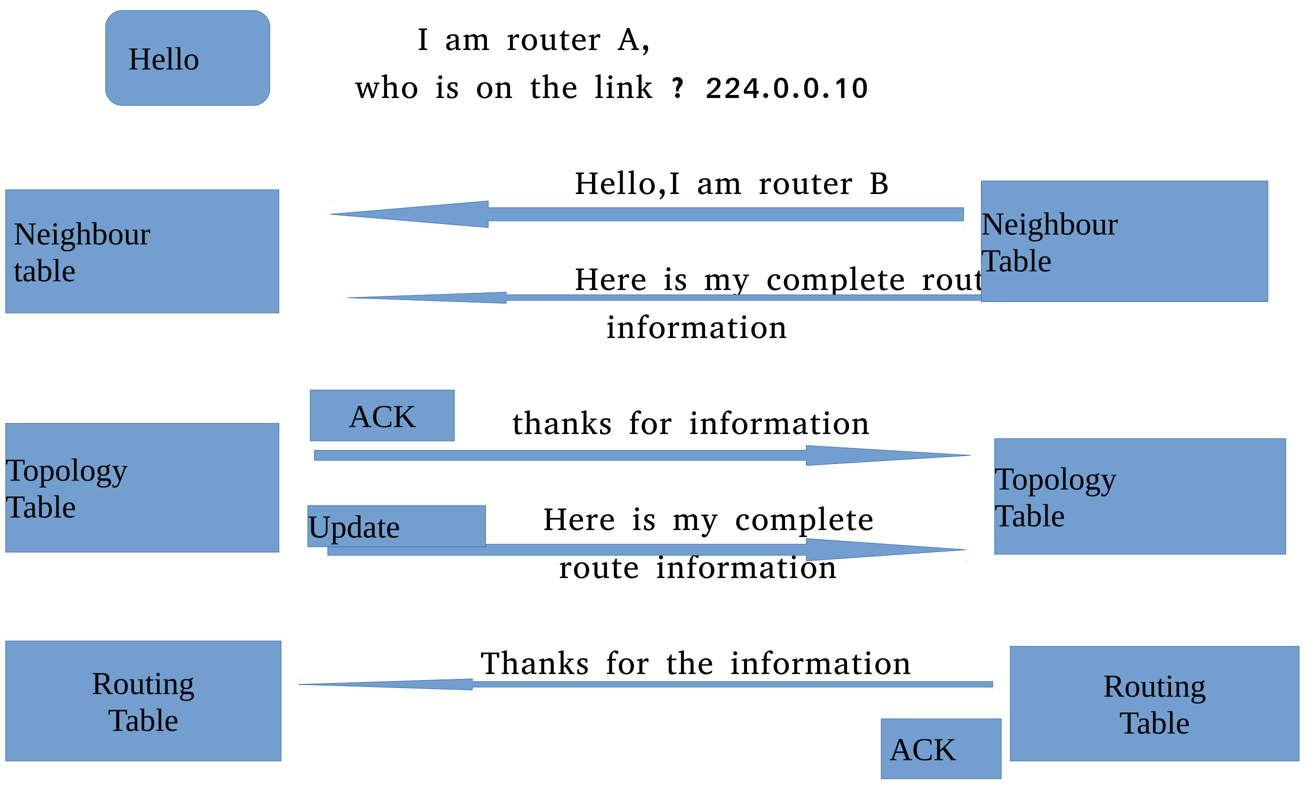
```
r1(config)#router rip
r1(config-router)#version 2
r1(config-router)#network 192.168.1.0
r1(config-router)#network 10.0.0.0

r2(config)#router rip
r2(config-router)#version 2
r2(config-router)#network 192.168.2.0
r2(config-router)#network 10.0.0.0
r2(config-router)#network 11.0.0.0
```

EIGRP (Enhanced Interior Gateway Routing Protocol)

1) Advanced distance vector.
2) Standard Protocol (initially was cisco proprietary).
3) Classless Routing Protocol.
4) Includes all features of GIRP.
5) Max Hop Count is 255 (100 by default).
6) Administrative distance is 90.
7) Flexible network design.

8) Multicast and Unicast instead of broadcast address.
9) 100% loop-free classless routing.
10) Easy configuration for WANs and LANs.
11) The multicast address is 224.0.0.10.





But the best route base on the bandwidth

EIGRP Tables

1) <u>Neighbor Table</u>

i) Contains list of directly connected routers

#show ip eigrp neighbor

2) <u>Topology Table</u> contains collection of best routes

i) List of all the best routes learned from each neighbor

#show ip eigrp topology

3) <u>Routing table</u> contains the best route in routing table which is selected by topology table

i) The best route to the destination

#show ip route

1) Updates are through Multicast (224.0.0.10).

2) Hello Packets are sent every 5 seconds.
3) Convergence rate is fast.
4) Supports IP,IPX and Apple Talk Protocols.
5) It uses DUAL (diffusion update algorithm).
6) Supports equal cost an unequal cost load balancing.

EIGRP Metric Calculation

EIGRP uses **Bandwidth + delay + load + MTU + reliability**

k1 k2 k3 k4 k5

2) By default uses bandwidth and delay in the metric calculation.
3) Formula with default **k** values **(k1 = 1,k2 = 0,k3 = 1,k4 = 0,k5 = 0).**
4) **Metric =[k1 * BW + ((k2 * BW)/(256-load)) + k3 * delay).**

EIGRP can calculate best route based on **5 factors** like **BW + Delay + load + MTU + reliability**

<u>Default Values</u>

Serial Link Bandwidth = 1554 kbps
Ethernet Link Bandwidth = 10 mbps
Fast ethernet Link Bandwidth = 100 mbps
giga ethernet link Bandwidth = 1000 mbps

we can change bandwidth when there is serial link between the routers

```
router#cont f
router(conf)#int s0/0
router(conf-if)#bandwidth 1000
```

<u>Delay</u>

serial = 20,000 microsecond
ethernet = 200 microsecond
fast ethernet = 100 microsecond
giga = 10 microsecond

Delay is the sum of all the delays of the links along the paths
Delay = [Delay in tens of microseconds] * 26
Bandwidth = [10,000,000/**(bandwidth in kbps)]*256**
10^8/64 * 256

Configuring EIGRP

```
router(config)#router eigrp <AS NO>
router(config-if)#network <network ID>

router1(config)#router eigrp 100
router1(config-if)#network 192.168.1.0
router1(config-if)#network 10.0.0.0
```

```
router2(config)#router eigrp 100
router2(config-if)#network 192.168.2.0
router2(config-if)#network 11.0.0.0
router2(config-if)#network 10.0.0.0

router3(config)#router eigrp 100
router3(config-if)#network 192.168.3.0
router3(config-if)#network 11.0.0.0
```

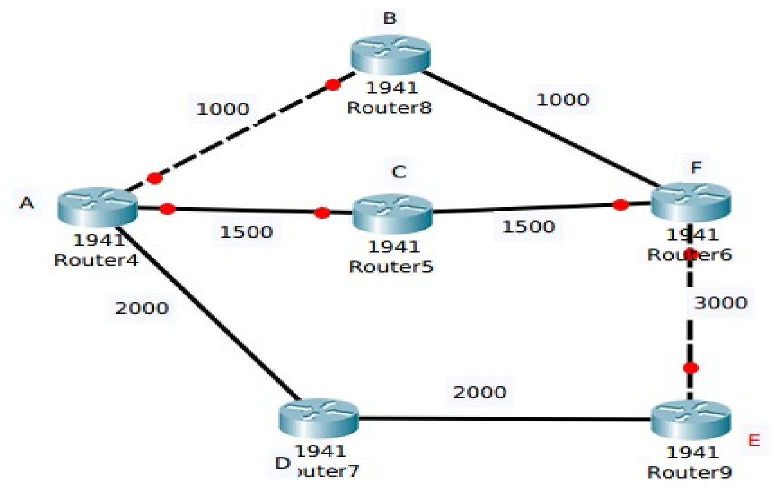
Dual Terminology

Feasible Distance

Total cost from local router to destination cost from local router = AD of next-hop router + cost between the local router and the next hop router.

Advertise Distance

Cost from the next-hop router to the destination



S.NO	A TO F	Feasible Distance	Advertise Distance
1	ABF	2000	1000
2	ACF	3000	1500
3	ADEF	7000	5000

Successor   Second best feasible successor

EIGRP also pre-calculates the second best route if satisfy the feasibility condition.

Successor : Best route to the destination

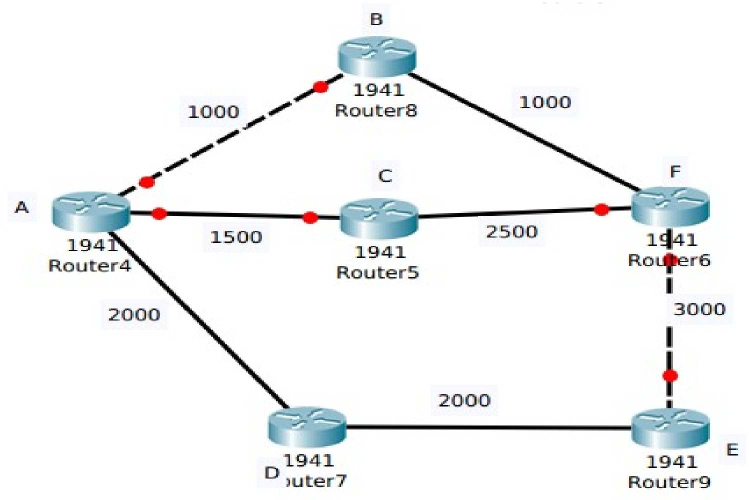
Feasible successsor : backup path

<u>Feasibility Conditions</u>

FD of current Successor route > AD of feasible successor

<u>EIGRP without feasible successor</u>

FD of current successor route > AD of feasible successor



S.NO	A TO F	Feasible Distance	Advertise Distance
1	ABF	2000	1000
2	ACF	4000	2500
3	ADEF	7000	5000

Successor **NO Feasible Successor**

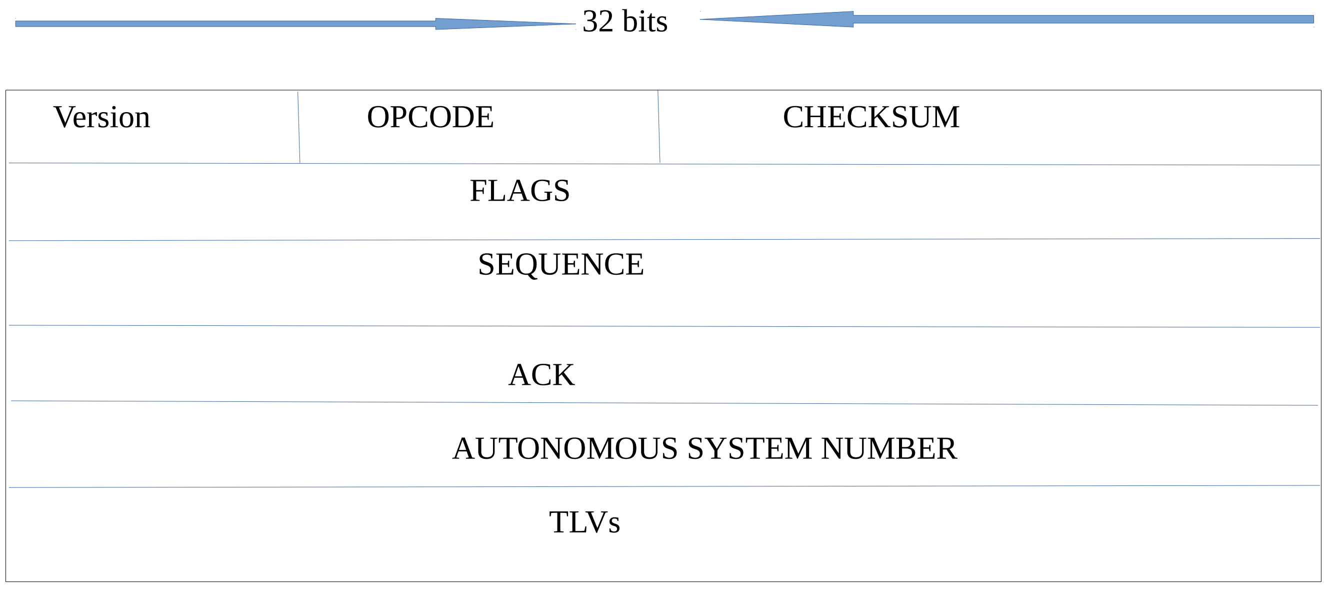
If will not calculate the second best route successor the router C is advertising to me AD = 2500 is the cost which is higher than my best route cost

show ip eirp Topology [110/100]

feasible distance advertise distance

EIGRP Packet Header

The IP Header of the EIGRP packet specifies IP protocol number **58** within it and the maximum length of the packet will be the IP MTU of the interface on which it is transmitted most of the time 1500 Octets.Following the IP header is the various **Type/Length/value** (**TLV) triplets.** These TLVs will not only carry the route entries but also provide fields the management of the **DUAL process,multicast sequencing, and IOS Software versions from the router.**



Version Field : The Version Field is 4-bit field used to indicate the protocol version of the originating EIGRP router process. The version has not changed since its release.
Opcode : The Opcode is a 4-bit field that specific the EIGRP message type,1-Update,3-Query,4-Reply,5-Hello,6=IPXSA,10=SIA Query,11 = SIA Reply.
Checksum : Checksum is a 24-bit field that is used to run a sanity check on the EIGRP Packet itself.This field is based on the entire EIGRP Packet and not including the IP header.
Flags : The Flags is a 32-bit field that is used to indicate either an INIT (when set 0x00000001). For a new EIGRP neighbor or for the conditional receive (CR) (second bit (0x00000002) used in the proprietary reliable multicasting algorithm for EIGRP reliable transport protocol (RTP).
Sequence : Sequence is a 12-bit field that contains sequence number used by RTP.This is used to ensure orderly delivery of reliable packets.
Acknowledgement : Acknowledgement is a 32-bit field that contains the sequence number last heard from the neighbor to which the packet rather than as a hello. Note that an **ACK field** will only be nonzero if the packet itself is a **unicast** because an acknowledgment are **never multicast.**
Autonomous System Number : Autonomous System Number is a 32-bit field that identifies the no of the EIGRP domain**.**
Type/Length/value : Type/Length/Value is a 32-bit triplet field that is used to carry route entries as well as provide EIGRP DUAL information.EIGRP supports several different types of TLVs as follows.

VALUE	TYPE
0X0001	EIGRP Parameters (General TLV Types)
0x0003	Sequence (General TLV Types)
0x0004	Software Version (General TLV Types)
0x0005	Next multicast sequence (General TLV Types)
0x0102	IP Internet Routers (IP-Specific TLV Types)
0x0103	IP External Routers(IP-Specific TLV Types)

TLV Fields : TLVs (Type/Length/Value) are not only found within EIGRP but can be found in other protocols like IS-IS for one. The type and length fields are fixed in size (typically 1-4 bytes),and the value field is of variable size.
Type : A binary code often simply alpha numeric,which indicates the kind of field that this part of the message represents
Length : The size of the value field (typically in bytes)
Value : Variable sized series of bytes which contains data for this part of the message

TLVs carry **EIGRP** management information .The parameters of **TLV** that are used to convey metric weights and the hold time.The sequence software **TLVs** are used by the cisco proprietary reliable multicast algorithm.As it was mentioned earlier **TLV** not only works with **IP** but also **IPX** and apple Talk.Each Internal and External routes **TLV** contains one route entry.Every **Update,Query and Reply** packet contains at least one Routes **TLV**.The Internal and External route TLVs include metric information for the route.

<u>IP Internal Routes TLV</u>

An internal route is a path to a destination within a EIGRP autonomous system

Type = 0x0102		Length
	Next Hop	
	Delay	
	Bandwidth	
MTU		Hop Count
Reliability	Load	Reserved
prefix lenght	Destination	

Next Hop : Next hop is the net-hop neighbouring router IP address.
Delay : Delay is the sum of the configured delays expressed in units of 10 microseconds. A delay of 0xffff ffff or 256 indicates an unreachable route.
Bandwidth : Bandwidth is 256 * BW (min) or 2,560,000,000 divided by the lowest configured bandwidth of any interface along the route.
MTU : MTU is the smallest maximum Transimission unit of any link along the route to the distination. This value is not used for the metric calculation.
Hop Count : Hop count is a number between 0x01 and 0xFF indicating the no of hops to the destination. A router will advertise a directly connected network with a hop count of 0.
Reliability : Reliability is a number between 0x01 and 0xFF that reflects along the routes calculated on a five-minute exponentially weighted average 0xFF indicates a 100 percent reliable link.
Load : Load is also a number between 0x01 and 0xFF reflecting the total outgoing load of the

interfaces along the route, calculated on a five-minute exponentially weighted average 0x01 indicates a minimally loaded link.

Reserved : Reserved is an unused field and is always 0x000.

Prefix Length: Specifies the no of network bits of the address mask.

Destination : Destination is the address of the route.

IP External Routes :

An External route is a path that leads to a destination outside of the EIGRP autonomous system and that has been redistributed into the EIGRP domain.

Type = 0x0103		Length
Next Hop		
Originating routers		
Originating as Number		
Arbitrary Tag		
External Protocol Metric		
Reserved	External Protocol ID	FLAGS
Delays		
Bandwidth		
MTU		Hop count
Reliability	Load	Reserved
prefix length	Destination	

1) **Next Hop :** Next hop is the next IP address on a multicast access network the router advertising the route might not be the best next hop router to the destination. The next hop field allows the “bilingual” router to tell its EIGRP neighbours,”use address A.B.C.D as the next hop instead of using my interface address”.

2) **Originating Router :** Originating Router is the IP address or router ID of the router that redistributed the external route into the EIGRP autonomous system.

3) **Originating Autonomous System Number :** Originating Autonomous System Number is the autonomous system number of the router originating the route.

4) **Arbitrary Tag :** May be used to carry a tag by route maps.

5) **External Protocol Metric :** Reserved Protocol Metric is the metric of the external protocol.

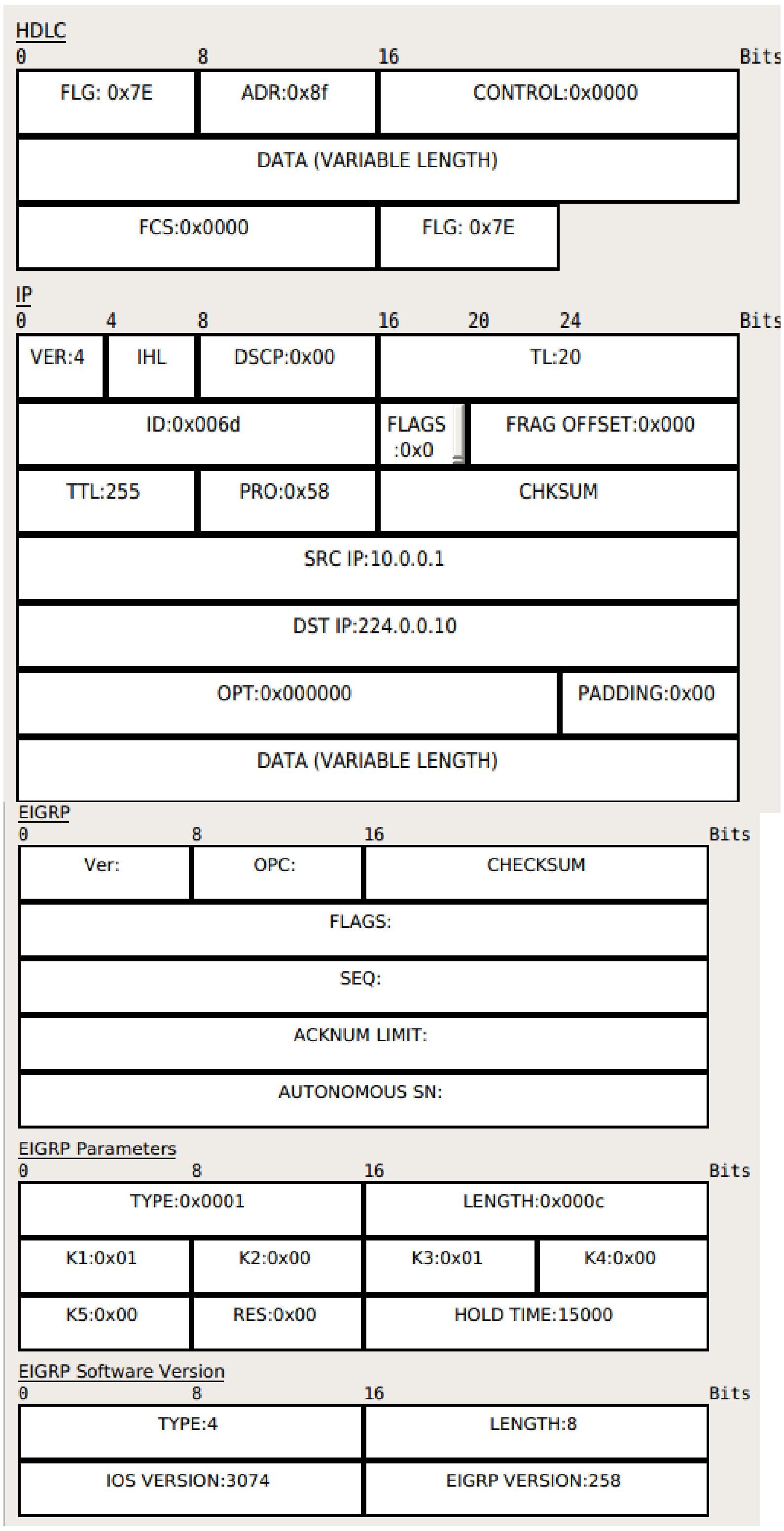
6) **Reserved :** Reserved is an unused field and is always 0x0000.

7) **External Protocol ID :**External Protocol ID specifies the protocol from which the external route was learned 0x01 = IGRP ,0x02 = EIGRP ,0x03 = static route ,0x04 = RIP ,0X05 =Hello, 0x06 = OSPF, 0x07 = IS-IS, 0x08 = EGP,0x09 = BGP,0x0A = IDRP,0x0B = connected link.

8) Flags : Flags currently constitute just 2 flags.If the rightmost bit of the eight bit field is set (0x01),the route is an external route.If the second bit is set (0x02),the route is an candidate default route.

EIGRP:

EIGRP sends hello packets once it has been enabled on a router for a particular network.These message are used to identify neighbors.



HDLC
0 8 16 Bits
FLG: 0x7E | ADR:0x8f | CONTROL:0x0000
DATA (VARIABLE LENGTH)
FCS:0x0000 | FLG: 0x7E

IP
0 4 8 16 20 24 Bits
VER:4 | IHL | DSCP:0x00 | TL:20
ID:0x006d | FLAGS :0x0 | FRAG OFFSET:0x000
TTL:255 | PRO:0x58 | CHKSUM
SRC IP:10.0.0.1
DST IP:224.0.0.10
OPT:0x000000 | PADDING:0x00
DATA (VARIABLE LENGTH)

EIGRP
0 8 16 Bits
Ver: | OPC: | CHECKSUM
FLAGS:
SEQ:
ACKNUM LIMIT:
AUTONOMOUS SN:

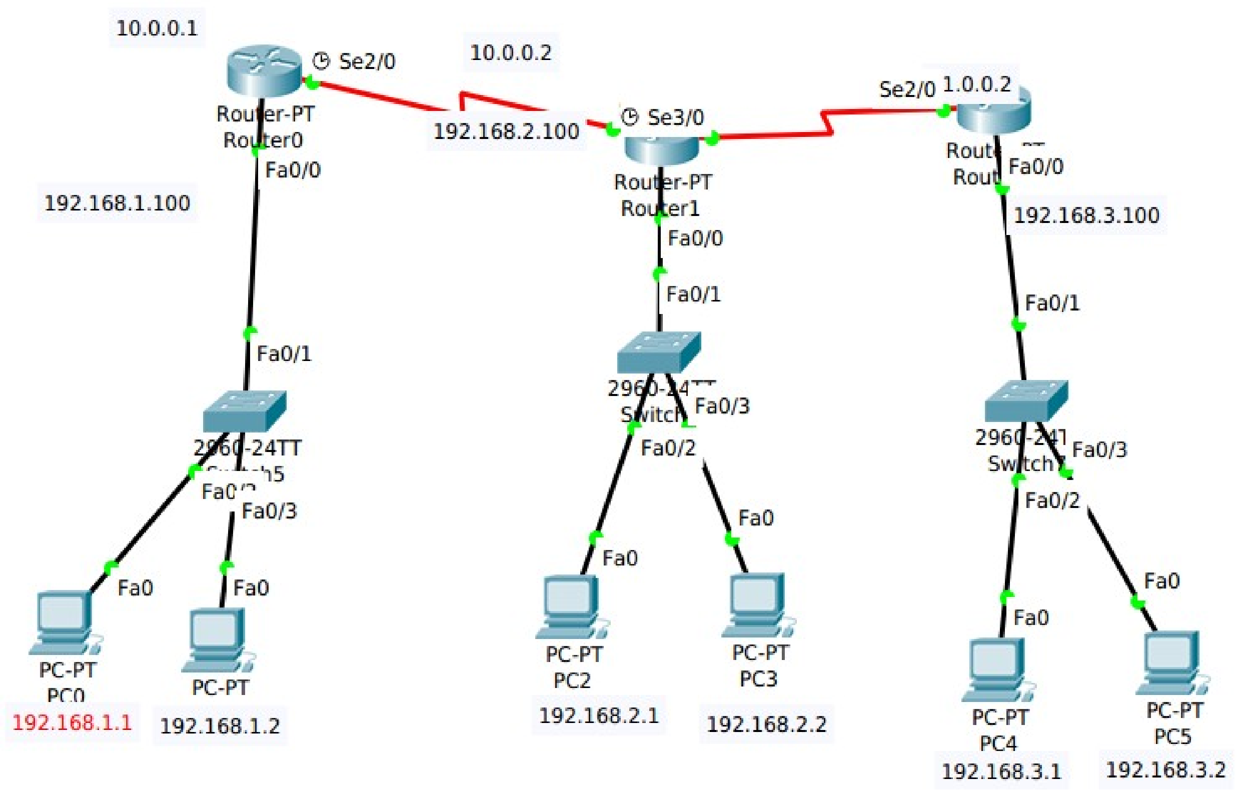
EIGRP Parameters
0 8 16 Bits
TYPE:0x0001 | LENGTH:0x000c
K1:0x01 | K2:0x00 | K3:0x01 | K4:0x00
K5:0x00 | RES:0x00 | HOLD TIME:15000

EIGRP Software Version
0 8 16 Bits
TYPE:4 | LENGTH:8
IOS VERSION:3074 | EIGRP VERSION:258

Version Field : The Version Field is 4-bit field used to indicate the protocol version of the originating EIGRP router process. The version has not changed since its release.
Opcode : The Opcode is a 4-bit field that specific the EIGRP message type,1-Update,3-Query,4-Reply,5-Hello,6=IPXSA,10=SIA Query,11 = SIA Reply.
Checksum : Checksum is a 24-bit field that is used to run a sanity check on the EIGRP Packet itself.This field is based on the entire EIGRP Packet and not including the IP header.
Flags : The Flags is a 32-bit field that is used to indicate either an INIT (when set 0x00000001).
For a new EIGRP neighbor or for the conditional receive (CR) (second bit (0x00000002) used in the proprietary reliable multicasting algorithm for EIGRP reliable transport protocol (RTP).
Sequence : Sequence is a 12-bit field that contains sequence number used by RTP.This is used to ensure orderly delivery of reliable packets.
Acknowledgement : Acknowledgement is a 32-bit field that contains the sequence number last heard from the neighbor to which the packet rather than as a hello. Note that an **ACK field** will only be nonzero if the packet itself is a **unicast** because an acknowledgment are **never multicast.**
Autonomous System Number : Autonomous System Number is a 32-bit field that identifies the no of the EIGRP domain**.**
Type/Length/value : Type/Length/Value is a 32-bit triplet field that is used to carry route entries as well as provide EIGRP DUAL information.

Type 0x0001 = Parameter List
Type 0x0004 = Software List

EIGRP Internals



EIGRP Configuration:

Router0

```
router#conf t
router(config)#hostname r0
r0(config)#int se2/0
r0(config-if)#ip address 10.0.0.1 255.0.0.0
r0(config-if)#exit
r0(config)#int fa0/0
r0(config-if)#ip address 192.168.1.100 255.255.255.0
r0(config-if)#exit

r0(config)#router eigrp 10
r0(config-router)#network 10.0.0.0
r0(config-router)#network 192.168.1.0
```

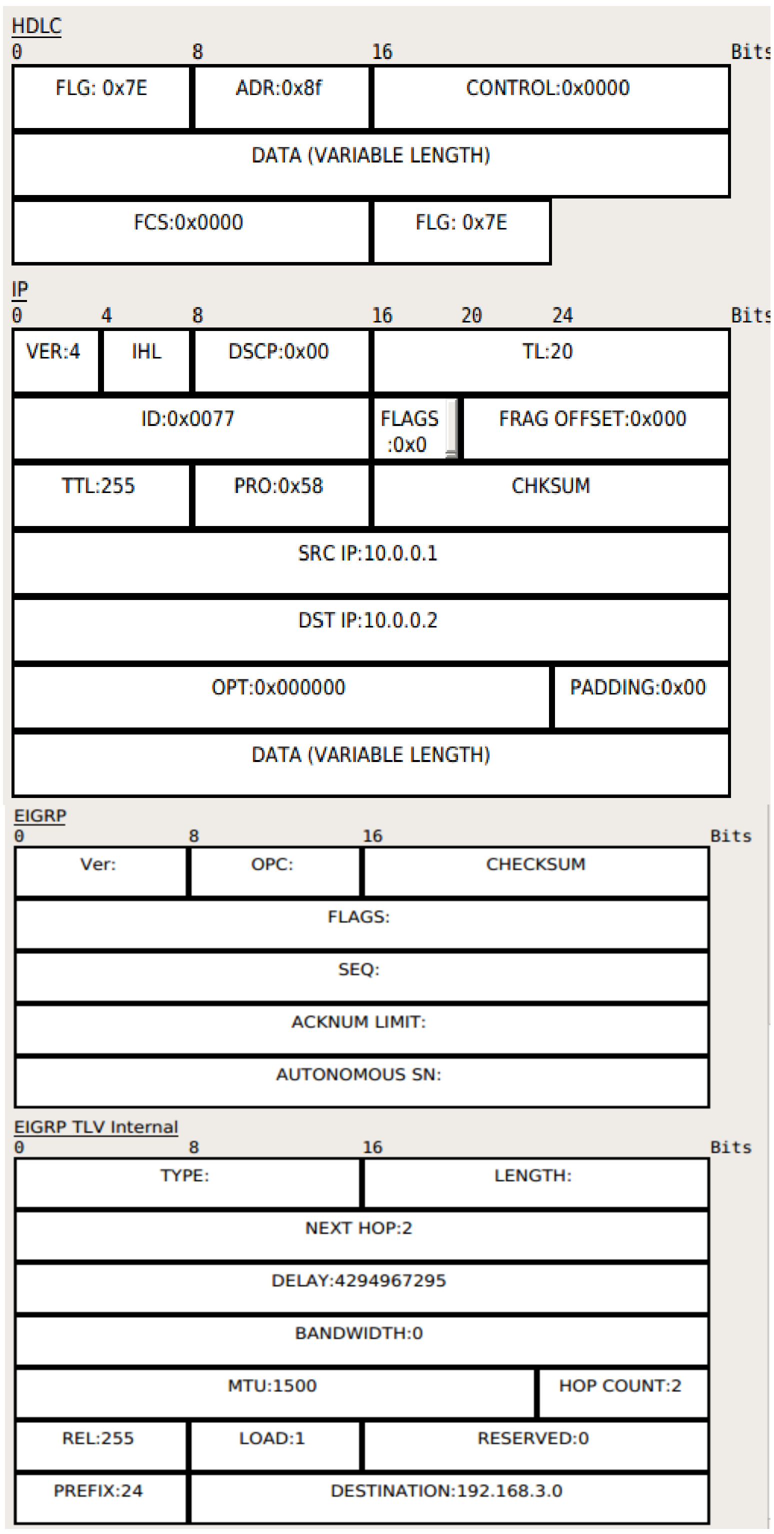
Router1

```
router#conf t
router(config)#hostname r1
r1(config)#int se2/0
```

```
r1(config-if)#ip address 10.0.0.2 255.0.0.0
r1(config-if)#exit
r1(config)#int se3/0
r1(config-if)#ip address 11.0.0.1 255.0.0.0
r1(config-if)#exit
r1(config)#int fa0/0
r1(config-if)#ip address 192.168.2.100 255.255.255.0
r1(config-if)#exit

r1(config)#router eigrp 10
r1(config-router)#network 10.0.0.0
r1(config-router)#network 192.168.2.0
r1(config-router)#network 11.0.0.0
```

Router2

```
router#conf t
router(config)#hostname r2
r2(config)#int se2/0
r2(config-if)#ip address 11.0.0.2 255.0.0.0
r2(config-if)#exit
r2(config)#int fa0/0
r2(config-if)#ip address 192.168.3.100 255.255.255.0
r2(config-if)#exit

r2(config)#router eigrp 10
r2(config-router)#network 11.0.0.0
r2(config-router)#network 192.168.3.0
```



HDLC

0 8 16 Bits

FLG: 0x7E	ADR:0x8f	CONTROL:0x0000
DATA (VARIABLE LENGTH)		
FCS:0x0000	FLG: 0x7E	

IP

0 4 8 16 20 24 Bits

VER:4	IHL	DSCP:0x00	TL:20	
ID:0x0077			FLAGS :0x0	FRAG OFFSET:0x000
TTL:255		PRO:0x58	CHKSUM	
SRC IP:10.0.0.1				
DST IP:10.0.0.2				
OPT:0x000000				PADDING:0x00
DATA (VARIABLE LENGTH)				

EIGRP

0 8 16 Bits

Ver:	OPC:	CHECKSUM
FLAGS:		
SEQ:		
ACKNUM LIMIT:		
AUTONOMOUS SN:		

EIGRP TLV Internal

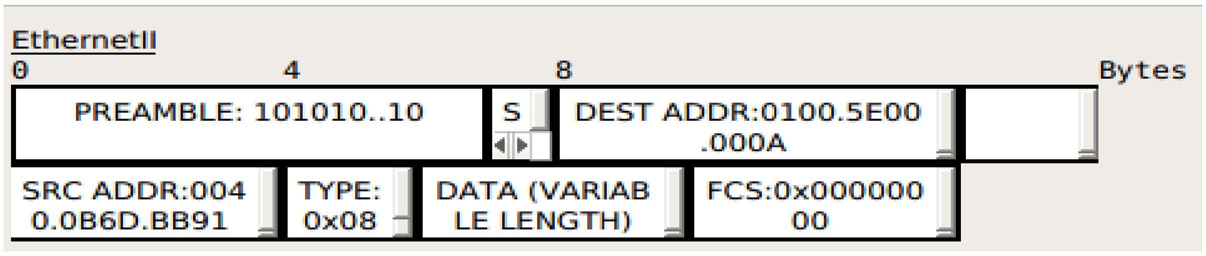
0 8 16 Bits

TYPE:		LENGTH:	
NEXT HOP:2			
DELAY:4294967295			
BANDWIDTH:0			
MTU:1500			HOP COUNT:2
REL:255	LOAD:1	RESERVED:0	
PREFIX:24	DESTINATION:192.168.3.0		

Packet between two routers

Next Hop : Next hop is the net-hop neighbouring router IP address.
Delay : Delay is the sum of the configured delays expressed in units of 10 microseconds. A delay of 0xffff ffff or 256 indicates an unreachable route.
Bandwidth : Bandwidth is 256 * BW (min) or 2,560,000,000 divided by the lowest configured bandwidth of any interface along the route.
MTU : MTU is the smallest maximum Transimission unit of any link along the route to the distination. This value is not used for the metric calculation.
Hop Count : Hop count is a number between 0x01 and 0xFF indicating the no of hops to the destination. A router will advertise a directly connected network with a hop count of 0.
Reliability : Reliability is a number between 0x01 and 0xFF that reflects along the routes calculated on a five-minute exponentially weighted average 0xFF indicates a 100 percent reliable link.
Load : Load is also a number between 0x01 and 0xFF reflecting the total outgoing load of the interfaces along the route, calculated on a five-minute exponentially weighted average 0x01 indicates a minimally loaded link.
Reserved : Reserved is an unused field and is always 0x000.
Prefix Length: Specifies the no of network bits of the address mask.
Destination : Destination is the address of the route.

Ethernet fields of EIGRP : If we are sending the packet from the **Switch to the router** or **switch to PC** there no need of **HDLC Header** here we need the **Ethernet header** but if we are sending the packet from **Router to Router t**here is no of **Ethernet header** but we need the **HDLC**



EthernetII

0 4 8 Bytes

PREAMBLE: 101010..10 | S | DEST ADDR:0100.5E00 .000A

SRC ADDR:0040.0B6D.BB91 | TYPE: 0x08 | DATA (VARIABLE LENGTH) | FCS:0x00000000

DEST ADDET : 0100.5E00.000A is the hex fromat of multicast address **224.0.0.10**

Multicast

Types of “packets”

Three types of packets can exist on an IPv4 network:
Unicast – A packet sent from one host to only one other host. A hub will forward a unicast out all ports. If a switch has a table entry for the unicast’s MAC address, it will forward it out only the appropriate port.
Broadcast – A packet sent from one host to all hosts on the IP subnet. Both hubs and switches will forward a broadcast out all ports. By definition, a router will not forward a broadcast from one segment to another.
Multicast – A packet sent from one host to a specific group of hosts. Switches, by default, will forward a multicast out all ports. A router, by default, will not forward a multicast from one segment to another.

Multicast Concepts

Remember, a multicast is a packet sent from one computer to a group of hosts. A host must join a multicast group in order to accept a multicast. Joining a multicast group can be accomplished statically or dynamically.

Multicast traffic is generally sent from a multicast server, to multicast clients. Very rarely is a multicast packet sent back from a client to the server.

Multicasts are utilized in a wide range of applications, most notably voice or video systems that have one source “serving” out data to a very specific group of clients.

The key to configuring multicast is to ensure only the hosts that require the multicast traffic actually receive it.

Multicast Addressing

IPv4 addresses are separated into several “classes.”
Class A: 1.1.1.1 – 127.255.255.255
Class B: 128.0.0.0 – 191.255.255.255
Class C: 192.0.0.0 – 223.255.255.255
Class D: 224.0.0.0 – 239.255.255.255

Class D addresses have been reserved for multicast. Within the Class D address space, several ranges have been reserved for specific purposes:

• **224.0.0.0 – 224.0.0.255 –** Reserved for routing and other network protocols, such as OSPF, RIP, VRRP, etc.

• **224.0.1.0 – 238.255.255.255 –** Reserved for "public" use, can be used publicly on the Internet. Many addresses in this range have been reserved for specific applications

• **239.0.0.0 – 239.255.255.255 –** Reserved for "private" use, and cannot be routed on the Internet.

The following outlines several of the most common multicast addresses
reserved for routing protocols:

• 224.0.0.1 – all hosts on this subnet
• 224.0.0.2 – all routers on this subnet
• 224.0.0.5 – all OSPF routers
• 224.0.0.6 – all OSPF Designated routers
• 224.0.0.9 – all RIPv2 routers
• 224.0.0.10 – all IGRP routers
• 224.0.0.12 – DHCP traffic
• 224.0.0.13 – all PIM routers
• 224.0.0.19-21 – ISIS routers
• 224.0.0.22 – IGMP traffic
• 224.0.1.39 – Cisco RP Announce

• 224.0.1.40 – Cisco RP Discovery

Multicast MAC Addresses

Unfortunately, there is **no ARP equivalent protocol** for multicast addressing. Instead, a reserved range of **MAC addresses** were created for multicast Ips. All **multicast MAC addresses b**egin with:

0100.5e

Recall that the first six digits of a MAC address identify the vendor code, and the last 6 digits identify the specific host address. To complete the MAC address, the last 23 bits of the multicast IP address are used.

For example, consider the following multicast IP address and its binary
equivalent:

224.65.130.195 = 11100000.01000001.10000010.11000011

Remember that a **MAC address is 48 bits long**, and that a **multicast MAC must begin with 0100.5e.** In binary, that looks like:

00000001.00000000.01011110.0

Add the **last 23 bits of the multicast IP address to the MAC**, and we get:

00000001.00000000.01011110.01000001.10000010.11000011

That should be **exactly 48 bits long**. Converting that to Hex format, **our full MAC address would be:**

0100.5e41.82c3

How did I convert this to Hex? Remember that hexadecimal is Base 16 mathematics. Thus, to represent a single hexadecimal digit in binary, we would need **4 bits (2^4 = 16).** So, we can break down the above **binary MAC address into groups of four bits:**

Binary	**0000**	**0001**	**0000**	**0000**	**0101**	**1110**	**0100**	**0001**	**1000**	**0010**	**1100**	**0011**
Decima1	**0**	**1**	**0**	**0**	**5**	**14**	**4**	**1**	**8**	**2**	**12**	**3**
Hex	**0**	**1**	**0**	**0**	**5**	**e**	**4**	**1**	**8**	**2**	**c**	**3**

Hence the MAC address of 0100.5e41.82c3.

Ready for some more math, you binary fiends?
Calculate what the multicast MAC address would be for the following IP addresses:

225.2.100.15 = 11100001.00000010.01100100.00001111
231.130.100.15 = 11100111.10000010.01100100.00001111

Remember that all multicast MACs begin with:

0100.5e = 00000001.00000000.01011110.0

So, add the last 23 digits of each of the above IP addresses to the MAC address, and we get:

225.2.100.15 = 00000001.00000000.01011110.00000010.01100100.00001111
231.130.100.15 = 00000001.00000000.01011110.00000010.01100100.00001111

In Hex, that would be:

225.2.100.15 = 0100.5e02.640f
231.130.100.15 = 0100.5e02.640f

Wait a second.... That's the exact same multicast MAC address, right? Double-checking our math, we see that it's perfect.

Believe it or not, each multicast MAC address can match 32 multicast IP addresses, because we're only taking the last 23 bits of our IP address.
https://www.google.com/search?q=dhc+p&ie=utf-8&oe=utf-8&client=firefox-b-ab
We already know that all multicast IP addresses MUST begin 1110. Looking at the 225.2.100.15 address in binary:

11100001.00000010.01100100.00001111

That leaves 5 bits in between our starting 1110, and the last 23 bits of our IP. Those 5 bits could be anything, and the multicast MAC address would be the same. Because 2 5 = 32, there are 32 multicast IP's per multicast MAC. -

According to the powers that be, the likelihood of two multicast systems utilizing the same multicast MAC is rare. The worst outcome would be that hosts joined to either multicast system would receive multicasts from both.

Inter– VLAN Routing Protocol

In previous chapters,we learnt how **VLANs** segment broadcast traffic on a switch and segment a switched network into different **LANs,**we also learnt how **VLANs** information can be transmitted to other switches in the network using **VTP** and how we can avoid layer 2 loops using **STP**.

Consider,this as the network administrator one of your task is to create and assign different users to **VLANs** in your network,you have three main departments which should be logically segmented using **VLANs,VLAN 10 – FINANCE,VLAN 20 – SALES and VLAN 30 – HR.**

The use of **VLANs** means that users would not be able to communicate across departments,i.e a user in **FINANCE,**would not be send a message to a user in SALES since they are on different broadcast domains.

We will discuss the role played by **inter-VLAN** routing in communication between different **VLANs.**We will learn how it works.Consider the various methods that can be used to implement it.

Consider **inter-VLAN** routing using route-on-a-stick and traditional **inter-VLAN** routing,compare the 2 style of implementation and finally verify and troubleshoot **inter-VLAN** routing.

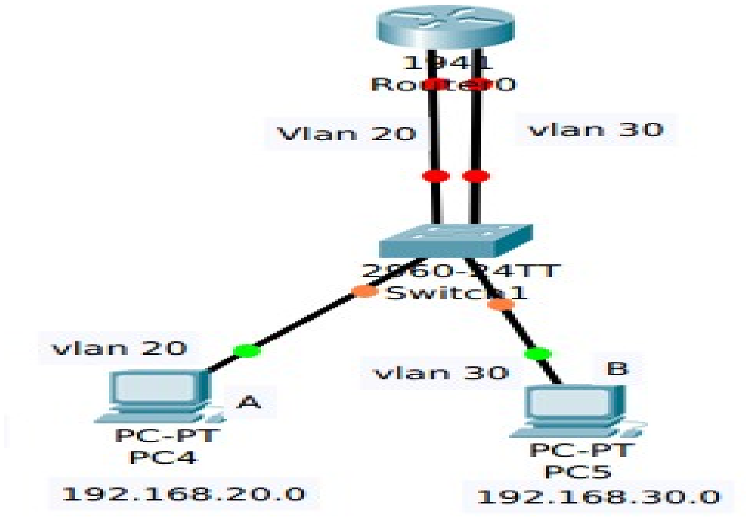
Introduction to Inter-Vlan-routing

When we learnt about VLANs,we said that each VLANs is usually on its own subnet switches mainly operates at layer 2 of the OSI model and therefore they do not examine the logical address.Therefore,user nodes located on different VLANs cannot communicates by default.In many cases we need connectivity between users located on different VLANs.The way this can be accomplished is through inter-VLAN routing.So we should maintain router.

Definition :

Inter-VLAN routing can be defined as a way to forward traffic between different **VLAN** by implementing a router in the network.As we learnt previously,**VLANs** logically segment the switch into different subnets when a router is connected to the switch an administrator can configure the router to forward the traffic between the various **VLANs** configured on the switch.The user nodes in the **VLANs** forwards traffic to the router which thhttps://www.google.com/search?q=dhc+p&ie=utf-8&oe=utf-8&client=firefox-b-aben forwards the traffic to the destination network regardless of the **VLAN** configured on the switch.

The figure below show this process work



ping for PC A to PC B

Information destined for PC B leaves PC A with VLAN 20 tag when it gets to Router changes the format of this message from VLAN 20 to VLAN 30 it then sends it back to the switch and the switch finally sends the message to its intended recipient PC B.

<u>There are 2 ways in which inter-VLAN routing can be accomplished</u>
<u>Traditional inter-VLAN routing :</u> Router-on-a-stick
Traditional inter-VLAN routing : In this type of inter-VLAN routing,a router is usually connnected to the switch using multiple interfaces.One for each VLAN.The interfaces on the router are configured as the default gateways for the VLANs configured on the stick.

The ports that connect to the router from the switch are configured in **access mode** in their corresponding **VLANs** when a user node send a message to a user connected to a different **VLANs** the message moves from their mode to the **access port** that connects to the router on their VLAN.When the router receives the packet,it examines the packet's destination IP address and forwards it to the correct network using the **access port** for the destination node since the router changed the **VLAN** information from the source **VLAN** to the destination **VLAN.**

In this form of **inter-VLAN** routing the router has to have as many **VLAN interfaces** as the no of VLANS configured on the switch.Therefore,if the switch as **10 VLANs** interface the router should have the same no of LAN interfaces

<u>Take the secenario shown below</u>
If **PC A** in **VLAN 20,**wanted to send a message to **PC B** in **VLAN 30,**the steps it would take are shown below.

1) **PC A** would check whether the destination **IPV4 address** is in the **VLAN** if it is not,it should need to forward the traffic to its **default gateway** which is the ip address on Fa0/0 on R1.

2) **PC A** then sends an **ARP requets** to **AS1** so as to determine the physical address of

fa0/0 on Router.Once the router replies, **PC A** can send the frame to the router as a unicast message,since AS1 has fa0/0's the MAC address,it can forward the frame directly to Router.

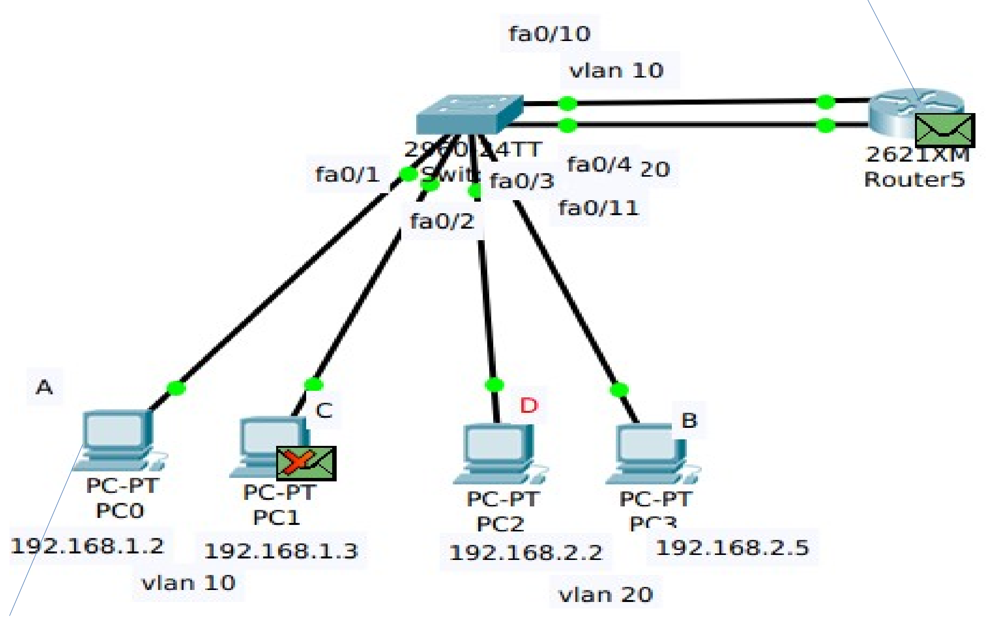
3) When the router receive frame,it compared the destination IP addresses by referring to its routing table so as to know to which interface it should send the data towards the destination node.

4) The router then sends an **ARP request** out the interface connected to the destination **VLAN** in this case out **fa0/1,**when the switch receives the message,it would flood it to its ports and in this case,**PC B** would reply with its **MAC address.**

5) Router would then use this information to frame the packet and finally send it to **PC B** as a **unicast frame.**

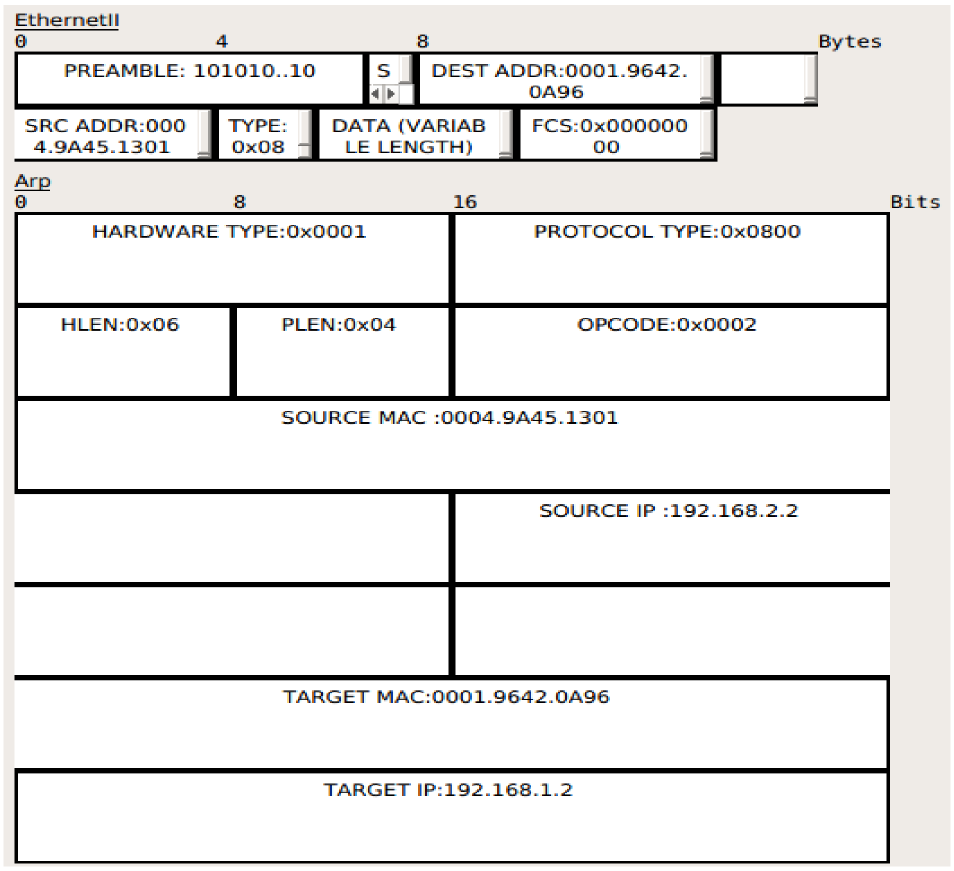
Device	Interface	IP address	subnet mask	default gateway
PC A	NIC	192.168.20.2	255.255.255.0	192.168.20.1
PC B	NIC	192.168.30.3	255.255.255.0	192.168.30.1
Router	Fa0/0 Fa0/1	192.168.20.1 192.168.30.1	255.255.255.0 255.255.255.0	

Mac address 0004.9A45.1301



Mac address 0001.9642.0A96

In the 1st packet of ARP the PC A will broadcast with the dest mac address as FFFF.FFFF.FFFF and gets the mac address of the router and then communicates with the PC B by getting the mac address of the PC B.



Inter-VLAN routing router-on-a-stick

With the example shown above,there are several concerns,suppose we had 10 or even 20 VLANs configured on the switch,even if the switch has enough ports to supports the connection to the router,it is highly unlikely that the router would have so many ethernet interfaces .Therefore we need a way to use the limited router information to support routing between many VLANs that way be on a switch.

Introduction to Router-on-a-stick

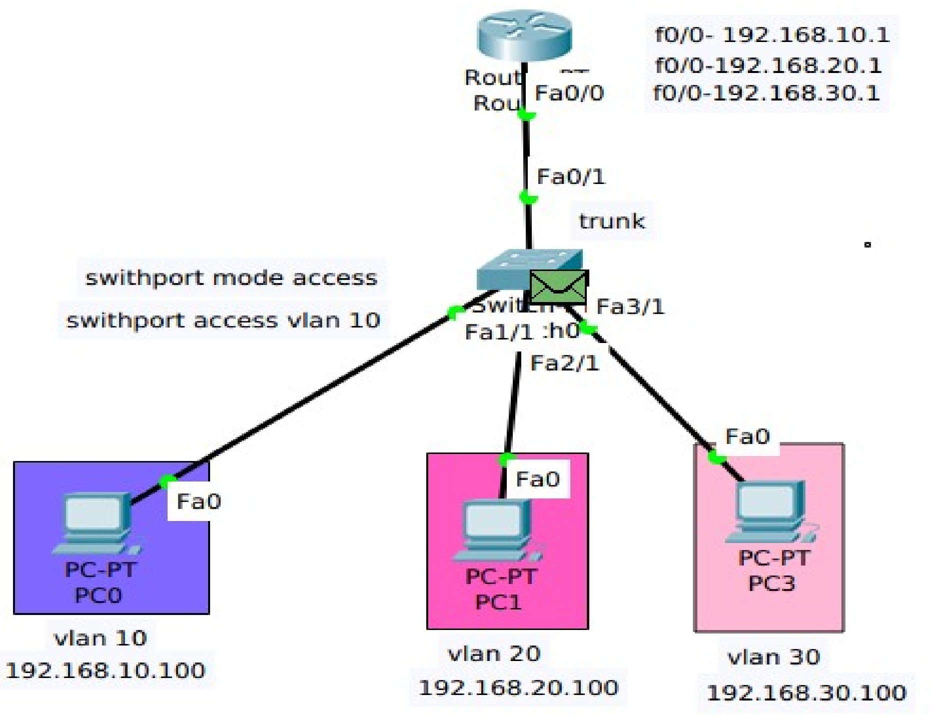
In the second type of inter-VLAN routing which is router-on-a-stick,the router is connected to the switch using a single interface.The switchport connecting to the router is configured as a **Trunk Link**.The single interface on the router is then configured with multiple IP addresses that correspond to the VLANs on the switch.This interface accepts traffic from all the VLANs and determine the destination network based on the source and destination IP in the packets.It then forwards the data to the switch with the correct VLAN information.

As you can see in the diagram below,the router is connected to the switch AS1 using a single ,physical network connection.

In this type of inter-VLAN routing the interface connecting the router to the switch is usually a **trunk link.** The router accepts traffic that is **tagged from the VLANs** on the switch through the **trunk link**.On the router the physical interface is divided into smaller interfaces called **Subinterfaces.**When the router receives the tagged traffic it forwards the traffic out of the **subinterface** that has the destination IP address.

Subinterfaces aren't real interfaces but they use the VLAN physical interfaces on the router to forward data to various **VLANs**.Each subinterfaces is configured with an IP address and assigned a **VLAN** based on the design.

Configure inter-VLAN routing using router-on-a-stick



In this section,we will configure **inter-VLAN routing** using router-on-a-stick and using the topology shown above.It has been modified by adding additional VLANs so as to show the effectiveness of using router-on-a-stick as opposed to traditional **inter-VLAN routing.**

Configuration of the inter VLAN routing-on-a-stick

configuration on the switch

```
switch>en
switch#conf t
switch(config)#hostname sw1
sw1(config)#vlan 10
sw1(config-if)#name 10
sw1(config-if)#vlan 20
sw1(config-if)#name 20
sw1(config-if)#vlan 30
sw1(config-if)#name 30
sw1(config-if)#exit

sw1(config)#vlan 10
sw1(config-if)#ip address 192.168.10.100 255.255.255.0
sw1(config-if)#exit
sw1(config)#vlan 20
sw1(config-if)#ip address 192.168.20.100 255.255.255.0
```

```
sw1(config-if)#exit
sw1(config)#vlan 30
sw1(config-if)#ip address 192.168.30.100 255.255.255.0
sw1(config-if)#exit
sw1(config)#int fa1/1
sw1(config-if)#switchport mode access
sw1(config-if)#switchport access vlan 10
sw1(config-if)#exit

sw1(config)#int fa2/1
sw1(config-if)#switchport mode access
sw1(config-if)#switchport access vlan 10
sw1(config-if)#exit

sw1(config)#int fa3/1
sw1(config-if)#switchport mode access
sw1(config-if)#switchport access vlan 10
sw1(config-if)#exit

sw1(config)#int fa0/1
sw1(config-if)#swithport mode trunk
sw1(config-if)#swithport trunk allowed vlan 10,20,30
```

Configuration on the router

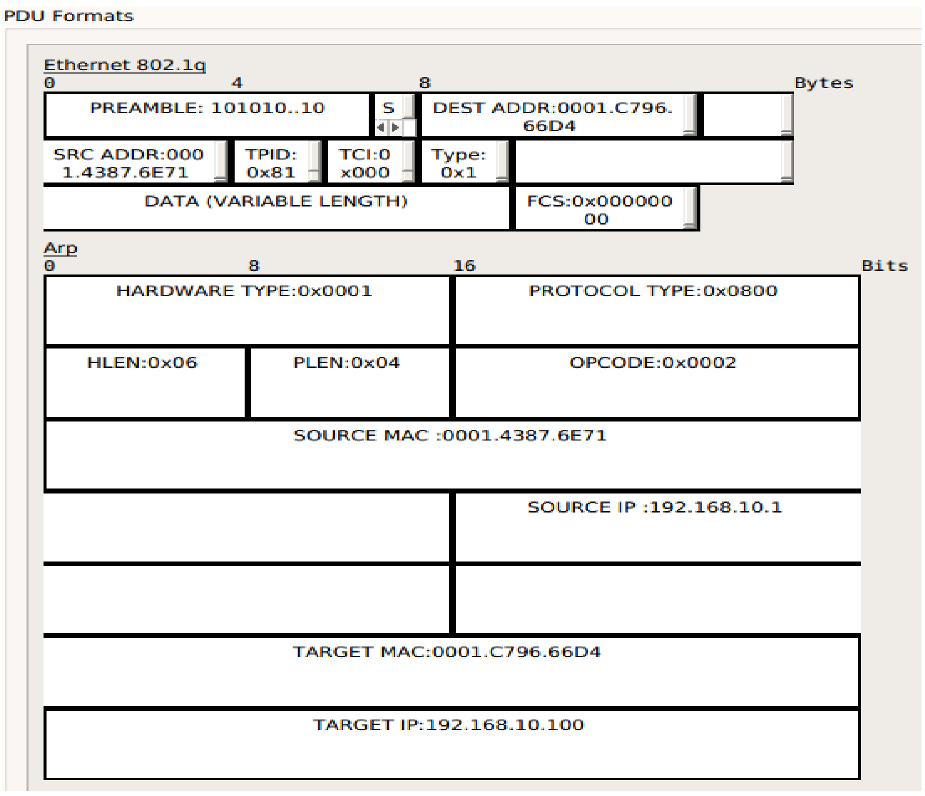
```
router>en
router#conf t
router(confighostname R1
R1(configint fa0/0
R1(config-if)#no shutdown
R1(config-if)#exit

R1(config)#int fa0/0.10
R1(config-subif)#encapsulated dot1q  10
R1(config-subif)#ip address 192.168.10.1 255.255.255.0
R1(config-subif)#exit

R1(config)#int fa0/0.20
R1(config-subif)#encapsulated dot1q  20
R1(config-subif)#ip address 192.168.20.1 255.255.255.0
R1(config-subif)#exit

R1(config)#int fa0/0.30
R1(config-subif)#encapsulated dot1q  30
```

```
R1(config-subif)#ip address 192.168.30.1 255.255.255.0
R1(config-subif)#end
```

Encapsulated with the 802.1q virtual lan ,vlan id

packet capture of inter vlan routing-on-a-stick

The packet is from switch to router and also same packet of router to switch use the Encapsulated with the 802.1q virtual lan ,vlan id


PDU Formats

Ethernet 802.1q

0 4 8 Bytes

PREAMBLE: 101010..10 | S | DEST ADDR:0001.C796.66D4

SRC ADDR:0001.4387.6E71 | TPID: 0x81 | TCI:0x000 | Type: 0x1

DATA (VARIABLE LENGTH) | FCS:0x00000000

Arp

0 8 16 Bits

HARDWARE TYPE:0x0001 | PROTOCOL TYPE:0x0800

HLEN:0x06 | PLEN:0x04 | OPCODE:0x0002

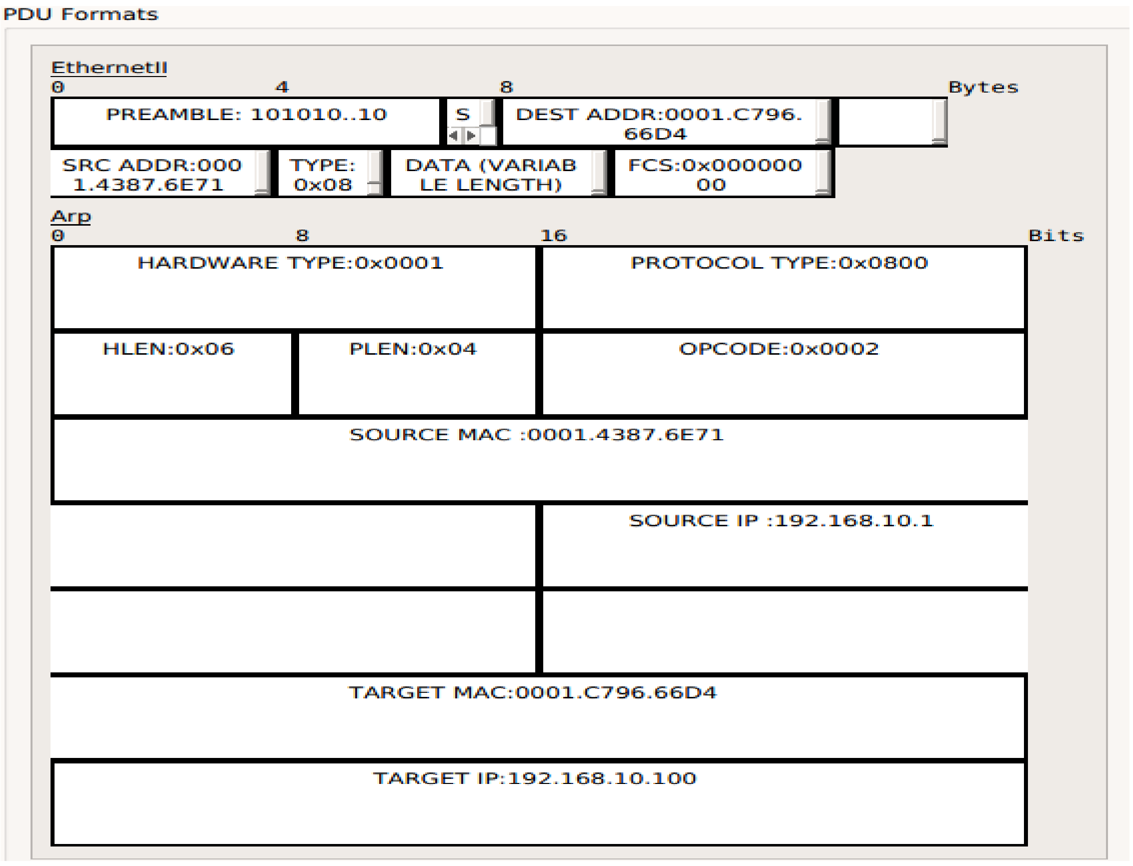
SOURCE MAC :0001.4387.6E71

SOURCE IP :192.168.10.1

TARGET MAC:0001.C796.66D4

TARGET IP:192.168.10.100

The packet which is from PC to Switch and Switch to PC will not use the Encapsulated with the 802.1q virtual lan ,vlan id



<u>inter-VLAN multilayer switch configuration</u>

A multilayer switch, known also as **Layer 3 Switch**, is a hybrid device combining a switch with a router. From a physical perspective, multilayer switches are identical to traditional switches. They can have 24 or 48 ethernet ports, and some SFP-ready ports. Furthermore, they can be stackable just like a Layer 2 Switch. The real difference is inside the box, both in *hardware* and *software*.

Multilayer switches comes with powerful *ASICs* (Application-Specific Integrated Circuits), hardware components that can be used for Layer 3 functionalities. One of the most important ASIC you can find on a Cisco Multilayer Switch is the **TCAM**, *Ternary Content-Addressable Memory*. With this equipment, Multilayer switches can perform **hardware routing**. As a result, they are much faster than routers for that, because they have dedicated hardware for that.

A traditional switch can only have a single IP address for management purposes. Multilayer switches can have **multiple IP addresses** instead, can support static routes and dynamic routing protocols.

Multilayer Switch vs. Router

If Multilayer Switches are faster than routers, *why would you use a router at all?* We need to explore the **key differences** between Multilayer Switches and Routers.

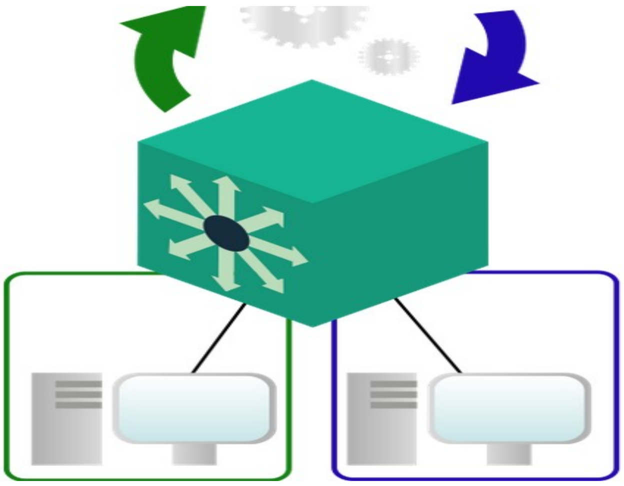
Multilayer switches are *very specific*. Their TCAM is designed to work with ethernet, and ethernet only (both copper and fiber). Router, instead, can work with many different technologies and supports **different interfaces**. Instead, in a router you can add some interfaces by inserting the appropriate module. This is not a feature you can find on a Multilayer switch.

Furthermore, Routers are extremely **feature-rich**. They support a lot of different protocols, fine tunings and customizations. Since Multilayer switches try to do everything they can in-hardware, they cannot have this granularity in the configuration, and they offer a limited set of features. These features may be good for the majority of uses, but there are some things that a Layer 3 switch simply can’t do.

Multilayer Switch for Inter-VLAN Routing

With Router on a Stick, our switches sent the traffic to a Router. Then, the router sent back the traffic to the switches after doing the frame rewrite and routing. With a Multilayer switch, *everything happens inside the Layer 3 switch*.

Multilayer switches supports **Switch Virtual Interfaces (SVIs)**, logical interfaces that can perform routing. They behave like a physical interface of a router: they have an IP address, and they insert a connected route into the routing table. However, they are completely virtual. You can have one SVI for each VLAN. When the switch receives a packet in a VLAN, which is intended *at Layer 2 for the switch itself* (MAC address of the SVI as destination), the switch performs routing.


All routing is performed inside the multilayer switch.

Since SVIs are logical interfaces, the *MAC address* associated with them is crafted in software, and *not burnt-in* into the device hardware. All frame-rewirte operations are performed inside the Multilayer switch. Furthermore, the Multilayer switch have all the feature of a traditional switch, and can support trunks to connect other traditional or multilayer switches.

Configuring the Multilayer Switch

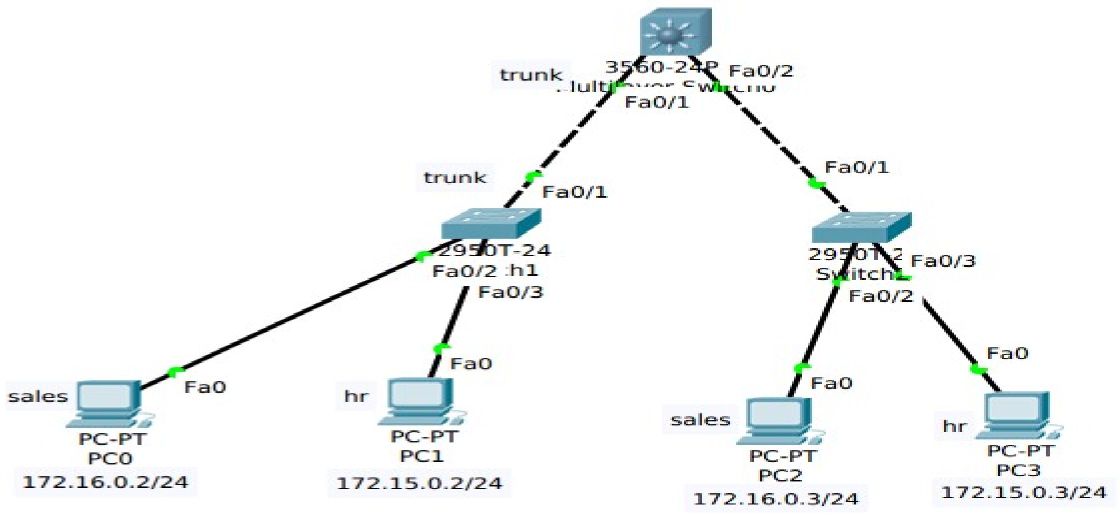
Turn on Routing

As stated earlier, the only configuration item we need to touch is the Multilayer Switch. For this lab, we are talking about the device in the middle: *“MLSW”*. It is a Legacy *Cisco Catalyst 3560*, the best that Packet Tracer can offer.

The very first step is to enable routing on this Multilayer switch. By default, these multilayer switches comes configured as traditional switches. In case you want to use them as Layer 2 switches, you need to enable this feature with a simple command. To do that, type `ip routing` in the configuration prompt. This is platform-dependent, some devices may be configured as multilayer switches in the factory defaults. Cisco 3560, however, are not these devices, and need you to manually type this command.

Once you type `ip routing`, nothing will happen apparently. However, the device has just constructed a routing table, and allocated the resources to work with it. This command is effective

immediately, you won’t need to do a reboot or anything else: it just works.



configuration

Multilayer Switch Configuration

switch>**en**
switch#**conf t**
switch(config)#**hostname ML1**
switch(config)#**vlan 2**

```
switch(config-if)#name sales
ML1(config-if)#vlan 3
ML1(config-if)#name hr
ML1(config-if)#exit
ML1(config)#int vlan 2
ML1(config-if)#ip address 172.16.0.1 255.255.255.0
ML1(config-if)#exit
ML1(config)#int vlan 3
ML1(config-if)#ip address 172.15.0.0 255.255.255.0
ML1(config-if)#exit

ML1(config)# int fa0/1
ML1(config-if)#switchport mode trunk
ML1(config-if)#switchport trunk encapsulation dotlq
ML1(config-if)#exit
ML1(config)# int fa0/2
ML1(config-if)#switchport mode trunk
ML1(config-if)#exit
```

configuration on switch

```
switch#conf t
switch(conf)#hostname sw1
sw1#vlan database
sw1(vlan)#vlan 2 name sales
sw1(vlan)#vlan 3 name hr
sw1(vlan)#exit

sw1(conf-if)#int vlan 2
sw1(conf-if)#ip address 172.16.0.2 255.255.255.0
sw1(conf-if)#exit

sw1(conf)#int vlan 3
sw1(conf-if)#ip address 172.15.0.2 255.255.255.0
sw1(conf-if)#exit

sw1#conf t
sw1(config)#int fa0/2
sw1(config)#switchport mode access
sw1(config)#switchport access vlan
sw1(config)#int fa0/1
sw1(config)#switch mode trunk
```

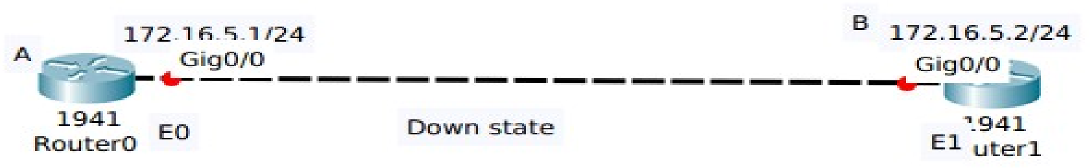
apply same on the switch 2

Open Shortest Path (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.

1) OSPF stand for Open Shortest Path First is a Standard Protocol.
2) It's a link state protocol.
3) It uses SPF (Shortest Path First) or Dijkistra Algorithm.
4) Unlimited hop count.
5) Metric is cost (cost = 10 ^ 8/Band width).
6) Administrative distance is 110.
7) It is a classless routing protocol.
8) It supports VLSM and CIDR.
9) It supports only equal cost load balancing.
10 Introduces the concept of Area's to ease management and control traffic.
11) Updates are sent through multicast address 244.0.0.5.
12) Faster Convergence
13) Sends Hello Packets every 10 seconds & Dead = 40 sec.
14) Incremental updates.

The Entire OSPF Process will goes in Seven Stages



Hello

I am Router A ID 172.16.5.1/24 to 244.0.0.5

Init State

Route B
Neighbours List
172.16.5.1/24 Interface
E1

I am Router B ID 172.16.5.2 and I See 172.16.5.1

Unicast to A

Hello

Router A
Neighbours List
172.16.5.2/24 Interface
EO

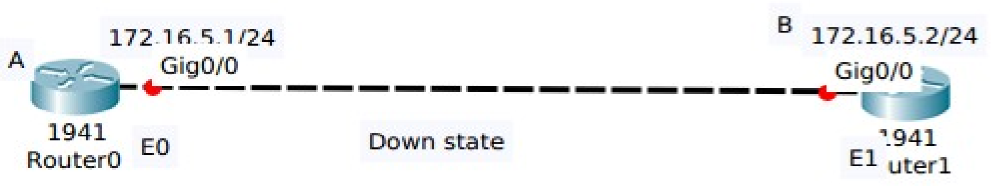
Two-Way state

<u>**Down Stage**</u> : First stage is **Down state** where the **Router A** will have the IP address of **172.16.5.1** and **Router B** will have the IP address **172.16.5.2**.In **Down State Route A** don't know

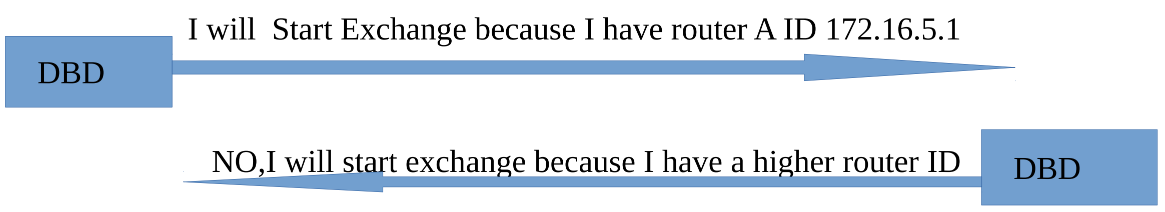
about the **Router B** and **Router B** don't know about the **Route A.**

<u>Second Stage INIT Stage</u> : In this Stage both the Routers will send the hello messages.
Two-Way State where they establish the neighbor relationship. They build the neighbor table.

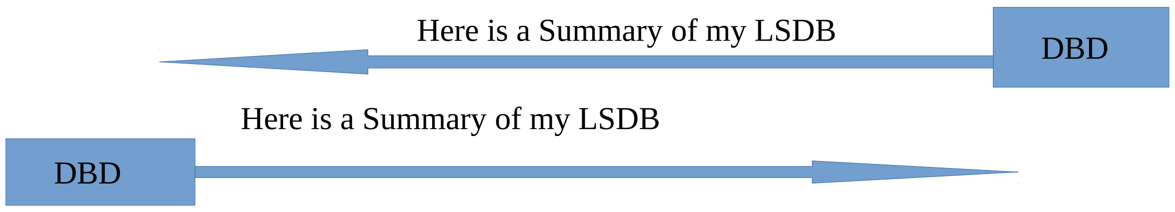
ii)Data Base Description (DBD)



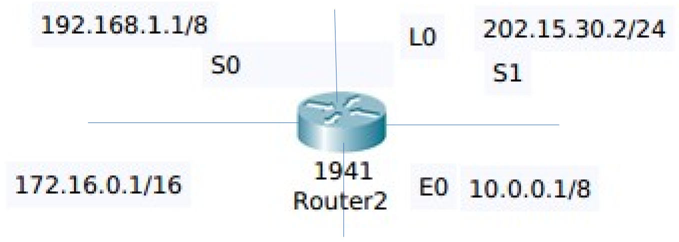
Exstart State


I will Start Exchange because I have router A ID 172.16.5.1
DBD
NO,I will start exchange because I have a higher router ID
DBD

Exchange State



<u>Exstart State</u> :
1) This Stage is Completly based on the **Router ID.**
2) The **height IP address** of the active physical interface of the router is **Router ID**
3) If **Logical interface** is configured,the highest **IP address** of the **logical interface** is **Router ID.**



In case **OSPF** the **Router** must be identified only by **one common name** it take the **highest IP** of the active **physical interface.**

1) We can either give manual **Router ID.**

2) By default it will take **highest IP** of the **Loop Back Interface.**
3) If there is no **Loop Back** it will take **highest IP** of the **Physical Interface.**

<u>OSPF Tables</u>
<u>Neighbor Table:</u>
1) Also Known as the **<u>adjacency database.</u>**
2) Contains list of directly connected **<u>routers (neighbors)</u>**.
3) **#show ip ospf neighbor**

<u>Database Table</u>
1) Typically referred to as **<u>LSDB (Link State Database)</u>** .
2) Contains information about all the possible routes to the network within the area.
3) **#show ip ospf database**

<u>Routing Table</u>
1) Contains list of best paths to each distination.
2) **#show ip route**

In the **<u>Exchange State</u>** all the neighbours will exchange the all the possible routes

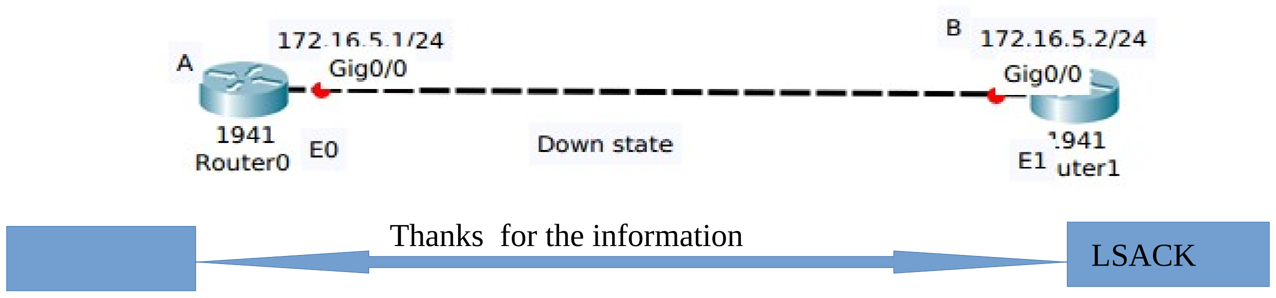
Distance Vector : Relays on the information given by the neighbour and it selects th best route.

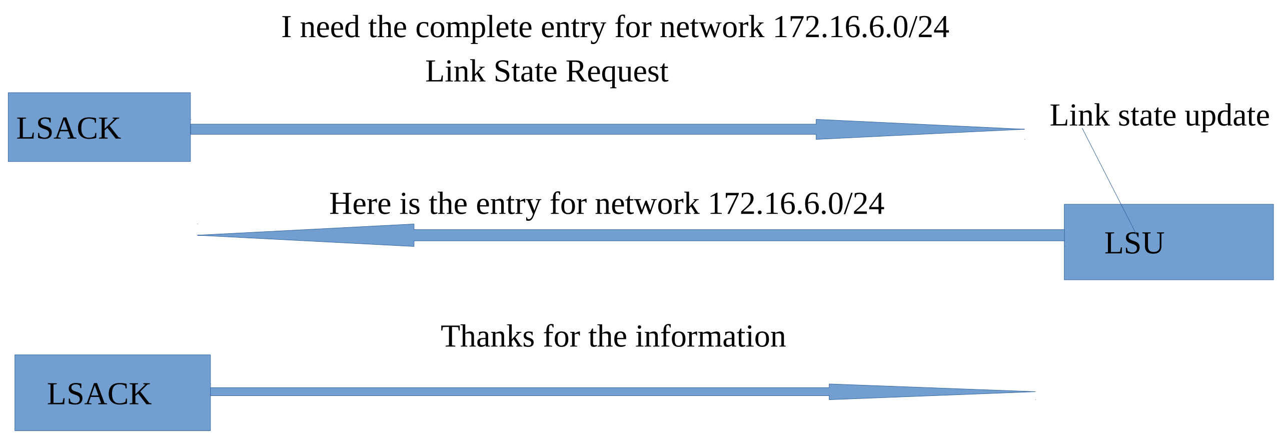
Link State : Even though it is learning about the particular network from the neighbour like PCD but still it is going to maintain each and every Link State Information.

Link State Mean :
1) How it is connected?
2) How many possible routes?
3) How many different routes? we have each and every possible **route information in OSPF database Table.**

OSPF datebase Table is like **google maps.**
All the **Routers** will have **same database information**

<u>Adding the Link-State Entries</u>




I need the complete entry for network 172.16.6.0/24
Link State Request
LSACK
Link state update
Here is the entry for network 172.16.6.0/24
LSU
Thanks for the information
LSACK

Full State

Link State Advertisement

LSA Type	**Description**
1	**Router Link State Advertisement**
2	**Network Link State Advertisement**
3 or 4	**Summary Link State Advertisement**
5	**Autonomous System external Link State Advertisement**
6	**Multicaste Link State Advertisement**
7	**Defined for not-so-stubby areas**
8	**External attributes Link State Advertisements for Border gateway protocol (BGP)**
9,10,11	**Opaque Link State Advertisement's**

Areas : OSPF divides an autonomous system into areas.An area is a collection of network of network,hosts,and routers all contained within an autonomous system.

Router inside an area flood the area with routing information.At the border of an area special routers called **area border routers .Summarize** the information about the area and send it to other **area**.Among the area inside an **autonomous system** is a special area called the **backbone** ,all the areas inside an **autonomous system** must be connected to the **backbone**.In other words,the **backbone** serves as a **primary area** and the other areas as secondary areas.This does not mean that the routers within areas cannot be connected to each other however.The **routers** inside the **backbone router** can also be an **area broader router**

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).As a matter of fact,a router can have multiple routing tables,each based on a differnt type of service

Type of Links :

1) Point to Point

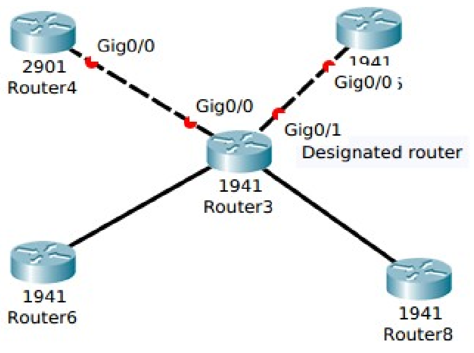
2) Transient

3) Stub
4) Virtual

<u>Point to Point</u> : A Point to Point link connects two packets without any other host or router in between

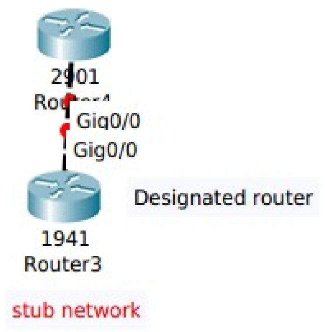
<u>Transient link</u>

A **transient link** is a network several routers attached to it.The data can enter through any of the **routers** and leave through any **router**.All **LANs** and some **WANs** with two or more routers are of this type.In this case,each router has many neighbours.



<u>Stub Link</u> :

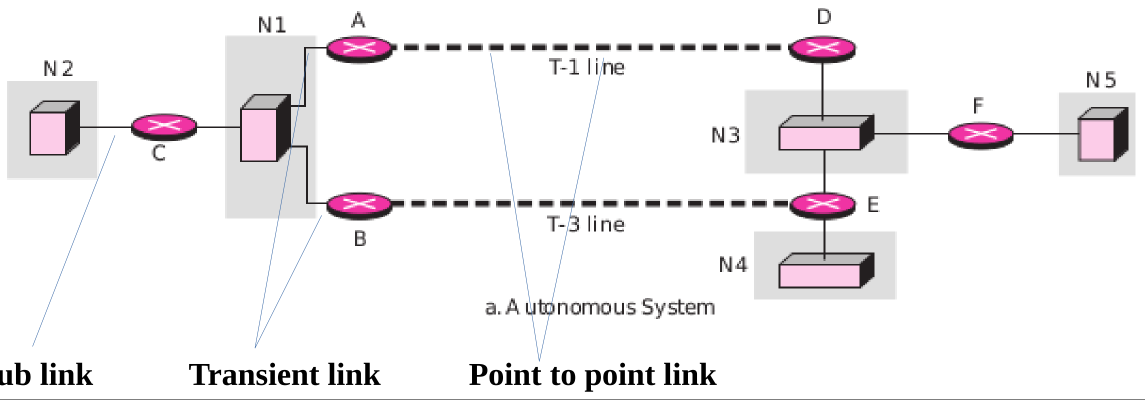
A **Stub Link** is a network that is connected to **only one router**.The data packets enter the network through this **single router** and **leave the network** through this **same router**.This is a special case of the **transient network**.We can show this situation using the router as a node and using the **designated router** for the network.



<u>Virtual Link</u> :

When the link between 2 routers is broken,the administrator may create a Virtual Link between them using a longer path that probably goes through several routers.

<u>Example of an AS in OSPF</u>



<u>OSPF Packet</u>

OSPF as Five Packets

1) Hello

2) Data Base Discription

3) Link State Request

4) Link State Update

5) Link State Knowledgement

1) Router Link State Update

2) Network Link State Update

3) Summary Link to Network

4) Summary Link to Autonomous System boundary router

5) External Link

Common Header : All **OSPF Packets** have the **common header** before studying the different types of Packets

0	**8**	**16**	**31**
Version	Type	Message Length	
Source Router IP Address			
Area Identification			
Checksum		Authentication type	
Authentication (32 bits)			

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 take place.

Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**

Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,when more **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password**

<u>Link State Update Packet :</u>

We first discuss the **Link State Update Packet,**the heart of the **OSPF operation.**It is used by a router to **advertise** the **states** of its links.The general format of the **Link State Update Packet**.

<u>Link State Update Packet :</u>

0 8 16 32

Version	Type	Message Length
Source Router IP Address		
Area Identification		
Checksum		Authentication type
Authentication (32 bits)		
Number of link state advertisements		
Link State Advertisement Any combination of five different kinds (Network link,Router link,Summary link to network,Summary to boundary router or external link)		

Each update packet may contain several different LSA's all five kinds have the same general header.This general header is shown

<u>Link State Advertisement General Header</u>

<u>Link State Age :</u>

This field indicates the no of **seconds elapsed** since this **message** was first generated.Recall that this type of message goes from **router to router (flooding)**.When a router creates the **message** the value of this field is **0.**When each **successive router** forwards this message,it estimates the **transit time** and adds it to the **cumulative value** of this field.

Link Stage Age	Reserved	E	T	Link State Type
Link State ID				
Advertising router				
Link State Sequence number				
Link State checksum				Length

<u>E-Flag :</u> If this **1-bit flag is set to 1**,it means that the area is a **Stub Area**.A **Stub Area** is an area that is connected to the backbone area by only one path

T-Flag : If this **1-bit is set to 1**,it means that the **router** can **handle multiple types of service.**

Link State Type : This field defines the LSA type.As we discussed before ,there are five different advertisement types.
1) Router Link
2) Network Link
3) Summary Link to Network
4) External Link

Link State ID :
The value of this field depends on the type of link.For **type 1(router link),**it is the I**P address of the router.**
For **type 2 (network link)**,it is the **IP address** of the **designated router**.For **type 3 (Summary link to network),**it is the address of the network **AS boundary router.**
For **type 4 (Summary Link to network).**It is the **IP address** of the **AS boundary router**.For **type 5 (external link)**,it is the address of the **external network.**

Advertising Router : This is the IP address ot the router advertising this message.

Link State Sequence Number : This is a sequence number assigned to each link state .

Link State Checksum : This is **not** the usual **checksum** instead,the value of this field is calculated using **Fletcher's checksum,**which is based on the **whole packet except for the age field.**

Length : This defines the **length of the whole packet in bytes.**

Router Link Link State Advertisement

A Router link defines the links of a true router.A true router uses this advertisement to annouce information about all of its links and what is at the link (neighbors).

The Router link LSA advertises all the links of a router (true router).The format of the router link packet.

Version	Type	Message Length		
Source Router IP Address				
Area Identification				
Checksum	Authentication type			
Authentication (32 bits)				
No of Advertisements				
Link Stage Age	Reserved	E	T	Link State Type
Link State ID				
Advertising router				
Link State Sequence number				
Link State checksum	Length			
Reserved	E	B	Reserved	No of router links
Link ID				
Link data				
Link type	#of Tos	Metric for Tos 0		
TOS	Reserved	Metric		

<u>Common Header :</u>

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**
Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password**

Link State Advertisement General Header

Link State Age :

This field indicates the no of **seconds elapsed** since this **message** was first generated.Recall that this type of message goes from **router to router (flooding)**.When a router creates the **message** the value of this field is **0.**When each **successive router** forwards this message,it estimates the **transit time** and adds it to the **cumulative value** of this field.

E-Flag : If this **1-bit flag is set to 1**,it means that the area is a **Stub Area**.A **Stub Area** is an area that is connected to the backbone area by only one path

T-Flag : If this **1-bit is set to 1**,it means that the **router** can **handle multiple types of service.**

Link State Type : This field defines the LSA type.As we discussed before ,there are five different advertisement types.

1) Router Link

2) Network Link

3) Summary Link to Network

4) External Link

Link State ID :

The value of this field depends on the type of link.For **type 1(router link),**it is the I**P address of the router.**

For **type 2 (network link)**,it is the **IP address** of the **designated router**.For **type 3 (Summary link to network),**it is the address of the network **AS boundary router.**

For **type 4 (Summary Link to network).**It is the **IP address** of the **AS boundary router**.For **type 5 (external link)**,it is the address of the **external network.**

Advertising Router : This is the IP address ot the router advertising this message.

Link State Sequence Number : This is a sequence number assigned to each link state .

Link State Checksum : This is **not** the usual **checksum** instead,the value of this field is calculated using **Fletcher's checksum,**which is based on the **whole packet except for the age field.**

Length : This defines the **length of the whole packet in bytes.**

ROUTER Header Fields :

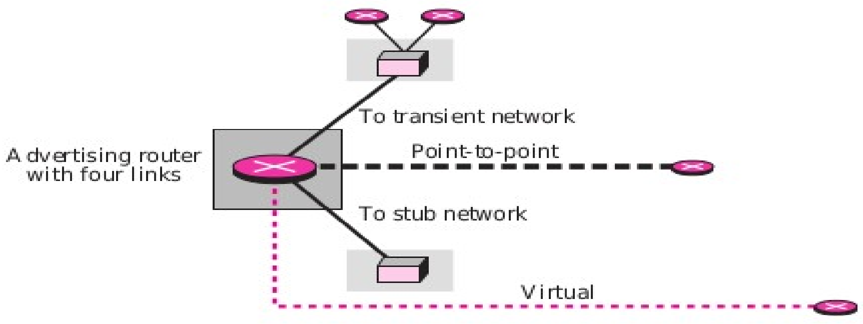
Link ID :

The value of this field depends on the type of link.

Link Data :

This field give additional information about the link. Again,the value depends on the type of the link.

Router Fields :



Link Type :

Four differnet types of links are defined based on the type of network to which the router is connected.

Link Type	Link Identification	Link Data
1.Type 1:Point-to-Point	Address of neighbor router	Interface numbered
2.Type 2:Trasient	Address of designated router	Router Address-family
3.Type 3:Stub	Network Address	Network Mask
3.Type 3:Virtual	Address of neighbor router	Router Address

No of Types Of Services (TOS):

This field defines the no of types of **services** announced for each **link-state.**

Metric for TOS 0:

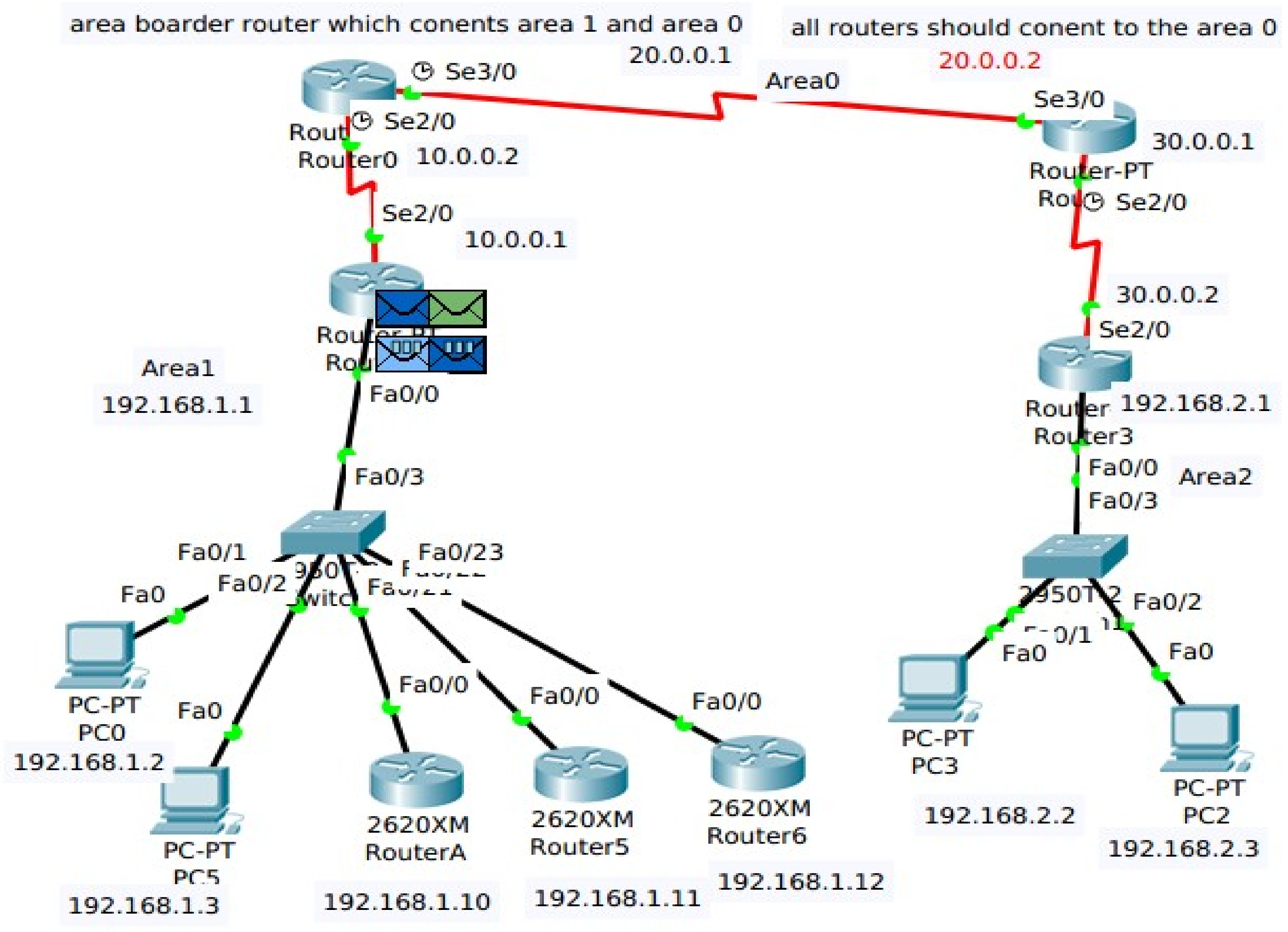
This field defines the **metric for the default type of service (TOS 0).**

TOS :

This field defines the **type of service.**

Metric :

This field defines the **metric** for the corresponding **TOS.**


area boarder router which conents area 1 and area 0
all routers should conent to the area 0
20.0.0.1
20.0.0.2
Area0
Se3/0
Se2/0
Router0
10.0.0.2
10.0.0.1
Se3/0
30.0.0.1
Router-PT
Se2/0
30.0.0.2
Se2/0
Router3
192.168.2.1
Fa0/0
Fa0/3
Area2
Area1
192.168.1.1
Fa0/0
Fa0/3
Fa0/1
Fa0/2
Fa0/23
Fa0
PC-PT
PC0
192.168.1.2
Fa0
PC-PT
PC5
192.168.1.3
Fa0/0
2620XM
RouterA
192.168.1.10
Fa0/0
2620XM
Router5
192.168.1.11
Fa0/0
2620XM
Router6
192.168.1.12
2950T
Fa0/2
Fa0
PC-PT
PC3
192.168.2.2
Fa0
PC-PT
PC2
192.168.2.3

OSPF LSU Packet

0 16 32

Version num : 2	Type : 4
Packet Length	
Router 1D : 192.168.1.1	
area ID : 0.0.0.1	
CHECK Sum : 0	AUTH TYPE :0
AUTHETICATION	
LSA :1	

OSPF Router LSA:

LSA AGE :0	OPTIONS : 0	LSA TYPE :1
LINK STATE : 192.168.1.1		
Advertising Router :192.168.1.1		
LS Sequence num : -2147483496		
CHECK SUM : 21598	Length : 60	
LSAS : 3		

OSPT Router Link :

Link ID : 20.0.0.1	Link data : 10.0.0.1	TYPE : 1
Metric : 64		

Link ID : 10.0.0.0	Link data : 255.0.0.0	TYPE : 3
Metric : 64		

Link ID : 192.168.1.0	Link data : 255.255.255.0	TYPE : 1
Metric : 64		

0 4 8 16 32

VER : 4	IHL	DSCP : 0x00	TL : 20	
ID :0xb091			FLAGS : 0x0	FRAG OFFSET : 0x000
TTL :1		PRO : 0x59	CHKSUM	
SRC IP : 10.0.0.1				
DES IP : 224.0.05				
OPT : 0x000000				PADDING : 0x00
DATA (VARIABLE LENGTH)				

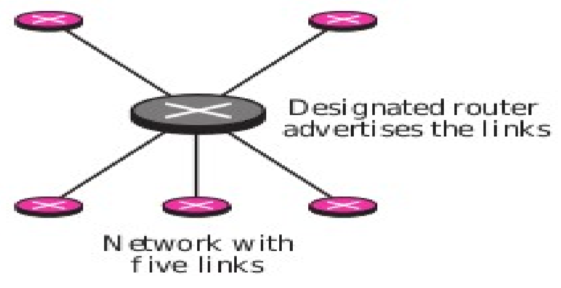
<u>HDLC</u>

FLG : 0x7E	ADDR : 0x8f	CONTROL : 0x0000
DATA (VARIABLE LENGTH)		
FCS : 0x0000		FLG : 0x7E

Network Link LSA :

A Network link defines the links of network.

A **designated router,**On behalf of the **transient network** distributes this type of **LSP packet**.The **Packet announces** the **existence** of all of the **routers** connected to the network.The format of the **Network Link.**



The Network Advertisement is shown in figure.The field of the network link LSA.

Version	Type	Message Length		
Source Router IP Address				
Area Identification				
Checksum	Authentication type			
Authentication (32 bits)				
No of Advertisements				
Link Stage Age	Reserved	E	T	Link State Type
Link State ID				
Advertising router				
Link State Sequence number				
Link State checksum	Length			
Network Mask				
Attached Router				

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**
Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password**

Link State Advertisement General Header

Link State Age :

This field indicates the no of **seconds elapsed** since this **message** was first generated.Recall that this type of message goes from **router to router (flooding)**.When a router creates the **message** the value of this field is **0.**When each **successive router** forwards this message,it estimates the **transit time** and adds it to the **cumulative value** of this field.
<u>E-Flag :</u> If this **1-bit flag is set to 1**,it means that the area is a **Stub Area**.A **Stub Area** is an area that is connected to the backbone area by only one path

<u>T-Flag :</u> If this **1-bit is set to 1**,it means that the **router** can **handle multiple types of service.**

<u>Link State Type :</u> This field defines the LSA type.As we discussed before ,there are five different advertisement types.
1) Router Link
2) Network Link
3) Summary Link to Network
4) External Link

<u>Link State ID :</u>
The value of this field depends on the type of link.For **type 1(router link),**it is the I**P address of the router.**
For **type 2 (network link)**,it is the **IP address** of the **designated router**.For **type 3 (Summary link to network),**it is the address of the network **AS boundary router.**
For **type 4 (Summary Link to network).**It is the **IP address** of the **AS boundary router**.For **type 5 (external link)**,it is the address of the **external network.**

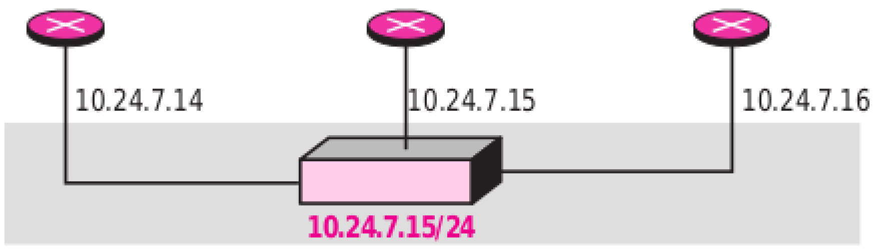
<u>Advertising Router :</u> This is the IP address ot the router advertising this message.

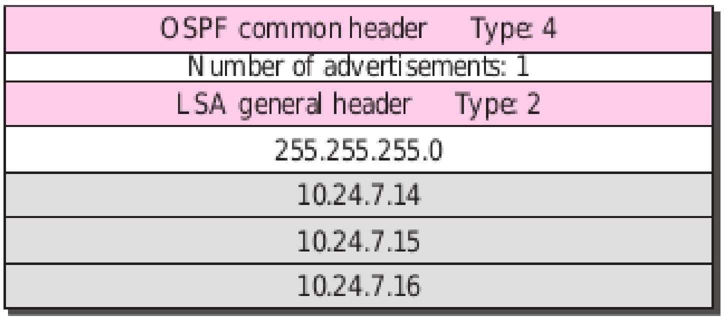
<u>Link State Sequence Number :</u> This is a sequence number assigned to each link state .

<u>Link State Checksum:</u> This is **not** the usual **chechttps://globaledge-pms.synergita.com/LogOnksum** instead,the value of this field is calculated using **Fletcher's checksum,**which is based on the **whole packet except for the age field.**

<u>Length :</u> This defines the **length of the whole packet in bytes.**

<u>Network Link Fields :</u>
<u>Netwouk Mask :</u>
This field defines the network mask.
<u>Attached Router :</u> This repeated fields defines the IP addresses of all attached routers.



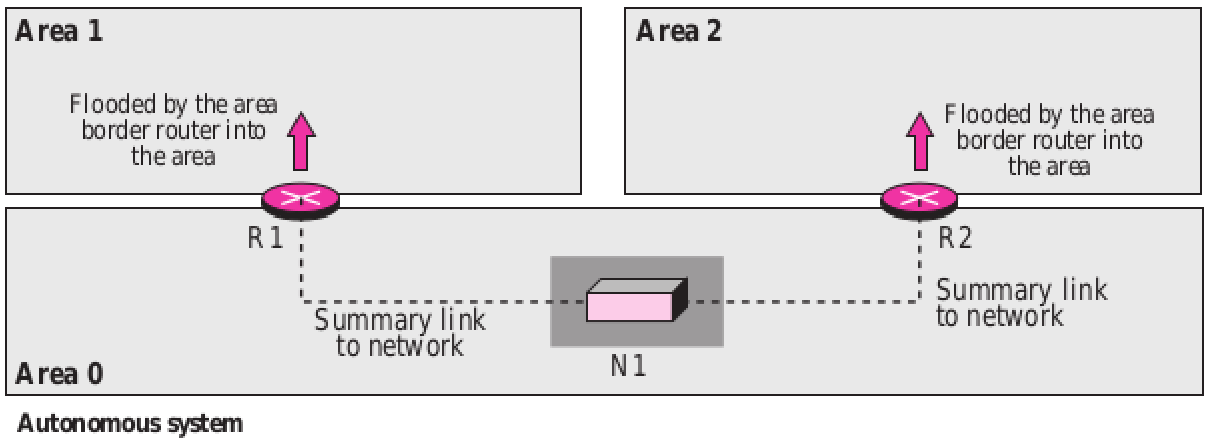


OSPF common header Type 4
Number of advertisements: 1
LSA general header Type 2
255.255.255.0
10.24.7.14
10.24.7.15
10.24.7.16

Summary Link to Network LSA :

Router Link and **Network Link Advertisement** flood the area with information about the **router links and Network** Links **inside an area.**But a router must also know about the **network outside its area;the area border routers can provide this information**.An **area boarder router** is active in more than **one area.**It receives **router link and network link advertisements** and,as well as see , creates a **routing table for each area.**

For example : Router R1 is an Area Boarder Router.It has 2 Routing tables,one for area 1 and one for area 0.R1 floods area 1 with information about how to reach a network located in area 0.In the same way,router R2 floods are 2 with information about how to reach the same network


Area 1
Flooded by the area border router into the area
Area 2
Flooded by the area border router into the area
R1
R2
Summary link to network
Summary link to network
N1
Area 0
Autonomous system

The **Summary link to network LSA** is used by the **Area Border Router** to announce the existence of other **network outside the area.**The **Summary link to network advertisement** is very simple.It consists of the **Network Mask** and the **Metric** for type of **Service.**Note that each advertisement annouces only one single network.If there is more than one network,a separate advertisement must be issued for each.The reader may ask why only the mask of the network is advertised.What about the network address itself?The IP address of the advertising router is announced in the header of the link state advertisement.From this information and the mask,one can deduce the network addresses.The format of this advertisement.

The fields of the **Summary link** to **Network LSA** as a follows.

Version	Type	Message Length		
Source Router IP Address				
Area Identification				
Checksum	Authentication type			
Authentication (32 bits)				
No of Advertisements				
Link Stage Age	Reserved	E	T	Link State Type
Link State ID				
Advertising router				
Link State Sequence number				
Link State checksum	Length			
Network Mask				
TOS	Metric			

<u>Common Header :</u>

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**
Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password**

<u>Link State Advertisement General Header</u>

<u>Link State Age :</u>

This field indicates the no of **seconds elapsed** since this **message** was first generated.Recall that this type of message goes from **router to router (flooding)**.When a router creates the **message** the value of this field is **0.**When each **successive router** forwards this message,it estimates the **transit time** and adds it to the **cumulative value** of this field.
E-Flag : If this **1-bit flag is set to 1**,it means that the area is a **Stub Area**.A **Stub Area** is an area that is connected to the backbone area by only one path

T-Flag : If this **1-bit is set to 1**,it means that the **router** can **handle multiple types of service.**

Link State Type : This field defines the LSA type.As we discussed before ,there are five different advertisement types.
1) Router Link
2) Network Link
3) Summary Link to Network
4) External Link

Link State ID :
The value of this field depends on the type of link.For **type 1(router link),**it is the I**P address of the router.**
For **type 2 (network link)**,it is the **IP address** of the **designated router**.For **type 3 (Summary link to network),**it is the address of the network **AS boundary router.**
For **type 4 (Summary Link to network).**It is the **IP address** of the **AS boundary router**.For **type 5 (external link)**,it is the address of the **external network.**

Advertising Router : This is the IP address ot the router advertising this message.

Link State Sequence Number : This is a sequence number assigned to each link state .

Link State Checksum: This is **not** the usual **chechttps://globaledge-pms.synergita.com/LogOnksum** instead,the value of this field is calculated using **Fletcher's checksum,**which is based on the **whole packet except for the age field.**

Length : This defines the **length of the whole packet in bytes.**

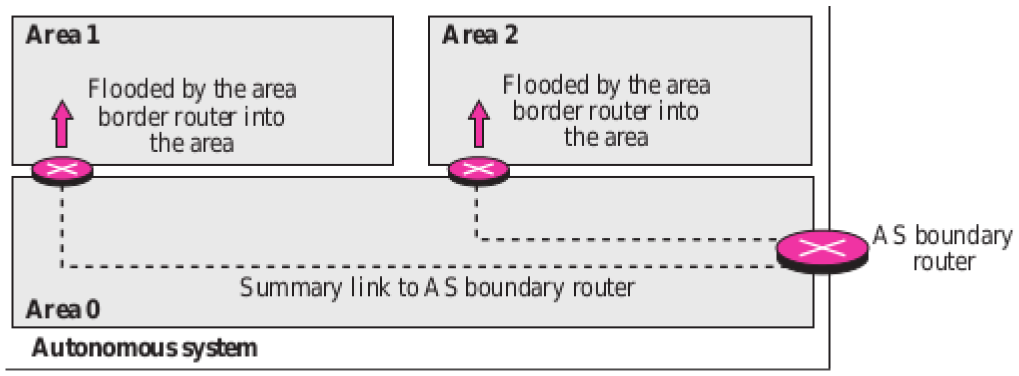
SUMMARY LINK TO NETWORK LSA

Network Mask : This field defines the **Network Mask TOS**.This field defines the **Type Of Service.**

Metric : This field defines the **metric** for the **Type Of Service** defined in the **TOS field,**

Summary Link to AS Boundary Router LSA

The previous Advertisemet lets every router know the cost to reach all of the network inside the autonomous system.But what about a network outside the autonomous system?If a router inside an area wants to send a packet **outside the autonomous system,** it should first know the route to an autonomous boundary router; the **summary link to AS boundary router** provides this information. The **area border routers** flood their areas with this information . This packet is used to announce the route to an AS boundary router. Its format is the same as the previous summary link. The packet just defines the network to which the AS boundary router is attached. If a message can reach the network, it can be picked up by the AS boundary router.


Area 1
Flooded by the area border router into the area
Area 2
Flooded by the area border router into the area
AS boundary router
Summary link to AS boundary router
Area 0
Autonomous system

The fields are the same as the fields in the **summary link to network advertisement message.**

Version	Type	Message Length		
Source Router IP Address				
Area Identification				
Checksum	Authentication type			
Authentication (32 bits)				
No of Advertisements				
Link Stage Age	Reserved	E	T	Link State Type
Link State ID				
Advertising router				
Link State Sequence number				
Link State checksum	Length			
All 0's				
TOS	Metric			

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**
Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password**

Link State Advertisement General Header

Link State Age :

This field indicates the no of **seconds elapsed** since this **message** was first generated.Recall that this type of message goes from **router to router (flooding)**.When a router creates the **message** the value of this field is **0.**When each **successive router** forwards this message,it estimates the **transit time** and adds it to the **cumulative value** of this field.
E-Flag : If this **1-bit flag is set to 1**,it means that the area is a **Stub Area**.A **Stub Area** is an area that is connected to the backbone area by only one path

T-Flag : If this **1-bit is set to 1**,it means that the **router** can **handle multiple types of service.**

Link State Type : This field defines the LSA type.As we discussed before ,there are five different advertisement types.
1) Router Link
2) Network Link
3) Summary Link to Network
4) External Link

Link State ID :
The value of this field depends on the type of link.For **type 1(router link),**it is the I**P address of the router.**
For **type 2 (network link)**,it is the **IP address** of the **designated router**.For **type 3 (Summary link to network),**it is the address of the network **AS boundary router.**
For **type 4 (Summary Link to network).**It is the **IP address** of the **AS boundary router**.For **type 5 (external link)**,it is the address of the **external network.**

Advertising Router : This is the IP address ot the router advertising this message.

Link State Sequence Number : This is a sequence number assigned to each link state .

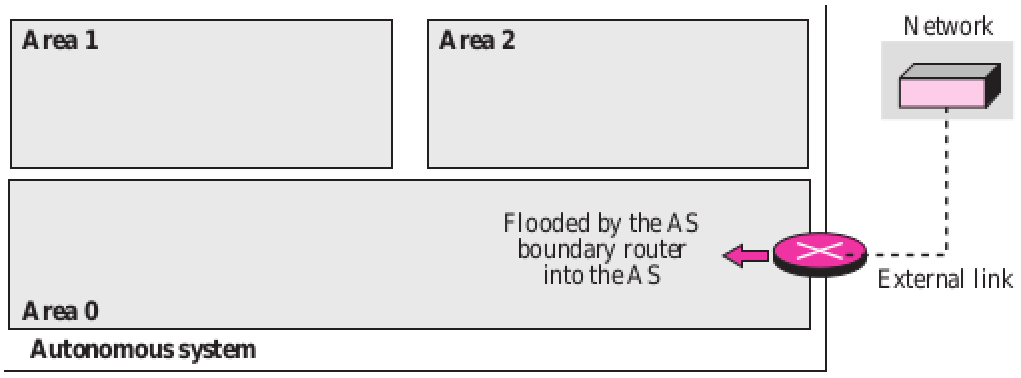
Link State Checksum : This is **not** the usual instead,the value of this field is calculated using **Fletcher's checksum,**which is based on the **whole packet except for the age field.**

Length : This defines the **length of the whole packet in bytes.**

External Link LSA

Although the previous advertisement lets each router know the route to an AS boundary router,this information is not enough.A **router inside an autonomous system wants to which network are available outside the autonomous system;**the external links advertisements provides this information.The **AS boundary router** floods the **AS system** with the cost of each **network** outside the **autonomous system** using a **routing table** created by an interdomain **routing protocol.** Each advertisement announces one single network. If there is more than one network, separate announcements are made. This is used to announce all the networks outside the

AS. The format of the LSA is similar to the **summary link to the AS boundary router LSA**, with the addition of two fields. The **AS boundary router** may define a **forwarding router that can provide a better route to the destination.** The packet also can include an **external route tag,** used by **other protocols, but not by OSPF.** The format of the packet



Version	Type	Message Length		
Source Router IP Address				
Area Identification				
Checksum		Authentication type		
Authentication (32 bits)				
No of Advertisements				
Link Stage Age	Reserved	E	T	Link State Type
Link State ID				
Advertising router				
Link State Sequence number				
Link State checksum		Length		
Network Mask				
TOS	Metric			
Forwarding Address				
External route tag				

Other Packets: Now we discussed four other packet types.They are not used as LSA'S,but are essential to the operations of OSPF.

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**
Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password**

Link State Advertisement General Header
Link State Age :
This field indicates the no of **seconds elapsed** since this **message** was first generated.Recall that this type of message goes from **router to router (flooding)**.When a router creates the **message** the value of this field is **0.**When each **successive router** forwards this message,it estimates the **transit time** and adds it to the **cumulative value** of this field.
E-Flag : If this **1-bit flag is set to 1**,it means that the area is a **Stub Area**.A **Stub Area** is an area that is connected to the backbone area by only one path

T-Flag : If this **1-bit is set to 1**,it means that the **router** can **handle multiple types of service.**

Link State Type : This field defines the LSA type.As we discussed before ,there are five different advertisement types.
1) Router Link
2) Network Link
3) Summary Link to Network
4) External Link

Link State ID :
The value of this field depends on the type of link.For **type 1(router link),**it is the I**P address of the router.**
For **type 2 (network link)**,it is the **IP address** of the **designated router**.For **type 3 (Summary link to network),**it is the address of the network **AS boundary router.**
For **type 4 (Summary Link to network).**It is the **IP address** of the **AS boundary router**.For **type 5 (external link)**,it is the address of the **external network.**

<u>Advertising Router :</u> This is the IP address ot the router advertising this message.

<u>Link State Sequence Number :</u> This is a sequence number assigned to each link state .

<u>Link State Checksum :</u>This is **not** the usual instead,the value of this field is calculated using **Fletcher's checksum,**which is based on the **whole packet except for the age field.**

<u>Length :</u> This defines the **length of the whole packet in bytes.**

Hello Packet :

OSPF uses the Hello messages to create **neighborhood relationships** and to test the reach **ability of neighbors**.This is the first step in **link state Routing.**Before a router can flood all of the other routers with information about its neighbors,it must know if they are alive,and it must know if they are reachable.

Hello Packet :

Version	**Type :1**	Message Length		
Source Router IP Address				
Area Identification				
Checksum	Authentication type			
Authentication (32 bits)				
Network Mask				
Hello interval	All 0's	E	T	Priority
Dead interval				
Designated router IP router				
Backup designated router IP address				
Neighbour IP address				

<u>Common Header :</u>

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**
Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password**

The fields of Hello packet

Network mask : This 32-bit field defines the network mask of the network over which the hello message is sent.

Hello interval : This 16-bit field defines the number of seconds between hello messages. E flag. This is a 1-bit flag. When it is set, it means that the area is a stub area.

T flag : This is a 1-bit flag. When it is set, it means that the router supports multiple
metrics.
Priority : This field defines the priority of the router. The priority determines the selection of the designated router. After all neighbors declare their priorities, the router with the highest priority is chosen as the designated router. The one with the second highest priority is chosen as the backup designated router. If the value of this field is 0, it means that the router never wants to be a designated or a backup designated router.

Dead interval : This 32-bit field defines the number of seconds that must pass before a router assumes that a neighbor is dead.

Designated router IP address : This 32-bit field is the IP address of the designated router for the network over which the message is sent.

Backup designated router IP address: This 32-bit field is the IP address of the backup designated router for the network over which the message is sent.

Neighbor IP address: This is a repeated 32-bit field that defines the routers that have agreed to be the neighbors of the sending router. In other words, it is a current list of all the neighbors from which the sending router has received the hello message.

Database Description Message

When a router is connected to the system for the first time or after a failure,it needs the complete link state database immediately.It cannot wait for all link state update packets to come from every other router before making its own database and calculating its routing table.Therefore,after a router is connected to the system,it sends hello packets to greet its neighbors.If this is the first time that the neighbors hear from the router,they send a database description message.The database description packet does not contain complete database information;it only gives an outline,the title of each line in the database.The newly connected router examines the outline,and finds out which lines of information about that particular link.When two routers want to exchange database description packets,one of them takes the role of master and the other the role of slave database can be divided into several messages.The formate of the database description packet.

Version	**Type : 2**	Message Length					
Source Router IP Address							
Area Identification							
Checksum	Authentication type						
Authentication (32 bits)							
All 0's	All 0's	E	B	All 0's	I	M	M S
Message Sequence Number							
Link Stage Age	Reserved	E	T	Link State Type			
Link State ID							
Advertising router							
Link State Sequence number							
Link State checksum	Length						

<u>Common Header :</u>

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**

Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password**

The Link State Database Description

E flag : This 1-bit flag is set to 1 if the advertising router is an autonomous boundary router (E stands for external).
B flag : This 1-bit flag is set to 1 if the advertising router is an area border router.
I flag : This 1-bit field, the initialization flag, is set to 1 if the message is the first message.
M flag : This 1-bit field, the more flag, is set to 1 if this is not the last message.
M/S flag : This 1-bit field, the master/slave bit, indicates the origin of the packet: master (M/S = 1) or slave (M/S = 0).
Message sequence number : This 32-bit field contains the sequence number of the message. It is used to match a request with the response.

Link State Advertisement General Header

Link State Age :
This field indicates the no of **seconds elapsed** since this **message** was first generated.Recall that this type of message goes from **router to router (flooding)**.When a router creates the **message** the value of this field is **0.**When each **successive router** forwards this message,it estimates the **transit time** and adds it to the **cumulative value** of this field.
E-Flag : If this **1-bit flag is set to 1**,it means that the area is a **Stub Area**.A **Stub Area** is an area that is connected to the backbone area by only one path

T-Flag : If this **1-bit is set to 1**,it means that the **router** can **handle multiple types of service.**

Link State Type : This field defines the LSA type.As we discussed before ,there are five different advertisement types.
1) Router Link
2) Network Link
3) Summary Link to Network
4) External Link

Link State ID :
The value of this field depends on the type of link.For **type 1(router link),**it is the I**P address of the router.**
For **type 2 (network link)**,it is the **IP address** of the **designated router**.For **type 3 (Summary link to network),**it is the address of the network **AS boundary router.**
For **type 4 (Summary Link to network).**It is the **IP address** of the **AS boundary router**.For **type 5 (external link)**,it is the address of the **external network.**

Advertising Router : This is the IP address ot the router advertising this message.

<u>Link State Sequence Number :</u> This is a sequence number assigned to each link state .

<u>Link State Checksum :</u> This is **not** the usual instead,the value of this field is calculated using **Fletcher's checksum,**which is based on the **whole packet except for the age field.**

<u>Length :</u> This defines the **length of the whole packet in bytes.**

Link State Request Packet

The format of the **link state request packet** . This is a packet that is sent by a router that needs information about a specific route or routes. It is answered with a link state update packet. It can be used by a newly connected router to request more information about some routes after receiving the database description packet. The three fields here are part of the LSA header, which has already been discussed. Each set of the three fields is a request for one single LSA. The set is repeated if more than one advertisement is desired.

Version	**Type : 3**	Message Length
Source Router IP Address		
Area Identification		
Checksum		Authentication type
Authentication (32 bits)		
Link State type		
Link State ID		
Advertising Router		

<u>Common Header :</u>

<u>Hi Friends,</u>

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**
Authentication : This **64 bit** field is the acual value of the **authentication data**.In the future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password.**

Link State Knowledgement Packet

Version	**Type : 5**	Message Length		
Source Router IP Address				
Area Identification				
Checksum		Authentication type		
Authentication (32 bits)				
No of Advertisements				
Link Stage Age	Reserved	E	T	Link State Type
Link State ID				
Advertising router				
Link State Sequence number				
Link State checksum		Length		

<u>Common Header :</u>

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 take place.
Area Identification : This 32-bit field defines the area within which the routing take 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**.At this time, **2 types** of **authentication** are defined.**O** for none and **1** for **password**
Authentication : This **64 bit** field is the acual value of the **authentication data**.In the

future,whenmore **authentication** types are defined,this field will contain the result of the **authentication calculation.**If the type is **1**,this field carries an **eight-character password.**

<u>Link State Advertisement General Header</u>

<u>Link State Age :</u>

This field indicates the no of **seconds elapsed** since this **message** was first generated.Recall that this type of message goes from **router to router (flooding)**.When a router creates the **message** the value of this field is **0.**When each **successive router** forwards this message,it estimates the **transit time** and adds it to the **cumulative value** of this field.

<u>E-Flag :</u> If this **1-bit flag is set to 1**,it means that the area is a **Stub Area**.A **Stub Area** is an area that is connected to the backbone area by only one path

<u>T-Flag :</u> If this **1-bit is set to 1**,it means that the **router** can **handle multiple types of service.**

<u>Link State Type :</u> This field defines the LSA type.As we discussed before ,there are five different advertisement types.

1) Router Link

2) Network Link

3) Summary Link to Network

4) External Link

<u>Link State ID :</u>

The value of this field depends on the type of link.For **type 1(router link),**it is the I**P address of the router.**

For **type 2 (network link)**,it is the **IP address** of the **designated router**.For **type 3 (Summary link to network),**it is the address of the network **AS boundary router.**

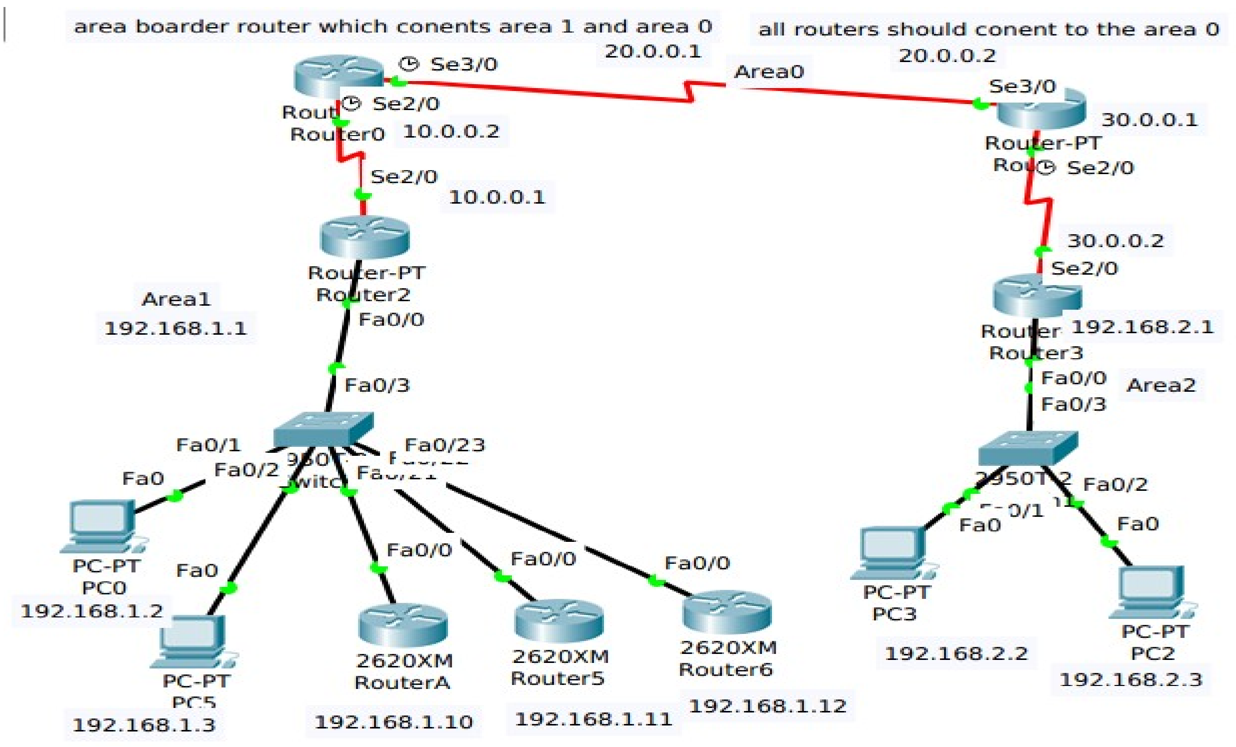
For **type 4 (Summary Link to network).**It is the **IP address** of the **AS boundary router**.For **type 5 (external link)**,it is the address of the **external network.**

<u>Advertising Router :</u> This is the IP address ot the router advertising this message.

<u>Link State Sequence Number :</u> This is a sequence number assigned to each link state .

<u>Link State Checksum :</u> This is **not** the usual instead,the value of this field is calculated using **Fletcher's checksum,**which is based on the **whole packet except for the age field.**

<u>Length :</u> This defines the **length of the whole packet in bytes.**



<u>OSPF Configuration on Router0</u>

```
router#conf t
router(config)#hostname r0
r0(config)#int se2/0
r0(config-if)#ip address 10.0.0.2 255.0.0.0
r0(config-if)#exit
r0(config)#int se3/0
r0(config-if)#ip address 20.0.0.2 255.0.0.0
r0(config-if)#exit

r0(config)#router ospf 1
r0(config-router)#network 10.0.0.0  0.255.255.255 area 0
r0(config-router)#network 20.0.0.0  0.255.255.255 area 0
```

<u>Configuration on Router1</u>

```
router#conf t
router(config)#hostname r1
r1(config)#int se2/0
r1(config-if)#ip address 30.0.0.1 255.0.0.0
r1(config-if)#exit
r1(config)#int se3/0
r1(config-if)#ip address 20.0.0.1 255.0.0.0
r1(config-if)#exit
```

r1(config)#**router ospf 1**
r1(config-router)#**network 30.0.0.0 0.255.255.255 area 0**
r1(config-router)#**network 20.0.0.0 0.255.255.255 area 0**

Configuration on Router2

router#**conf t**
router(config)#**hostname r2**
r2(config)#**int se2/0**
r2(config-if)#**ip address 30.0.0.2 255.0.0.0**
r2(config-if)#**exit**
r2(config)#**int fa0/0**
r2(config-if)#**ip address 192.168.1.1 255.255.255.0**
r2(config-if)#**exit**

r2(config)#**router ospf 1**
r2(config-router)#**network 10.0.0.0 0.255.255.255 area 0**
r2(config-router)#**network 192.168.1.0 0.0.0.255 area 1**

Configuration on Router3

router#**conf t**
router(config)#**hostname r3**
r3(config)#**int se2/0**
r3(config-if)#**ip address 10.0.0.1 255.0.0.0**
r3(config-if)#**exit**
r3(config)#**int fa0/0**
r3(config-if)#**ip address 192.168.2.1 255.255.255.0**
r3(config-if)#**exit**

r3(config)#**router ospf 1**
r3(config-router)#**network 30.0.0.0 0.255.255.255 area 0**
r3(config-router)#**network 192.168.2.0 0.0.0.255 area 1**

Hi Friends,

Redistributing Routes of OSPF

Redistributing routes into **OSPF** from other **Routing Protocols** or from static will cause these routes to become **OSPF external routes**.To redistribute routes into OSPF.

Redistribute OSPF into other Protocol use of a valid matrix

Whenever you Redistribute OSPF into other protocols,you have to respect the rules of those protocols.In particular,the metric applied should match the metric used by the protocol.
For ex: The **RIP metric** is a hop count ranging between **1 and 16**,where **1 indicates that a network is one** hop away and **16 indicates** that the **network is unreachable.**

Multiple Redistribution

Mutual Redistribution between protocols should be done very carefully and in a controlled manner.Incorrect configuration could lead to potential looping of routing infomation.A rule of thumb.
Redistributing Routes in OSPF

redistribute protocol [process id] [metric value] [metric-type value] [route-map map-tag] [subnets]

The protocol and process-id are the protocol that we are injecting into **OSPF a**nd its **process-id** if exits.The **metric is the cost** we are assigning to the **external route.**If **no metric** is specified**,OSPF** puts a default value of **20** when **redistributing routes** from **all protocols except BGP routes**,when get a **metric of 1.**The metric type is discussed in the next paragraph.
The **route-map** is a method used to control the **redistribution** of routes between **routing domains.**The format of a route map is.
The **route-map** is a method used to control the **redistribution of routes** between **routing domains.** The format of a route map is

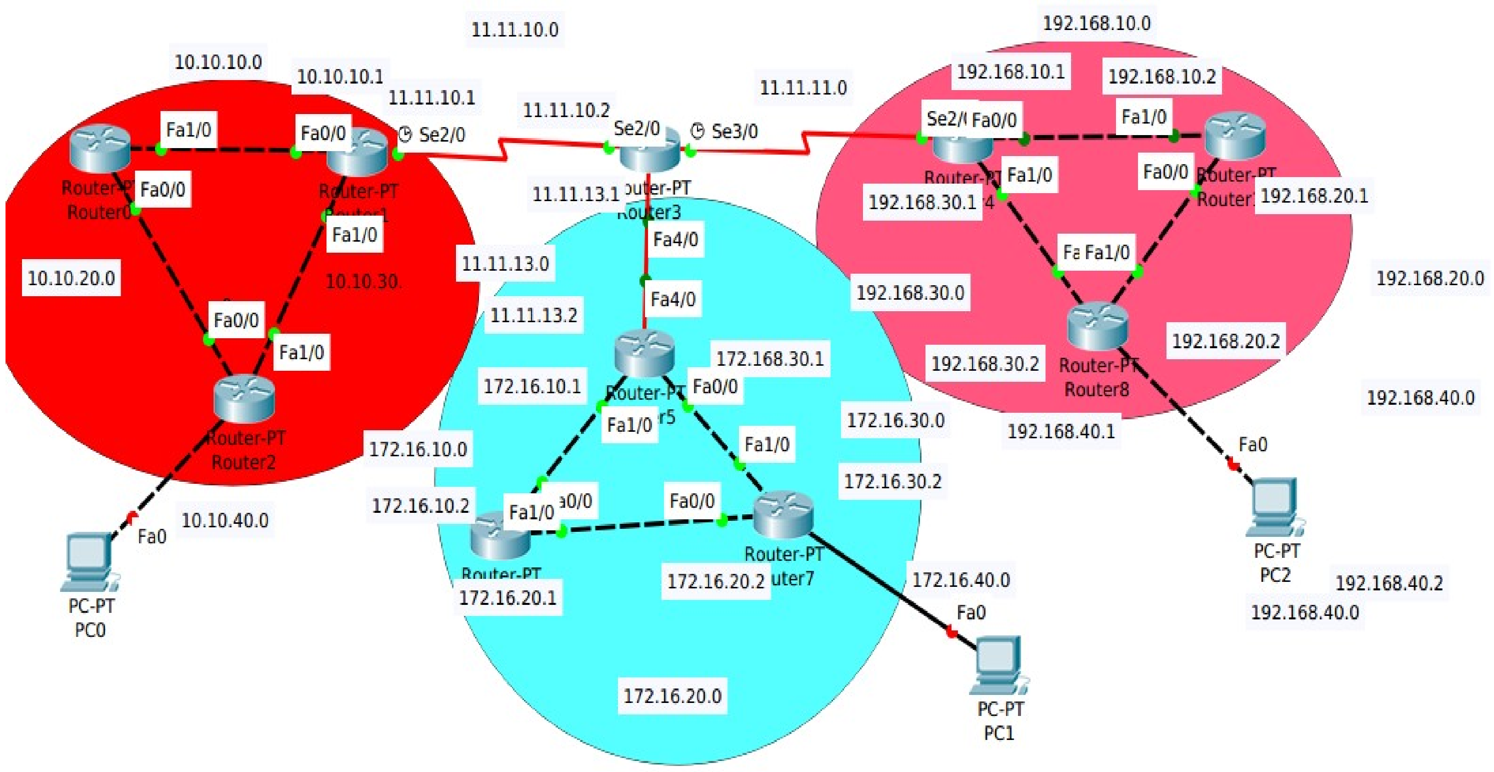
route-map map-tag [Epermit | deny | [sequence-number]

E1 vs E2 External Routes :

External routes fall under **2 categories,external type 1 and external type 2**.The **difference the 2** is in the way the **cost (metric)** of the **route is being calculated.**The cost of a type 2 route is **always the external cost**,irrespective of the **interior cost to reach that route**.A **type 1 cost** is the addition of the **external cost** and the **internal cost** used to reach **that route.**A **type 1 route** is always preffered over a type 2 route for the same destination.

Redistributing OSPF into othe Protocol (VLSM)

Another issue to consider is VLSM (variable length subnet guied).OSPF can carry multiple subnet information for the same major net,but other protocols such as RIP and IGRP
It have 2 different subnet mask it checks in routing table.



OSPF Redistribution Configuration

For router0, router1, router2 we configure the rip protocol

Router 1

```
router#conf t
router(config)#hostname r1
r1#(config)#router rip
r1(config-router)#version 2
r1(config-router)#network 11.11.10.0
r1(config-router)#network 10.10.10.0
router(config-router)#network 10.10.30.0
```

Router0

```
router#conf t
router(config)#hostname r0
r0#(config)#router rip
r0(config-router)#version 2
r0(config-router)#network 10.10.10.0
r0r(config-router)#network 10.10.20.0
```

Router2

```
router#conf t
router(config)#hostname r2
```

```
r2#(config)#router rip
r2(config-router)#version 2
r2r(config-router)#network 10.10.40.0
r2r(config-router)#network 10.10.20.0
r2r(config-router)#network 10.10.30.0
```

For router4,router8,router10 configure eigrp

Router4

```
router#conf t
router(config)#hostname r4
r4(config)#router eigrp 10
r4(config-router)#network 192.168.10.0
r4(config-router)#network 192.168.30.0
r4(config-router)#network 11.11.11.0
r4(config-router)#do wr
```

Router8bgcolor="0xFFFFFF">

```
router#conf t
router(config)#hostname r8
r8(config)#router eigrp 10
r8(config-router)#network 192.168.30.0
r8(config-router)#network 192.168.20.0
r8(config-router)#network 192.168.40.0
r4(config-router)#do wr
```

Router10

```
router#conf t
router(config)#hostname r10
r10(config)#router eigrp 10
r10(config-router)#network 192.168.10.0
r10(config-router)#network 192.168.20.0
r4(config-router)#do wr
```

For the router5,router7,and router6 we configure the OSPF

Router5

```
router#conf t
router(config)#hostname r5
r5(config)#router ospf 1
r5(config-router)#network 11.11.13.0  0.0.0.3 area 0
r5(config-router)#network 172.16.10.0  0.0.0.3 area 0
r5(config-router)#network 172.16.30.0  0.0.0.3 area 0
r5(config-router)#do wr
```

Router7

```
router#conf t
router(config)#hostname r7network 172.16.30.0  0.0.0.3 area 0
r7(config)#router ospf 1
r7(config-router)#network 172.16.40.0  0.0.0.255 area 0
r7(config-router)#network 172.16.30.0  0.0.0.3 area 0
r7(config-router)#network 172.16.20.0  0.0.0.3 area 0
r7(config-router)#do wr
```

Router6

```
router#conf t
router(config)#hostname r6
r6(config)#router ospf 1
r6(config-router)#network 172.16.10.0  0.0.0.3 area 0
r6(config-router)#network 172.16.20.0  0.0.0.3 area 0
r6(config-router)#do wr
```