Chapter 5
Network/Internet Layer Protocols and Addressing

IP Address:

Each computer in a TCP/IP network must be given a unique identifier, or IP address. This address, which operates at Layer 3, allows one computer to locate another computer on a network. All computers also have a unique physical address, which is known as a MAC address. These are assigned by the manufacturer of the NIC. MAC addresses operate at Layer 2 of the OSI model.

An IP address (IPv4) is a 32-bit sequence of ones and zeros. To make the IP address easier to work with, it is usually written as four decimal numbers separated by periods. For example, an IP address of one computer is 192.168.1.2. Another computer might have the address 128.10.2.1. This is called the dotted decimal format. Each part of the address is called an octet because it is made up of eight binary digits. For example, the IP address 192.168.1.8 would be 11000000.10101000.00000001.00001000 in binary notation. The dotted decimal notation is an easier method to understand than the binary ones and zeros method. This dotted decimal notation also prevents a large number of transposition errors that would result if only the binary numbers were used.

Ipv4 Header:

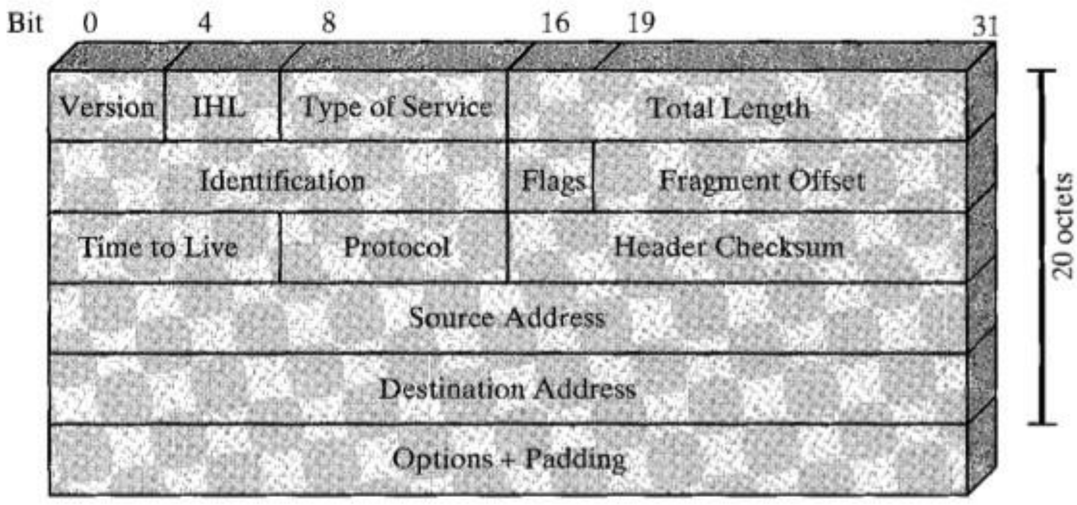


Fig: IPV4 Header

- **Version (4 bits):** Indicates the version number, to allow evolution of the protocol.
- **Internet Header Length (IHL 4 bits):** Length of header in 32 bit words. The minimum value is five for a minimum header length of 20 octets.
- **Type-of-Service:** The Type-of-Service field contains an 8-bit binary value that is used to determine the priority of each packet. This value enables a Quality-of-Service (QoS) mechanism to be applied to high priority packets, such as those carrying telephony voice data. The router processing the packets can be configured to decide which packet it is to forward first base on the Type-of-Service value.
- **Total length:** Total datagram length, in octets.
- **Identifier (16 bits):** A sequence number that, together with the source address, destination address, and user protocol, is intended to uniquely identify a datagram. Thus, the identifier should be unique for the datagram's source address, destination address, and user protocol for the time during which the datagram will remain in the internet.
- **Fragment Offset:** A router may have to fragment a packet when forwarding it from one medium to another medium that has a smaller MTU. When fragmentation occurs, the IPv4 packet uses the Fragment Offset field and the MF flag in the IP header to reconstruct the packet when it arrives at the destination host. The fragment offset field identifies the order in which to place the packet fragment in the reconstruction.
- **Flags(3 bits):** Only two of the bits are currently defined: MF(More Fragments) and DF(Don't Fragment):
- **More Fragments flag (MF):** The More Fragments (MF) flag is a single bit in the Flag field used with the Fragment Offset for the fragmentation and reconstruction of packets. The More Fragments flag bit is set; it means that it is not the last fragment of a packet. When a receiving host sees a packet arrive with the MF = 1, it examines the Fragment Offset to see where this fragment is to be placed in the reconstructed packet. When a receiving host receives a frame with the MF = 0 and a non-zero value in the Fragment offset, it places that fragment as the last part of the reconstructed packet. An unfragmented packet has all zero fragmentation information (MF = 0, fragment offset =0).
- **Don't Fragment flag (DF):** The Don't Fragment (DF) flag is a single bit in the Flag field that indicates that fragmentation of the packet is not allowed. If the Don't Fragment flag bit is set, then fragmentation of this packet is NOT permitted. If a router needs to fragment a packet to allow it to be passed downward to the Data Link layer but the DF bit is set to 1, then the router will discard this packet.
- **IP Destination Address**: The IP Destination Address field contains a 32-bit binary value that represents the packet destination Network layer host address.
- **IP Source Address**: The IP Source Address field contains a 32-bit binary value that represents the packet source Network layer host address.
- **Time-to-Live:** The Time-to-Live (TTL) is an 8-bit binary value that indicates the remaining "life" of the packet. The TTL value is decreased by at least one each time the packet is processed by a router (that is, each hop). When the value becomes zero, the router discards or drops the packet and it is removed from the network data flow. This mechanism prevents packets that cannot reach their destination from being forwarded indefinitely between routers in a routing loop. If routing loops were permitted to continue, the network would become congested with data packets that will never reach their destination. Decrementing the TTL value at each hop ensures that it eventually becomes zero and that the packet with the expired TTL field will be dropped.

- **Protocol:** This 8-bit binary value indicates the data payload type that the packet is carrying. The Protocol field enables the Network layer to pass the data to the appropriate upper-layer protocol.
  Example values are:
  01 ICMP
  06 TCP
  17 UDP
- **Header checksum (16 bits):** An error-detecting code applied to the header only. Because some header fields may change during transit (e.g., time to live, segmentation-related fields), this is reverified and recomputed at each router. The checksum field is the 16-bit one's complement addition of all 16-bit words in the header. For purposes of computation, the checksum field is itself initialized to a value of zero.
- **Options (variable):** Encodes the options requested by the sending user.
- **Padding (variable):** Used to ensure that the datagram header is a multiple of 32 bits.
- **Data (variable):** The data field must be an integer multiple of 8 bits. The maximum length of the datagram (data field plus header) is 65,535 octets.

IP addresses are divided into class:

IP Address Class	First Octet Address Range	Used for
Class A	0-127	Unicast (Very large Network)
Class B	128-191	Unicast (Medium to Large Network)
Class C	192-223	Unicast (Small Network)
Class D	224-239	Multicast
Class E	240-255	Reserved

Class A Blocks
A class A address block was designed to support extremely large networks with more than 16 million host addresses. Class A IPv4 addresses used a fixed /8 prefix with the first octet to indicate the network address. The remaining three octets were used for host addresses.

The first bit of a Class A address is always 0. With that first bit a 0, the lowest number that can be represented is 00000000, decimal 0. The highest number that can be represented is 01111111, decimal 127. The numbers 0 and 127 are reserved and cannot be used as network addresses. Any address that starts with a value between 1 and 126 in the first octet is a Class A address.

No. of Class A Network: 2^7
No. of Usable Host address per Network: 2^{24}-2 (Minus 2 because 2 addresses are reserved for network and broadcast address)

Class B Blocks
Class B address space was designed to support the needs of moderate to large size networks with more than 65,000 hosts. A class B IP address used the two high-order octets to indicate the network address. The other two octets specified host addresses. As with class A, address space for the remaining address classes needed to be reserved.

The first two bits of the first octet of a Class B address are always 10. The remaining six bits may be populated with either 1s or 0s. Therefore, the lowest number that can be represented with a Class B address is 10000000, decimal 128. The highest number that can be represented is 10111111, decimal 191. Any address that starts with a value in the range of 128 to 191 in the first octet is a Class B address.

No of Class B Network: 2^{14}

No. of Usable Host address per Network: $2^{16}-2$

Class C Blocks:
The class C address space was the most commonly available of the historic address classes. This address space was intended to provide addresses for small networks with a maximum of 254 hosts. Class C address blocks used a /24 prefix. This meant that a class C network used only the last octet as host addresses with the three high-order octets used to indicate the network address.

A Class C address begins with binary 110. Therefore, the lowest number that can be represented is 11000000, decimal 192. The highest number that can be represented is 11011111, decimal 223. If an address contains a number in the range of 192 to 223 in the first octet, it is a Class C address.

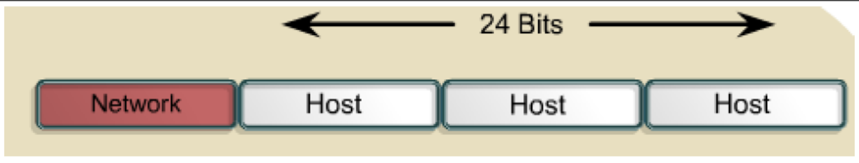
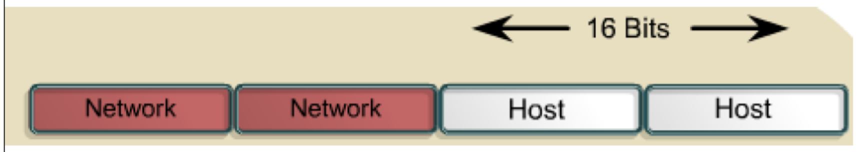
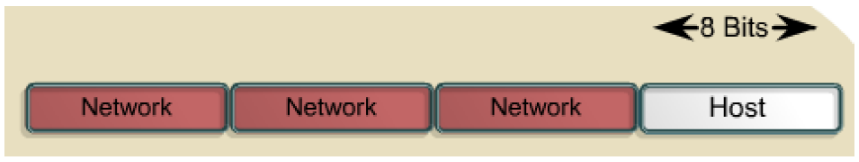
No of Class C Network: 2^{21}
No. of Usable Host address per Network: $2^{8}-2$

Class D Blocks:
The Class D address class was created to enable multicasting in an IP address. A multicast address is a unique network address that directs packets with that destination address to predefined groups of IP addresses. Therefore, a single station can simultaneously transmit a single stream of data to multiple recipients.

The Class D address space, much like the other address spaces, is mathematically constrained. The first four bits of a Class D address must be 1110. Therefore, the first octet range for Class D addresses is 11100000 to 11101111, or 224 to 239. An IP address that starts with a value in the range of 224 to 239 in the first octet is a Class D address.

Class E Block:
A Class E address has been defined. However, the Internet Engineering Task Force (IETF) reserves these addresses for its own research. Therefore, no Class E addresses have been released for use in the Internet. The first four bits of a Class E address are always set to 1s. Therefore, the first octet range for Class E addresses is 11110000 to 11111111, or 240 to 255.

Every IP address also has two parts. The first part identifies the network (Network ID) where the system is connected and the second part identifies the system (Host ID).

Class A	
Class B	
Class C	

Within the address range of each IPv4 network, we have three types of addresses:

- Network address - The address by which we refer to the network
- Broadcast address - A special address used to send data to all hosts in the network
- Host addresses - The addresses assigned to the end devices in the network

Special Ipv4 addresses:

- **Default Route:** we represent the IPv4 default route as 0.0.0.0. The default route is used as a "catch all" route when a more specific route is not available. The use of this address also reserves all addresses in the 0.0.0.0 - 0.255.255.255 (0.0.0.0 /8) address block.
- **Network and Broadcast Addresses:** As explained earlier, within each network the first and last addresses cannot be assigned to hosts. These are the network address and the broadcast address, respectively.
- **Loopback:** One such reserved address is the IPv4 loopback address 127.0.0.1. The loopback is a special address that hosts use to direct traffic to them. Although only the single 127.0.0.1 address is used, addresses 127.0.0.0 to 127.255.255.255 are reserved. Any address within this block will loop back within the local host. No address within this block should ever appear on any network.
- **Link-Local Addresses:** IPv4 addresses in the address block 169.254.0.0 to 169.254.255.255 (169.254.0.0 /16) are designated as link-local addresses. These addresses can be automatically assigned to the local host by the operating system in environments where no IP configuration is available. These might be used in a small peer-to-peer network or for a host that could not automatically obtain an address from a Dynamic Host Configuration Protocol (DHCP) server.
- **TEST-NET Addresses:** The address block 192.0.2.0 to 192.0.2.255 (192.0.2.0 /24) is set aside for teaching and learning purposes. These addresses can be used in documentation and network examples.
- **Network Prefixes:** An important question is: How do we know how many bits represent the network portion and how many bits represent the host portion? When we

express an IPv4 network address, we add a prefix length to the network address. The prefix length is the number of bits in the address that gives us the network portion. For example, in 172.16.4.0 /24, the /24 is the prefix length - it tells us that the first 24 bits are the network address. This leaves the remaining 8 bits, the last octet, as the host portion.

Private and Public IP addresses:

- Public IP addresses: Public IP addresses are assigned by the InterNIC (Internet's Network Information Centre) and consists of class based network Ids or blocks of CIDR based addresses (called CIDR blocks) that are globally rout-able to the Internet and are unique.
- Private IP address: An address that is used for internal networks. These addresses are not rout-able to the Internet.
  The private address blocks are:
  10.0.0.0 to 10.255.255.255 (10.0.0.0 /8)
  172.16.0.0 to 172.31.255.255 (172.16.0.0 /12)
  192.168.0.0 to 192.168.255.255 (192.168.0.0 /16)

Subnet Mask:
To define the network and host portions of an address, the devices use a separate 32-bit pattern called a subnet mask. We express the subnet mask in the same dotted decimal format as the IPv4 address. The subnet mask is created by placing a binary 1 in each bit position that represents the network portion and placing a binary 0 in each bit position that represents the host portion.
The prefix and the subnet mask are different ways of representing the same thing - the network portion of an address.

Default Subnet Mask:
Class A: 255.0.0.0
Class B: 255.255.0.0
Class C: 255.255.255.0

CIDR:
A routing system used by routers and gateways on the backbone of the Internet for routing packets. CIDR replaces the old class method of allocating 8, 16, or 24 bits to the network ID, and instead allows any number of contiguous bits in the IP address to be allocated as the network ID. For example, if a company needs a few thousand IP addresses for its network, it can allocate 11 or 12 bits of the address for the network ID instead of 8 bits for a class C (which wouldn’t work because you would need to use several class C networks) or 16 bits for class B (which is wasteful).

How It Works?
CIDR assigns a numerical prefix to each IP address. For example, a typical destination IP address using CIDR might be 177.67.5.44/13. The prefix 13 indicates that the first 13 bits of the IP address identify the network, while the remaining 32 - 13 = 19 bits identify the host. The prefix helps to identify the Internet destination gateway or group of gateways to which the packet will be forwarded. Prefixes vary in size, with longer prefixes indicating more specific destinations. Routers use the longest possible prefix in their routing tables when determining how to forward each packet. CIDR enables packets to be sent to groups of networks instead of to individual networks, which considerably simplifies the

complex routing tables of the Internet’s backbone routers.

How to Create Subnets?
To create subnetworks, you take bits from the host portion of the IP address and reserve them to define the subnet address.

How many bits to borrow?

I. No of subnetwork = 2^{BB}
II. No. of usable hosts per subnetwork=2^{BR}-2

TB=BR + BB
TB=Total bits in host portion
BB=Bits borrowed
BR=Bits Remaining

Subnetting Class C Addresses
There are many different ways to subnet a network. The right way is the way that works best for you. In a Class C address, only 8 bits are available for defining the hosts. Remember that subnet bits start at the left and go to the right, without skipping bits. This means that the only Class C subnet masks can be the following:

Binary	Decimal	CIDR
00000000	0	/24
10000000	128	/25
11000000	192	/26
11100000	224	/27
11110000	240	/28
11111000	248	/29
11111100	252	/30

We can’t use a /31 or /32 because we have to have at least 2 host bits for assigning IP addresses to hosts.

All you need to do is answer five simple questions:
How many subnets does the chosen subnet mask produce?
How many valid hosts per subnet are available?
What are the valid subnets?

I. What’s the broadcast address of each subnet?
II. What are the valid hosts in each subnet?

Subnetting Class C Address: 192.168.10.0/26
255.255.255.192 (/26)
In this second example, we’re going to subnet the network address 192.168.10.0 using the subnet mask 255.255.255.192.
192.168.10.0 = Network addresses
255.255.255.192 = Subnet mask
Now, let’s answer the big five:
How many subnets? Since 192 is 2 bits on (11000000), the answer would be 2^2 = 4 subnets.

How many hosts per subnet? We have 6 host bits off (11000000), so the equation would be $2^6 – 2 = 62$ hosts.

What are the valid subnets? 256 – 192 = 64. Remember, we start at zero and count in our block size, so our subnets are 0, 64, 128, and 192. (Magic Number=256-Subnet Mask)

What’s the broadcast address for each subnet? The number right before the value of the next subnet is all host bits turned on and equals the broadcast address. For the zero subnet, the next subnet is 64, so the broadcast address for the zero subnet is 63.

What are the valid hosts? These are the numbers between the subnet and broadcast address. The easiest way to find the hosts is to write out the subnet address and the broadcast address. This way, the valid hosts are obvious.

The following table shows the 0, 64, 128, and 192 subnets, the valid host ranges of each, and the broadcast address of each subnet:

The Subnet	0	64	128	192
The Broadcast Address	63	127	191	255
Usable Host Address	1-62	65-126	129-190	193-254

Subnetting Class B Address: 172.16.0.0/17
255.255.128.0 (/17)
172.16.0.0 = Network address
255.255.128.0 = Subnet mask

Subnets? $2^1 = 2$ (same as Class C).
Hosts? $2^{15} – 2 = 32,766$ (7 bits in the third octet, and 8 in the fourth).
Valid subnets? 256 – 128 = 128. 0, 128. Remember that subnetting is performed in the third octet, so the subnet numbers are really 0.0 and 128.0, as shown in the next table.

These are the exact numbers we used with Class C; we use them in the third octet and add a 0 in the fourth octet for the network address.

Broadcast address for each subnet?

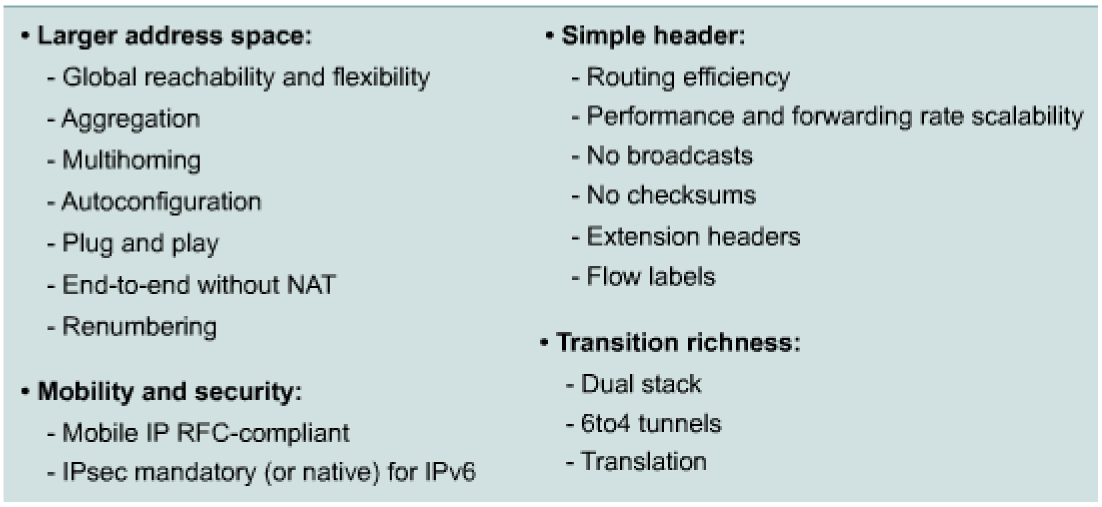
Valid hosts?

The following table shows the two subnets available, the valid host range, and the broadcast address of each:

Subnet	172.16.0.0	172.16.128.0
Broadcast	172.16.127.255	172.16.255.255
Usable Host Range	172.16.0.1-172.16.127.254	172.16.128.1-172.16.255.254

IPV6:

Features of IPV6:



- **Larger address space:**
  - Global reachability and flexibility
  - Aggregation
  - Multihoming
  - Autoconfiguration
  - Plug and play
  - End-to-end without NAT
  - Renumbering
- **Mobility and security:**
  - Mobile IP RFC-compliant
  - IPsec mandatory (or native) for IPv6
- **Simple header:**
  - Routing efficiency
  - Performance and forwarding rate scalability
  - No broadcasts
  - No checksums
  - Extension headers
  - Flow labels
- **Transition richness:**
  - Dual stack
  - 6to4 tunnels
  - Translation

- **Larger address space**: Offers improved global reachability and flexibility; the aggregation of prefixes that are announced in routing tables; multihoming to several Internet service providers (ISPs) auto configuration that can include link-layer addresses in the address space; plug-and-play options; public-to private readdressing end to end without address translation; and simplified mechanisms for address renumbering and modification.
- **Simpler header:** Provides better routing efficiency; no broadcasts and thus no potential threat of broadcast storms; no requirement for processing checksums; simpler and more efficient extension header mechanisms; and flow labels for per-flow processing with no need to open the transport inner packet to identify the various traffic flows.
- **Mobility and security:** Ensures compliance with mobile IP and IPsec standards functionality; mobility is built in, so any IPv6 node can use it when necessary; and enables people to move around in networks with mobile network devices—with many having wireless connectivity.

  Mobile IP is an Internet Engineering Task Force (IETF) standard available for both IPv4 and IPv6. The standard enables mobile devices to move without breaks in established network connections. Because IPv4 does not automatically provide this kind of mobility, you must add it with additional configurations.

  IPsec is the IETF standard for IP network security, available for both IPv4 and IPv6. Although the functionalities are essentially identical in both environments, IPsec is mandatory in IPv6. IPsec is enabled on every IPv6 node and is available for use. The availability of IPsec on all nodes makes the IPv6 Internet more secure. IPsec also requires keys for each party, which implies a global key deployment and distribution.
- **Transition richness:** You can incorporate existing IPv4 capabilities in IPv6 in the following ways:
  - ➢ Configure a dual stack with both IPv4 and IPv6 on the interface of a network device.

- Use the technique IPv6 over IPv4 (also called 6to4 tunneling), which uses an IPv4 tunnel to carry IPv6 traffic. This method (RFC 3056) replaces IPv4-compatible tunneling (RFC 2893). Cisco IOS Software Release 12.3(2)T (and later) also allows protocol translation (NAT-PT) between IPv6 and IPv4. This translation allows direct communication between hosts speaking different protocols.

IPV6 Header

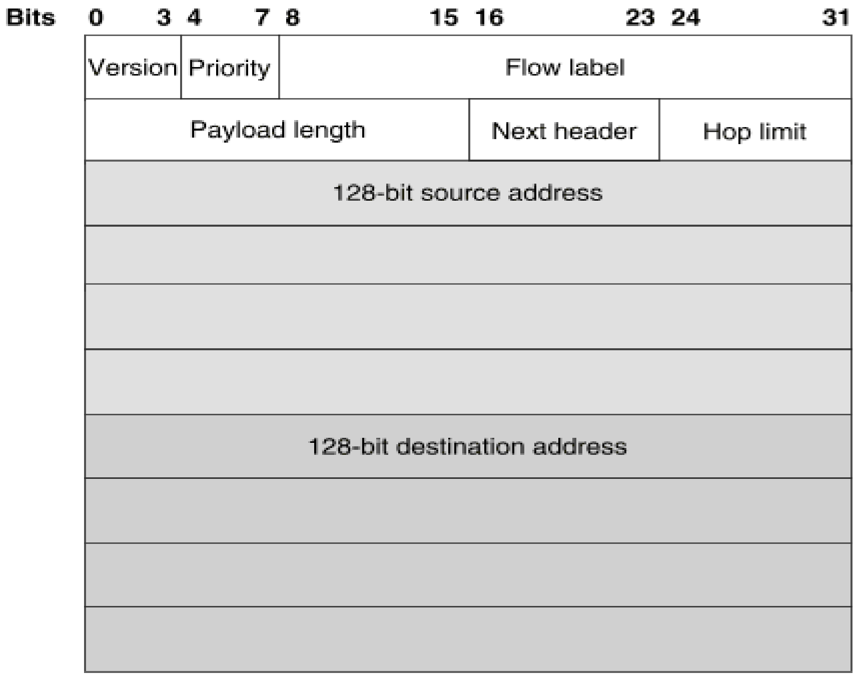


Figure: IPV6 Header

Specifically, IPv6 omits the following fields in its header.

- header length (the length is constant)
- identification
- flags
- fragment offset (this is moved into fragmentation extension headers)
- header checksum (the upper-layer protocol or security extension header handles data integrity)

IPv6 options improve over IPv4 by being placed in separate extension headers that are located between the IPv6 header and the transport-layer header in a packet. Most extension headers are not examined or processed by any router along a packet's delivery path until it arrives at its final destination. This mechanism improves router performance for packets containing options. In IPv4, the presence of any options requires the router to examine all options.

Another improvement is that IPv6 extension headers, unlike IPv4 options, can be of arbitrary length and the total amount of options that a packet carries is not limited to 40 bytes. This feature, and the manner in which it is processed, permit IPv6 options to be used for functions that were not practical in IPv4, such as the IPv6 Authentication and Security Encapsulation options.

By using extension headers, instead of a protocol specifier and options fields, newly defined extensions can be integrated more easily into IPv6.

IPV6 Addressing:

Address Representation:
Represented by breaking 128 bit into Eight 16-bit segments (Each 4 Hex character each). Each segment is written in Hexadecimal separated by colons. Hex digit are not case sensitive.
Rule 1:
Drop leading zeros:
2001:0050:0000:0235:0ab4:3456:456b:e560
2001:050:0:235:ab4:3456:456b:e560
Rule2:
Successive fields of zeros can be represented as "::" , But double colon appear only once in the address. FF01:0:0:0:0:0:0:1
FF01::1
Note : An address parser identifies the number of missing zeros by separating the two parts and entering 0 until the 128 bits are complete. If two "::" notations are placed in the address, there is no way to identify the size of each block of zeros.

IPV4 VS IPV6

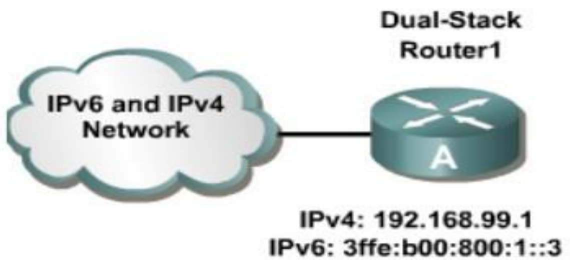
IPV4	IPV6
source and destination addresses are 32 bits	Source and destination addresses are 128 bits.
ipv4 support small address space.	Supports a very large address space sufficient for each and every people on earth.
ipv4 header includes checksum.	ipv6 header doesn't includes the checksum.
addresses are represented in dotted decimal format. (Eg. 192.168.5.1)	Addresses are represented in 16-bit segments Each segment is written in Hexadecimal separated by colons. (Eg. 2001:0050:020c:0235:0ab4:3456:456b:e560
Header includes options	All optional data is moved to IPV6 extension header
Broadcast address are used to send traffic to all nodes on a subnet	There is no IPV6 broadcast address. Instead a link local scope all-nodes multicast address is used
No identification of packet flow for QOS handling by router is present within the ipv4 header	Packet flow identification for QOS handling by routers is present within the IPV6 header using the flow label field.
uses host address (A) resource records in the Domain name system(DNS) to map host names to ipv4 addresses.	Uses AAAA records in the DNS to map host names ipv6 addresses.
Both routers and the sending host fragment packets.	Only the sending host fragments packets; routers do not.
ICMP Router Discovery is used to determine the IPv4 address of the best default gateway, and it is optional	. ICMPv6 Router Solicitation and Router Advertisement messages are used to determine the IP .address of the best default gateway, and they are required.

IPV6 Transition Mechanism:

I. Dual Stack
II. Tunneling Technique
III. Translation technique

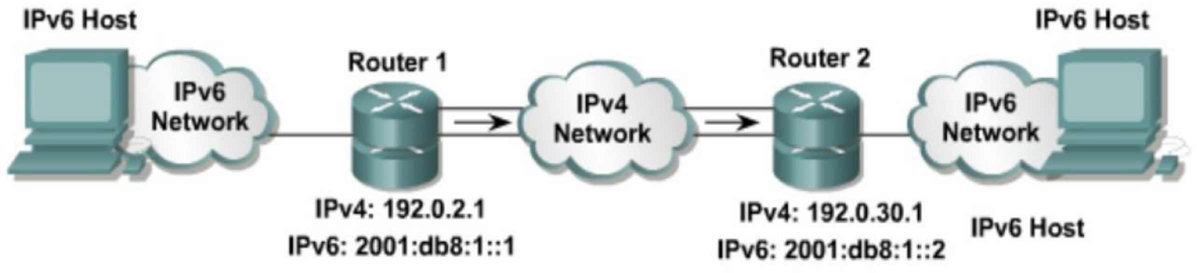
Dual Stack:

Dual stack is an integration method where a node has implementation and connectivity to both Ipv4 and ipv6 network. If both ipv4 and ipv6 are configured on an interface, this interface is dual-stacked.



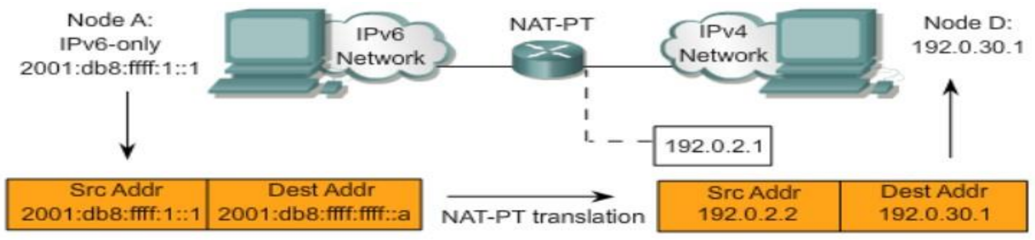
Tunneling Technique

With manually configured IPv6 tunnels, an IPv6 address is configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.



NAT-Protocol Translation (NAT-PT)

It is a translation mechanism that sits between an IPv6 network and an Ipv4 network. The translator translates IPv6 packets into IPv4 packets and vice versa.



Design issues for the network layer.

The network layer has been designed with the following goals:

I. The services provided should be independent of the underlying technology. Users of the service need not be aware of the physical implementation of the network - for all they know, they're messages could be transported via carrier pigeon! This design goal has great importance when we consider the great variety of networks in operation. In the area of Public networks, networks in underdeveloped countries are nowhere near the technological prowess of those in the countries like the US or Ireland. The design of the layer must not disable us from connecting to networks of different technologies.

II. The transport layer (that is the host computer) should be shielded from the number, type and different topologies of the subnets he uses. That is, all the transport layer want is a communication link, it need not know how that link is made.

III. Finally, there is a need for some uniform addressing scheme for network addresses.

With these goals in mind, two different types of service emerged: Connection oriented and connectionless. A connection-oriented service is one in which the user is given a "reliable" end to end connection. To communicate, the user requests a connection, then uses the connection to his heart’s content, and then closes the connection. A telephone call is the classic example of a connection oriented service.

In a connection-less service, the user simply bundles his information together, puts an address on it, and then sends it off, in the hope that it will reach its destination. There is no guarantee that the bundle will arrive. So connection less service is one reminiscent of the postal system. A letter is sent, that is, put in the post box. It is then in the "postal network" where it gets bounced around and hopefully will leave the network in the correct place, that is, in the addressee's letter box. We can never be totally sure that the letter will arrive, but we know that there is a high probability that it will, and so we place our trust in the postal network.

Now, the question was which service would the network layer provide, a connection-oriented or a connectionless one?

With a connection oriented service, the user must pay for the length (ie the duration) of his connection. Usually this will involve a fixed start up fee. Now, if the user intends to send a constant stream of data down the line, this is great - he is given a reliable service for as long as he wants. However, say the user wished to send only a packet or two of data - now the cost of setting up the connection greatly overpowers the cost of sending that one packet. Consider also the case where the user wishes to send a packet once every 3 minutes. In a connection-oriented service, the line will thus be idle for the majority of the time, thus wasting bandwidth. So, connection-oriented services seem to be useful only when the user wishes to send a constant stream of data.

One would therefore think that the reliable nature of the connection oriented service would prompt people to choose it over the connectionless service - this is in fact not the case. One can never ensure that the network is 100% reliable; in fact for many applications we must assume that the network is not reliable at all. With this in mind, many applications perform their own error detection, flow and congestion control at a higher level in the protocol stack, that is, on their own machine, in the transport layer. So, if the sender and the receiver are going to engage in their own control mechanisms, why put this functionality into the network layer? This is the argument for the connectionless service: the network layer should provide a raw means of sending packets from a to b, and that is all. Proponents of this argument are quick to point out that the standard of our networks has increased greatly in the past years, that packets of information rarely ever do get lost, so much of the correction facilities in the network layer are redundant and serve only to complicate the layer and slow down transfer.

It’s interesting to note here that it is easy to provide a connection oriented service over an inherently connectionless service, so in fact defining the service of the network layer as connectionless is the general solution. However, at the time of defining the network layer, the controversy between the two camps was (and still is) unresolved, and so instead of deciding on one service, the ISO allowed both.

Routing and Routing Protocols:

The primary responsibility of a router is to direct packets destined for local and remote networks by:

- Determining the best path to send packets
- Forwarding packets toward their destination

The router uses its routing table to determine the best path to forward the packet. When the router receives a packet, it examines its destination IP address and searches for the best match with a network address in the router's routing table. The routing table also includes the interface to be used to forward the packet. Once a match is found, the router encapsulates the IP packet into the data link frame of the outgoing or exit interface, and the packet is then forwarded toward its destination.

Static Routes:
Static routes are configured manually; network administrators must add and delete static routes to reflect any network topology changes. In a large network, the manual maintenance of routing tables could require a lot of administrative time. On small networks with few possible changes, static routes require very little maintenance. Static routing is not as scalable as dynamic routing because of the extra administrative requirements. Even in large networks, static routes that are intended to accomplish a specific purpose are often configured in conjunction with a dynamic routing protocol.

When to use static Routing:
A network consists of only a few routers. Using a dynamic routing protocol in such a case does not present any substantial benefit. On the contrary, dynamic routing may add more administrative overhead.

A network is connected to the Internet only through a single ISP. There is no need to use a dynamic routing protocol across this link because the ISP represents the only exit point to the Internet.

A large network is configured in a hub-and-spoke topology. A hub-and-spoke topology consists of a central location (the hub) and multiple branch locations (spokes), with each spoke having only one connection to the hub. Using dynamic routing would be unnecessary because each branch has only one path to a given destination-through the central location.

Connected Routes:
Those network that are directly connected to the Router are called connected routes and are

not needed to configure on the router for routing. They are automatically routed by the Router.

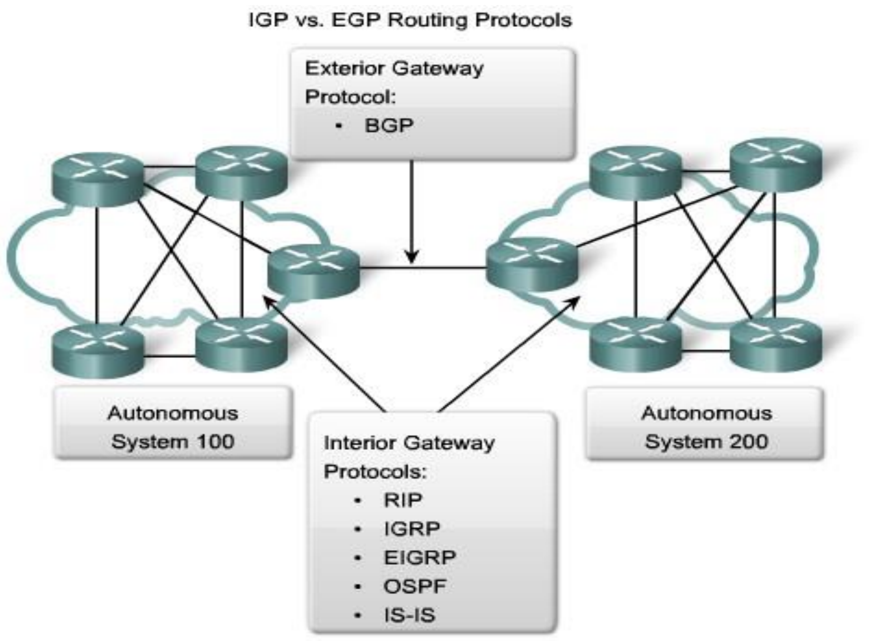
Dynamic Routes:
Dynamic routing protocol uses a route that a routing protocol adjusts automatically for topology or traffic changes.

Non-adaptive routing algorithm When a ROUTER uses a non-adaptive routing algorithm it consults a static table in order to determine to which computer it should send a PACKET of data. This is in contrast to an ADAPTIVE ROUTING ALGORITHM, which bases its decisions on data which reflects current traffic conditions (Also called static route)

Adaptive routing algorithm When a ROUTER uses an adaptive routing algorithm to decide the next computer to which to transfer a PACKET of data, it examines the traffic conditions in order to determine a route which is as near optimal as possible. For example, it tries to pick a route which involves communication lines which have light traffic. This strategy is in contrast to a NON-ADAPTIVE ROUTING ALGORITHM. (Also called Dynamic route)

Routing Protocol:
A routing protocol is the communication used between routers. A routing protocol allows routers to share information about networks and their proximity to each other. Routers use this information to build and maintain routing tables.

Autonomous System:
An AS is a collection of networks under a common administration that share a common routing strategy. To the outside world, an AS is viewed as a single entity. The AS may be run by one or more operators while it presents a consistent view of routing to the external world. The American Registry of Internet Numbers (ARIN), a service provider, or an administrator assigns a 16-bit identification number to each AS.


IGP vs. EGP Routing Protocols
Exterior Gateway Protocol:
• BGP
Autonomous System 100
Autonomous System 200
Interior Gateway Protocols:
• RIP
• IGRP
• EIGRP
• OSPF
• IS-IS

Dynamic Routing Protocol:

A. Interior Gateway protocol (IGP)
   I. Distance Vector Protocol
   II. Link State Protocol
B. Exterior Gateway Protocol (EGP)

Interior gateway protocol (IGP): Within one Autonomous System.

Exterior Routing Protocol (EGP): Between the Autonomous System. Example BGP (Boarder gateway protocol)

Metric:
There are cases when a routing protocol learns of more than one route to the same destination. To select the best path, the routing protocol must be able to evaluate and differentiate between the available paths. For this purpose a metric is used. A metric is a value used by routing protocols to assign costs to reach remote networks. The metric is used to determine which path is most preferable when there are multiple paths to the same remote network.

Each routing protocol uses its own metric. For example, RIP uses hop count, EIGRP uses a combination of bandwidth and delay, and Cisco's implementation of OSPF uses bandwidth.

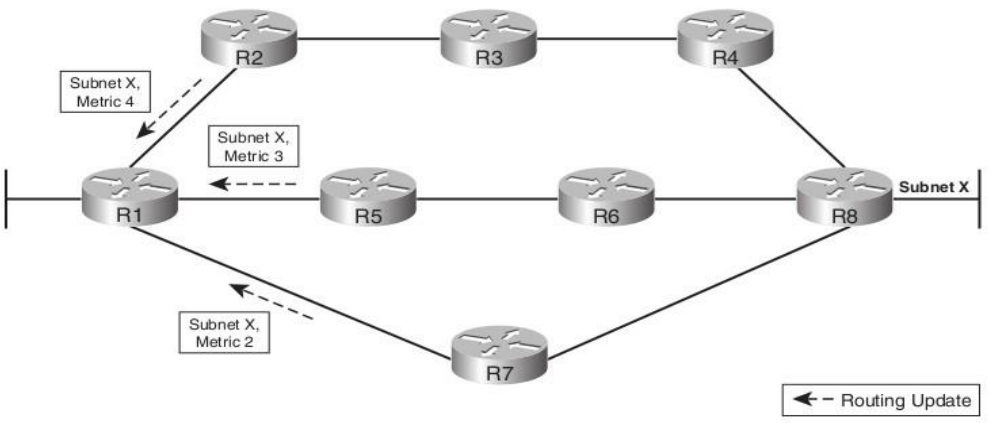
Distance Vector Routing Algorithm:

As the name implies, distance vector means that routes are advertised as vectors of distance and direction. Distance is defined in terms of a metric such as hop count and direction is simply the next-hop router or exit interface. A router using a distance vector routing protocol does not have the knowledge of the entire path to a destination network. Instead the router knows only:

- The direction or interface in which packets should be forwarded and
- The distance or how far it is to the destination network.

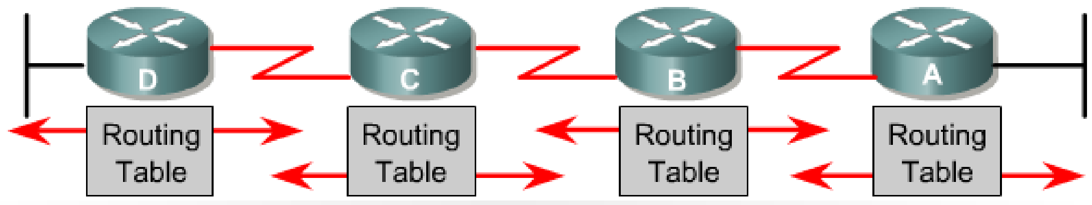
To show you more exactly what a distance vector protocol does, Figure shows a view of what a router learns with a distance vector routing protocol. The figure shows an internetwork in which R1 learns about three routes to reach subnet X:

- The four-hop route through R2
- The three-hop route through R5
- The two-hop route through R7

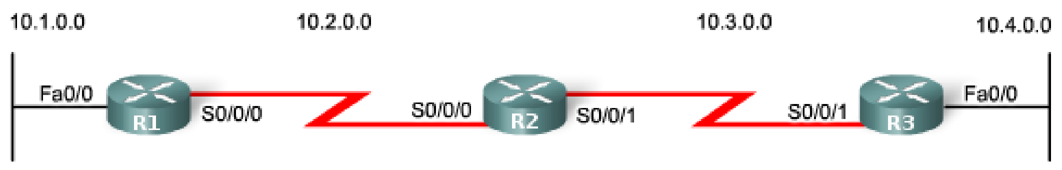


R1 learns about the subnet, and a metric associated with that subnet, and nothing more. R1 must then pick the best route to reach subnet X. In this case, it picks the two-hop route through R7, because that route has the lowest metric.

Distance vector protocols typically use the Bellman-Ford algorithm for the best path route determination.



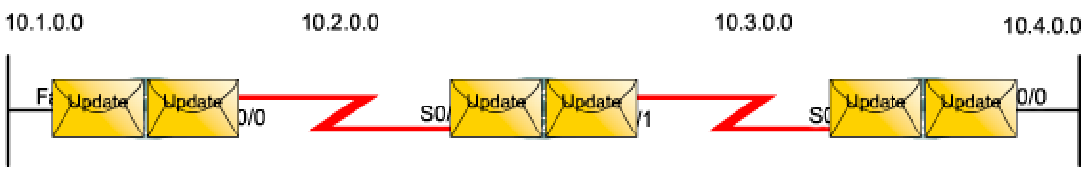
Pass periodic copies of a routing table to neighbor routers and accumulate distance vectors.



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0

Network	Interface	Hop
10.3.0.0	S0/0/0	0
10.4.0.0	Fa0/0	0

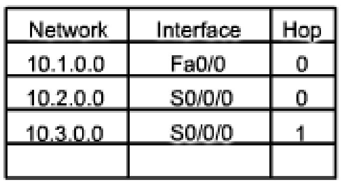


Initial Update:
R1

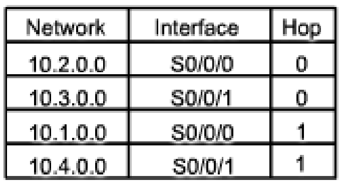
- Sends an update about network 10.1.0.0 out the Serial0/0/0 interface
- Sends an update about network 10.2.0.0 out the FastEthernet0/0 interface
- Receives update from R2 about network 10.3.0.0 with a metric of 1
- Stores network 10.3.0.0 in the routing table with a metric of 1

R2

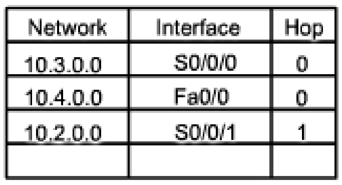
- Sends an update about network 10.3.0.0 out the Serial 0/0/0 interface
- Sends an update about network 10.2.0.0 out the Serial 0/0/1 interface
- Receives an update from R1 about network 10.1.0.0 with a metric of 1
- Stores network 10.1.0.0 in the routing table with a metric of 1
- Receives an update from R3 about network 10.4.0.0 with a metric of 1
- Stores network 10.4.0.0 in the routing table with a metric of 1



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1



Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1



Network	Interface	Hop
10.3.0.0	S0/0/0	0
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1

R3

- Sends an update about network 10.4.0.0 out the Serial 0/0/0 interface
- Sends an update about network 10.3.0.0 out the FastEthernet0/0
- Receives an update from R2 about network 10.2.0.0 with a metric of 1
- Stores network 10.2.0.0 in the routing table with a metric of 1

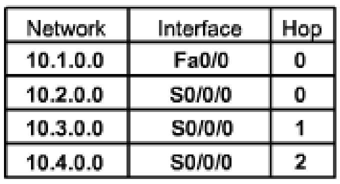
After this first round of update exchanges, each router knows about the connected networks of their directly connected neighbors. However, did you notice that R1 does not yet know about 10.4.0.0 and that R3 does not yet know about 10.1.0.0? Full knowledge and a converged network will not take place until there is another exchange of routing information.

Next Update:
R1

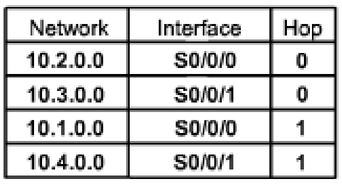
- Sends an update about network 10.1.0.0 out the Serial 0/0/0 interface.
- Sends an update about networks 10.2.0.0 and 10.3.0.0 out the FastEthernet0/0 interface.
- Receives an update from R2 about network 10.4.0.0 with a metric of 2.
- Stores network 10.4.0.0 in the routing table with a metric of 2.
- Same update from R2 contains information about network 10.3.0.0 with a metric of 1. There is no change; therefore, the routing information remains the same.

R2

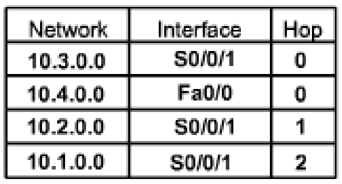
- Sends an update about networks 10.3.0.0 and 10.4.0.0 out of Serial 0/0/0 interface.
- Sends an update about networks 10.1.0.0 and 10.2.0.0 out of Serial 0/0/1 interface.
- Receives an update from R1 about network 10.1.0.0. There is no change; therefore, the routing information remains the same.
- Receives an update from R3 about network 10.4.0.0. There is no change; therefore, the routing information remains the same.



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2



Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1



Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

R3

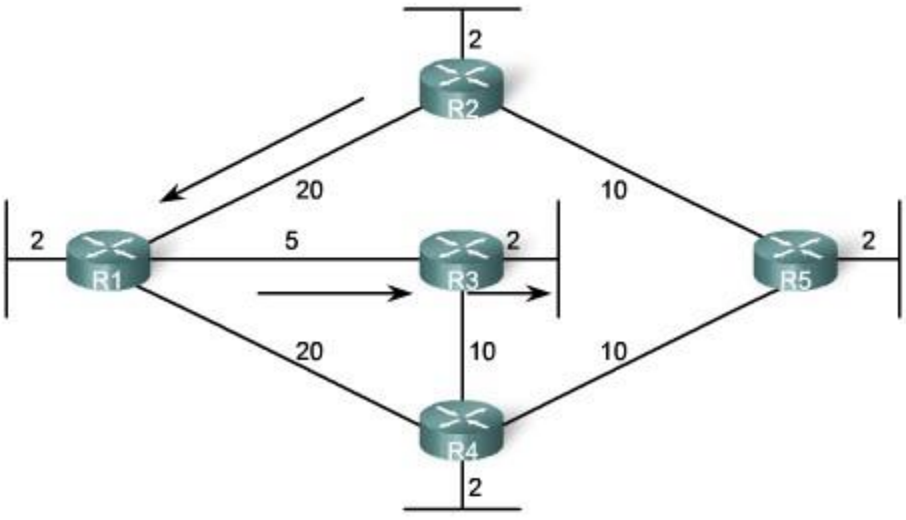
- Sends an update about network 10.4.0.0 out the Serial 0/0/0 interface.
- Sends an update about networks 10.2.0.0 and 10.3.0.0 out the FastEthernet0/0 interface.
- Receives an update from R2 about network 10.1.0.0 with a metric of 2.
- Stores network 10.1.0.0 in the routing table with a metric of 2.
- Same update from R2 contains information about network 10.2.0.0 with a metric of 1. There is no change; therefore, the routing information remains the same.

Note: Distance vector routing protocols typically implement a technique known as split horizon. Split horizon prevents information from being sent out the same interface from which it was received. For example, R2 would not send an update out Serial 0/0/0 containing the network 10.1.0.0 because R2 learned about that network through Serial 0/0/0.

Link State Routing Algorithm:

Also known as shortest path Routing algorithm.
Link states:
Information about the state of (Router interfaces) links is known as link-states. As you can see in the figure, this information includes:

- The interface's IP address and subnet mask.
- The type of network, such as Ethernet (broadcast) or Serial point-to-point link.
- The cost of that link.
- Any neighbor routers on that link.



Shortest Path for host on R2 LAN to reach host on R3 LAN:
R2 to R1 (20) + R1 to R3 (5) + R3 to LAN (2) = 27

So exactly how does a link-state routing protocol work? All routers will complete the following generic link-state routing process to reach a state of convergence:

1. Each router learns about its own links, its own directly connected networks. This is done by detecting that an interface is in the up state.
2. Each router is responsible for meeting its neighbors on directly connected networks. link state routers do this by exchanging Hello packets with other link-state routers on directly connected networks.
3. Each router builds a Link-State Packet (LSP) containing the state of each directly connected link. This is done by recording all the pertinent information about each neighbor, including neighbor ID, link type, and bandwidth.
4. Each router floods the LSP to all neighbors, who then store all LSPs received in a database. Neighbors then flood the LSPs to their neighbors until all routers in the area have received the LSPs. Each router stores a copy of each LSP received from its neighbors in a local database.

5. Each router uses the database to construct a complete map of the topology and computes the best path to each destination network. Like having a road map, the router now has a complete map of all destinations in the topology and the routes to reach them. The SPF algorithm is used to construct the map of the topology and to determine the best path to each network.

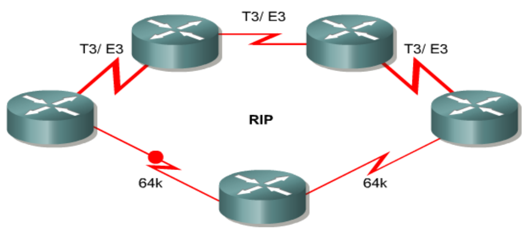
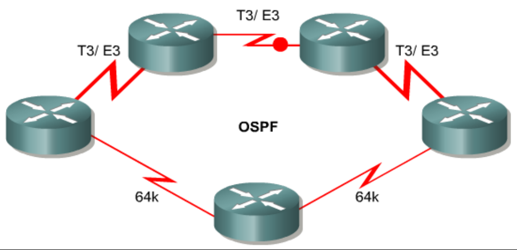
Advantages of Link state Routing protocol:

Build the topological map:
Link-state routing protocols create a topological map, or SPF tree of the network topology. Distance vector routing protocols do not have a topological map of the network.

Faster Convergence:
When receiving a Link-state Packet (LSP), link-state routing protocols immediately flood the LSP out all interfaces except for the interface from which the LSP was received. This way, it achieves the faster convergence. With distance vector routing algorithm, router needs to process each routing update and update its routing table before flooding them out other interfaces.

Event Driven Updates:
After the initial flooding of LSPs, link-state routing protocols only send out an LSP when there is a change in the topology. The LSP contains only the information regarding the affected link. Unlike some distance vector routing protocols, link-state routing protocols do not send periodic updates.

Distance vector vs. Link state:

S. No	Distance Vector	Link State
1	Uses hop count as Metric	Uses shortest path.
2	2View the network from the perspective of neighbor.	Gets common view of entire network topology.
3	Has frequent and periodic updates	Has event triggered updates.
4	Slow convergence	Faster convergence
5	Susceptible to routing loops.	Not as susceptible to routing loops.
6	Easy to configure and administer.	Difficult to configure and administer.
7	Requires less memory and processing power of routers	Requires more processing power and memory than distance vector.
8	Consumes a lot of Bandwidth.	Consumes less BW than distance vector
9	Passes copies of routing table to neighbor routers.	Passes link-state routing updates to other routers.
10	Eg. RIP 	Eg. OSPF  T3/ E3 T3/ E3 T3/ E3 OSPF 64k 64k