11/21/2017

Advanced networking with IPv6

Prepared By: Er Loknath Regmi

Chapter -1

Networking Protocols

Introduction:

Networking engineering is a complicated task, which involves software, firmware, chip level engineering, hardware, and electric pulses. To ease network engineering, the whole networking concept is divided into multiple layers. Each layer is involved in some particular task and is independent of all other layers. But as a whole, almost all networking tasks depend on all of these layers. Layers share data between them and they depend on each other only to take input and send output.

Layered Tasks:

In layered architecture of Network Model, one whole network process is divided into small tasks. Each small task is then assigned to a particular layer which works dedicatedly to process the task only. Every layer does only specific work.

In layered communication system, one layer of a host deals with the task done by or to be done by its peer layer at the same level on the remote host. The task is either initiated by layer at the lowest level or at the top most level. If the task is initiated by the-top most layer, it is passed on to the layer below it for further processing. The lower layer does the same thing, it processes the task and passes on to lower layer. If the task is initiated by lower most layer, then the reverse path is taken.

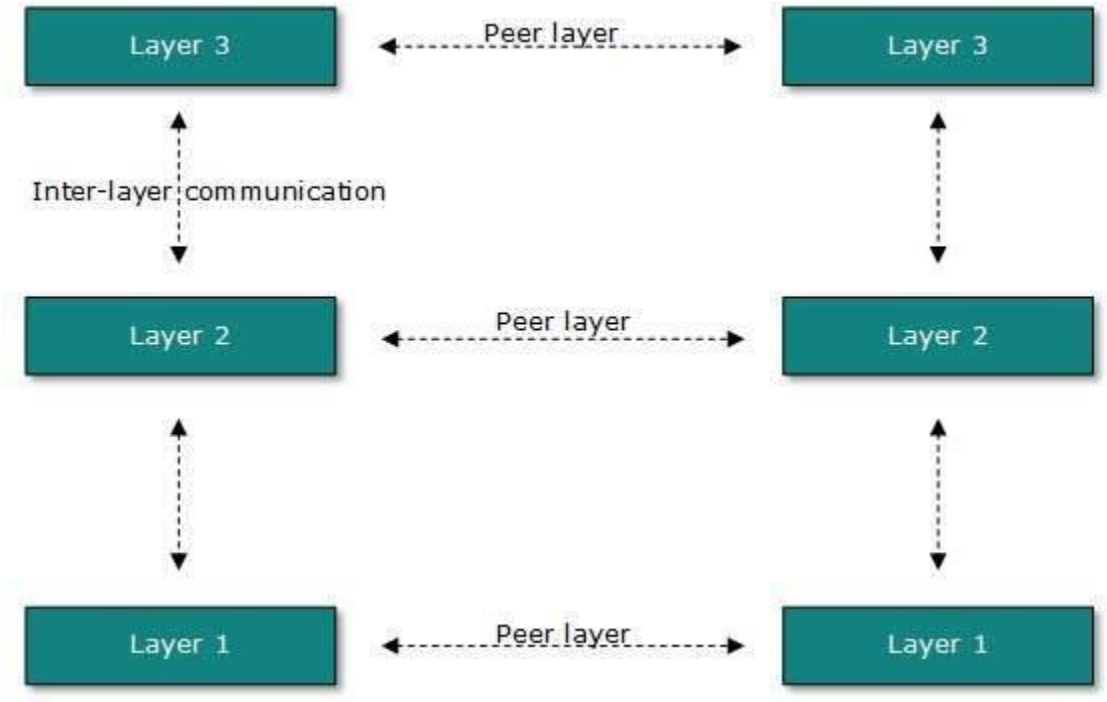


Fig: Laired Architecture Basis

Every layer clubs together all procedures, protocols, and methods which it requires to execute its piece of task. All layers identify their counterparts by means of encapsulation header and tail.

OSI Model:

Open System Interconnect is an open standard for all communication systems. OSI model is established by International Standard Organization (ISO). This model has seven layers:

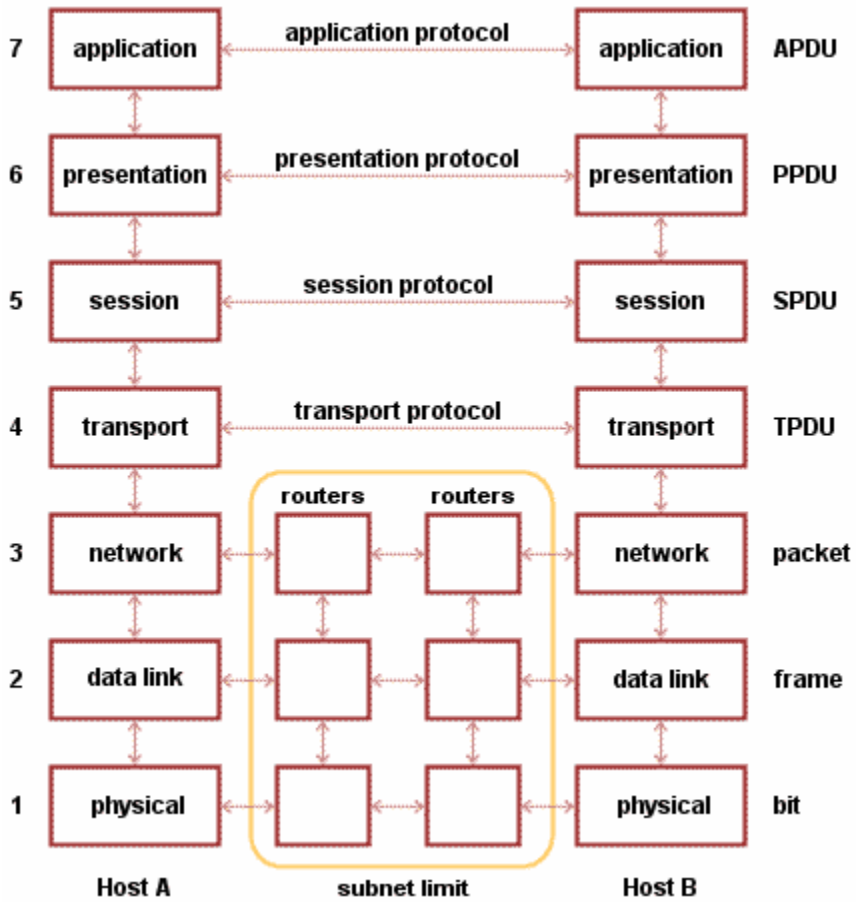


Fig: OSI refrence model

- **Application Layer**: This layer is responsible for providing interface to the application user. This layer encompasses protocols which directly interact with the user.
- **Presentation Layer**: This layer defines how data in the native format of remote host should be presented in the native format of host.
- **Session Layer**: This layer maintains sessions between remote hosts. For example, once user/password authentication is done, the remote host maintains this session for a while and does not ask for authentication again in that time span.

- **Transport Layer**: This layer is responsible for end-to-end delivery between hosts.
- **Network Layer**: This layer is responsible for address assignment and uniquely addressing hosts in a network.
- **Data Link Layer**: This layer is responsible for reading and writing data from and onto the line. Link errors are detected at this layer.
- **Physical Layer**: This layer defines the hardware, cabling wiring, power output, pulse rate etc.

Internet Model:

Internet uses TCP/IP protocol suite, also known as Internet suite. This defines Internet Model which contains four layered architecture. OSI Model is general communication model but Internet Model is what the internet uses for all its communication. The internet is independent of its underlying network architecture so is its Model. This model has the following layers:

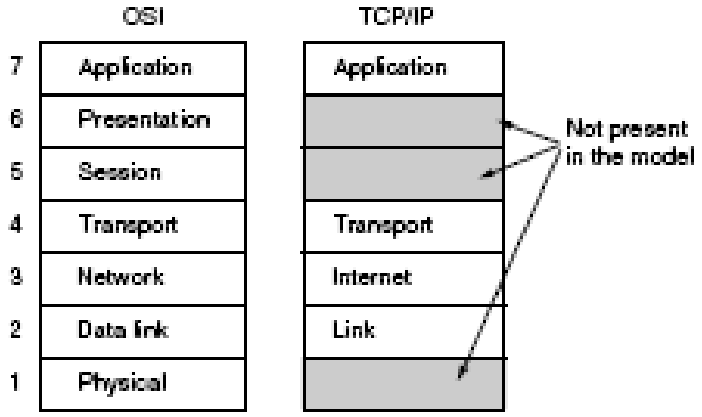


Fig: TCP/IP Protocol Suite

- **Application Layer**: This layer defines the protocol which enables user to interact with the network. For example, FTP, HTTP etc.
- **Transport Layer**: This layer defines how data should flow between hosts. Major protocol at this layer is Transmission Control Protocol (TCP). This layer ensures data delivered between hosts is in-order and is responsible for end-to-end delivery.
- **Internet Layer**: Internet Protocol (IP) works on this layer. This layer facilitates host addressing and recognition. This layer defines routing.

- **Link Layer**: This layer provides mechanism of sending and receiving actual data. Unlike its OSI Model counterpart, this layer is independent of underlying network architecture and hardware.

The Internet Protocol (IP):

IP is sometimes referred to as an unreliable protocol. This does not mean that IP will not accurately deliver data across a network. IP is unreliable because it does not perform error checking and correction. That function is handled by upper layer protocols from the transport or application layers.
IP performs the following operations:

→ Defines a packet and an addressing scheme
→ Transfers data between the internet layer and network access layer.
→ Routers packets to remote hosts.
→ The main function of IP is forwarding and addressing in the internet.

UDP (User Datagram Protocol):

UDP is the connectionless transport protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagram without guaranteed delivery. It relies on higher layer protocols to handle error and retransmit data.
UDP doesn't use window or Asks reliability is provided by application layer protocols. UDP is designed for applications that do not need to put sequence of segments together.
The following application layer protocols use UDP: TFTP, SNMP, DHCP, and DNS
Hence,

- Used in transport layer
- Offers unreliable connectionless service
- Provides faster service than that of TCP.
- Offers minimum error checking mechanism.
- Supports multicasting because connectionless.
- Offers minimum flow control mechanism.
- Also used by SNMP (Simple Network Management Protocol)

UDP Segment Structure:

Source port number (16)	Destination port number (16)
UDP segment length (16)	UDP checksum (16)
Data	

→ Source port – number of the port that sends data.
→ Destination port – Number of the port that receives data.

→ Length – calculated of bytes in header and data.
→ Checksum – calculated checksum of the header and data field.
→ Data – upper-layer protocol data.

Transmission Control Protocol (TCP):

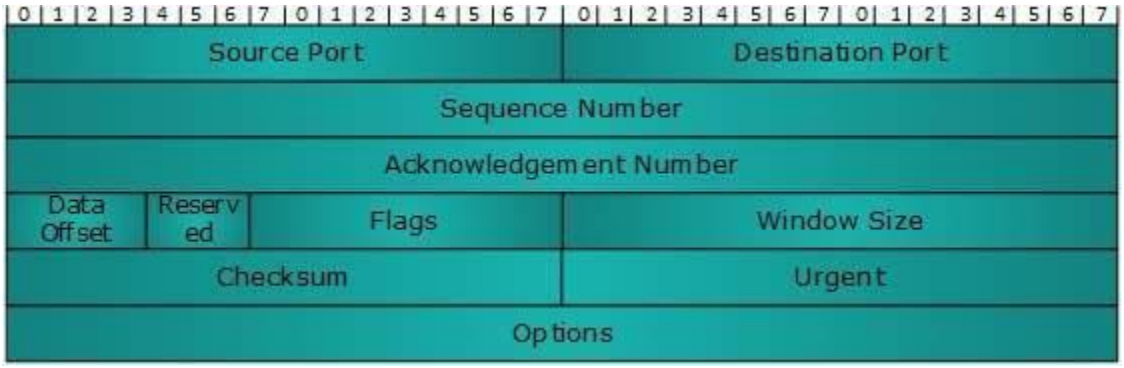
The transmission Control Protocol (TCP) is one of the most important protocols of Internet Protocols suite. It is most widely used protocol for data transmission in communication network such as internet.

Features

- TCP is reliable protocol. That is, the receiver always sends either positive or negative acknowledgement about the data packet to the sender, so that the sender always has bright clue about whether the data packet is reached the destination or it needs to resend it.
- TCP ensures that the data reaches intended destination in the same order it was sent.
- TCP is connection oriented. TCP requires that connection between two remote points be established before sending actual data.
- TCP provides error-checking and recovery mechanism.
- TCP provides end-to-end communication.
- TCP provides flow control and quality of service.
- TCP operates in Client/Server point-to-point mode.
- TCP provides full duplex server, i.e. it can perform roles of both receiver and sender.

Header:

The length of TCP header is minimum 20 bytes long and maximum 60 bytes.



- **Source Port (16-bits)** - It identifies source port of the application process on the sending device.
- **Destination Port (16-bits)** - It identifies destination port of the application process on the receiving device.
- **Sequence Number (32-bits)** - Sequence number of data bytes of a segment in a session.
- **Acknowledgement Number (32-bits)** - When ACK flag is set, this number contains the next sequence number of the data byte expected and works as acknowledgement of the previous data received.
- **Data Offset (4-bits)** - This field implies both, the size of TCP header (32-bit words) and the offset of data in current packet in the whole TCP segment.
- **Reserved (3-bits)** - Reserved for future use and all are set zero by default.
- **Flags (1-bit each)**
    - **NS** - Nonce Sum bit is used by Explicit Congestion Notification signaling process.
    - **CWR** - When a host receives packet with ECE bit set, it sets Congestion Windows Reduced to acknowledge that ECE received.
    - **ECE** -It has two meanings:
        - If SYN bit is clear to 0, then ECE means that the IP packet has its CE (congestion experience) bit set.
        - If SYN bit is set to 1, ECE means that the device is ECT capable.
    - **URG** - It indicates that Urgent Pointer field has significant data and should be processed.
    - **ACK** - It indicates that Acknowledgement field has significance. If ACK is cleared to 0, it indicates that packet does not contain any acknowledgement.
    - **PSH** - When set, it is a request to the receiving station to PUSH data (as soon as it comes) to the receiving application without buffering it.
    - **RST** - Reset flag has the following features:
        - It is used to refuse an incoming connection.
        - It is used to reject a segment.
        - It is used to restart a connection.
    - **SYN** - This flag is used to set up a connection between hosts.

- **FIN** - This flag is used to release a connection and no more data is exchanged thereafter. Because packets with SYN and FIN flags have sequence numbers, they are processed in correct order.

- **Windows Size** - This field is used for flow control between two stations and indicates the amount of buffer (in bytes) the receiver has allocated for a segment, i.e. how much data is the receiver expecting.
- **Checksum** - This field contains the checksum of Header, Data and Pseudo Headers.
- **Urgent Pointer** - It points to the urgent data byte if URG flag is set to 1.
- **Options** - It facilitates additional options which are not covered by the regular header. Option field is always described in 32-bit words. If this field contains data less than 32-bit, padding is used to cover the remaining bits to reach 32-bit boundary.

Network Layer Routing:

When a device has multiple paths to reach a destination, it always selects one path by preferring it over others. This selection process is termed as Routing. Routing is done by special network devices called routers or it can be done by means of software processes.The software based routers have limited functionality and limited scope.

A router is always configured with some default route. A default route tells the router where to forward a packet if there is no route found for specific destination. In case there are multiple path existing to reach the same destination, router can make decision based on the following information:

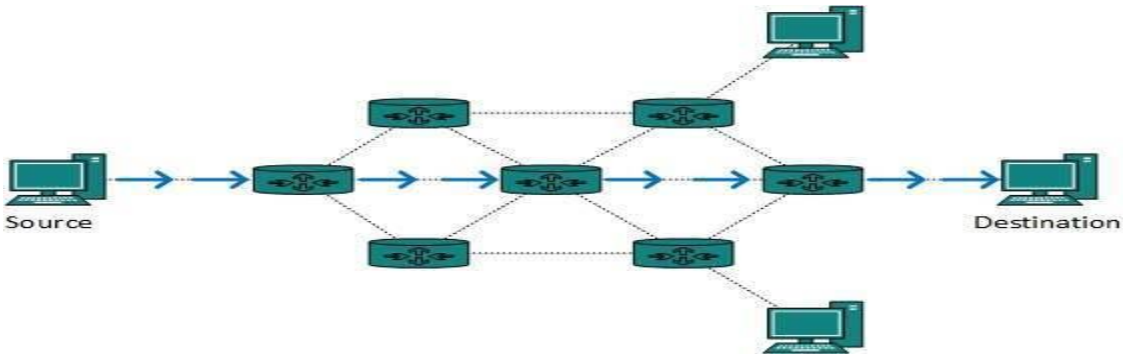
- Hop Count
- Bandwidth
- Metric
- Prefix-length
- Delay

Routes can be statically configured or dynamically learnt. One route can be configured to be preferred over others.

Unicast routing:

Most of the traffic on the internet and intranets known as unicast data or unicast traffic is sent with specified destination. Routing unicast data over the internet is called unicast routing. It is the

simplest form of routing because the destination is already known. Hence the router just has to look up the routing table and forward the packet to next hop.



Broadcast routing:

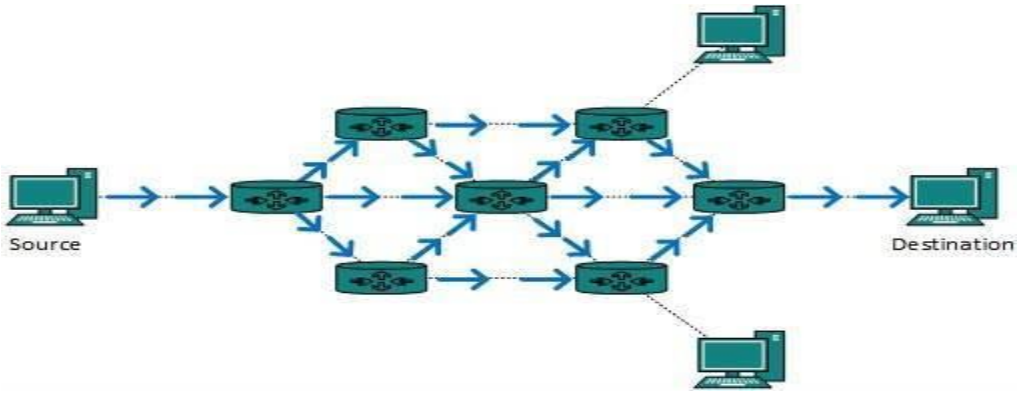
By default, the broadcast packets are not routed and forwarded by the routers on any network. Routers create broadcast domains. But it can be configured to forward broadcasts in some special cases. A broadcast message is destined to all network devices.

Broadcast routing can be done in two ways (algorithm):

- A router creates a data packet and then sends it to each host one by one. In this case, the router creates multiple copies of single data packet with different destination addresses. All packets are sent as unicast but because they are sent to all, it simulates as if router is broadcasting.

  This method consumes lots of bandwidth and router must destination address of each node.

- Secondly, when router receives a packet that is to be broadcasted, it simply floods those packets out of all interfaces. All routers are configured in the same way.

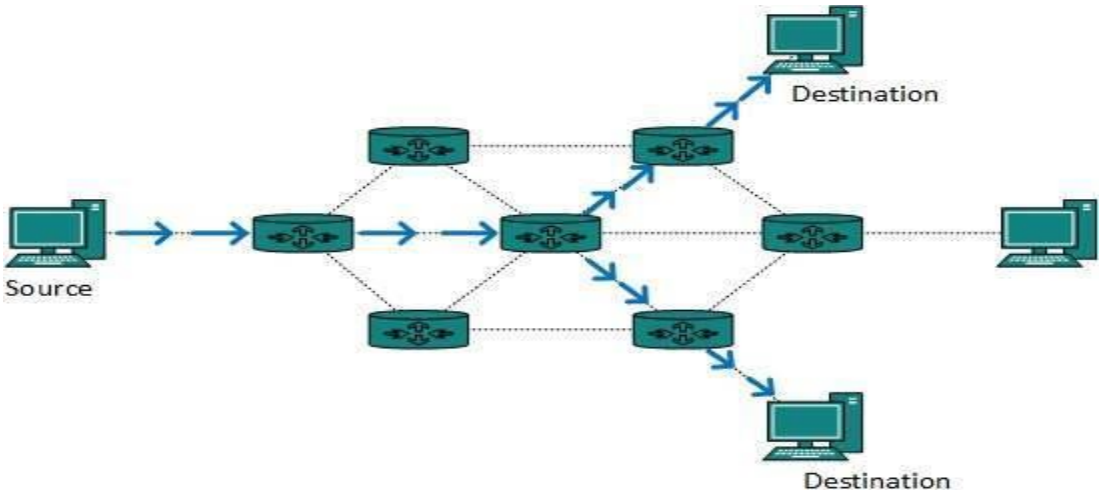


This method is easy on router's CPU but may cause the problem of duplicate packets received from peer routers.

Reverse path forwarding is a technique, in which router knows in advance about its predecessor from where it should receive broadcast. This technique is used to detect and discard duplicates.

Multicast Routing :

Multicast routing is special case of broadcast routing with significance difference and challenges. In broadcast routing, packets are sent to all nodes even if they do not want it. But in Multicast routing, the data is sent to only nodes which wants to receive the packets.

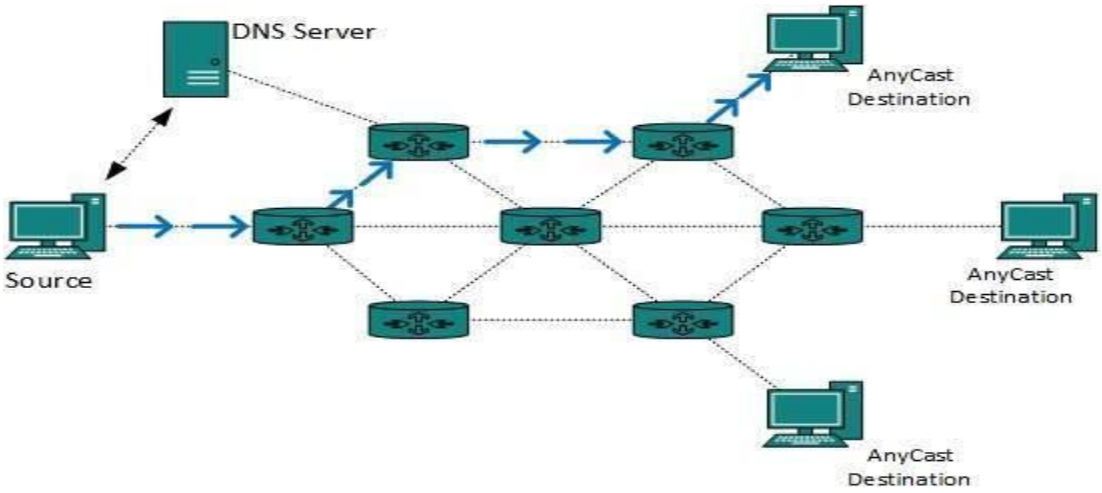


The router must know that there are nodes, which wish to receive multicast packets (or stream) then only it should forward. Multicast routing works spanning tree protocol to avoid looping.

Multicast routing also uses reverse path Forwarding technique, to detect and discard duplicates and loops.

Anycast Routing :

Anycast packet forwarding is a mechanism where multiple hosts can have same logical address. When a packet destined to this logical address is received, it is sent to the host which is nearest in routing topology.



Anycast routing is done with help of DNS server. Whenever an Anycast packet is received it is enquired with DNS to where to send it. DNS provides the IP address which is the nearest IP configured on it.

Unicast Routing Protocols :

There are two kinds of routing protocols available to route unicast packets:

- **Distance Vector Routing Protocol**

  Distance Vector is simple routing protocol which takes routing decision on the number of hops between source and destination. A route with less number of hops is considered as the best route. Every router advertises its set best routes to other routers. Ultimately, all routers build up their network topology based on the advertisements of their peer routers,

  For example Routing Information Protocol (RIP).

- **Link State Routing Protocol**

  Link State protocol is slightly complicated protocol than Distance Vector. It takes into account the states of links of all the routers in a network. This technique helps routes build a common graph of the entire network. All routers then calculate their best path for routing purposes.for example, Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (ISIS).

Multicast Routing Protocols :

Unicast routing protocols use graphs while Multicast routing protocols use trees, i.e. spanning tree to avoid loops. The optimal tree is called shortest path spanning tree.

- **DVMRP** - Distance Vector Multicast Routing Protocol

- **MOSPF** - Multicast Open Shortest Path First
- **CBT** - Core Based Tree
- **PIM** - Protocol independent Multicast

Protocol Independent Multicast is commonly used now. It has two flavors:

- **PIM Dense Mode**

  This mode uses source-based trees. It is used in dense environment such as LAN.

- **PIM Sparse Mode**

  This mode uses shared trees. It is used in sparse environment such as WAN.

Routing Algorithms:

The routing algorithms are as follows:

Flooding

Flooding is simplest method packet forwarding. When a packet is received, the routers send it to all the interfaces except the one on which it was received. This creates too much burden on the network and lots of duplicate packets wandering in the network.

Time to Live (TTL) can be used to avoid infinite looping of packets. There exists another approach for flooding, which is called Selective Flooding to reduce the overhead on the network. In this method, the router does not flood out on all the interfaces, but selective ones.

Shortest Path

Routing decision in networks, are mostly taken on the basis of cost between source and destination. Hop count plays major role here. Shortest path is a technique which uses various algorithms to decide a path with minimum number of hops.

Common shortest path algorithms are:

- Dijkstra's algorithm
- Bellman Ford algorithm
- Floyd Warshall algorithms

CIDR (Classless Inter Domain Routing):

CIDR was introduced in 1993 replacing the previous generation of IP address syntax – classful networks. CIDR allowed for more efficient use of IPv4 address space and prefix aggregation, known as route summarization or supernetting.
CIDR allows routers to group routes together to reduce the bulk of routing information carried by core routers. With CIDR, IP addresses and their subnet mask are written as four octets, separated by periods, followed by a forward slash (/) and a two digit number that represents the network mask.

- e.g. 10.1.1.0/30
- 172.16.1.16/28
- 192.168.1.32/27

Unidirectional Link Routing:

UDLR provides mechanisms for a router to emulate a bidirectional link to enable the routing of unicast and multicast packets over a physical unidirectional interface, such as a broadcast satellite link. However, there must be a back channel or other path between the routers that share a physical unidirectional link (UDL). A UDLR tunnel is a mechanism for unicast and multicast traffic; Internet Group Management Protocol (IGMP) UDLR and IGMP Proxy are mechanisms for multicast traffic.

Both unicast and multicast routing protocols forward data on interfaces from which they have received routing control information. This model works only on bidirectional links for most existing routing protocols. However, some networks use broadcast satellite links, which are unidirectional. For networks that use broadcast satellite links, accomplishing two-way communication over broadcast satellite links presents a problem in terms of discovering and sharing knowledge of a network topology.

Specifically, in unicast routing, when a router receives an update message on an interface for a prefix, it forwards data for destinations that match that prefix out that same interface. This is the case in distance vector routing protocols. Similarly, in multicast routing, when a router receives a join message for a multicast group on an interface, it forwards copies of data destined for that group out that same interface. Based on these principles, existing unicast and multicast routing protocols cannot be supported over UDLs. UDLR is designed to enable the operation of routing protocols over UDLs without changing the routing protocols themselves.

UDLR enables a router to emulate the behavior of a bidirectional link for IP operations over UDLs. UDLR has three complementary mechanisms for bidirectional link emulation, which are described in the following sections:

- UDLR Tunnel
- IGMP UDLR
- IGMP Proxy

Chapter -2

Next Generation Internet

IPv6 – Overview:

Internet Protocol version 6 is a new addressing protocol designed to incorporate all the possible requirements of future Internet known to us as Internet version 2. This protocol as its predecessor IPv4, works on the Network Layer (Layer-3). Along with its offering of an enormous amount of logical address space, this protocol has ample features to which address the shortcoming of IPv4.

Brief History:

After IPv4's development in the early 80s, the available IPv4 address pool begun to shrink rapidly as the demand of addresses exponentially increased with Internet. Taking pre-cognizance of the situation that might arise, IETF, in 1994, initiated the development of an addressing protocol to replace IPv4. The progress of IPv6 can be tracked by means of the RFC published:

- 1998 – RFC 2460 – Basic Protocol
- 2003 – RFC 2553 – Basic Socket API
- 2003 – RFC 3315 – DHCPv6
- 2004 – RFC 3775 – Mobile IPv6
- 2004 – RFC 3697 – Flow Label Specification
- 2006 – RFC 4291 – Address architecture (revision)
- 2006 – RFC 4294 – Node requirement

On June 06, 2012, some of the Internet giants chose to put their Servers on IPv6. Presently they are using Dual Stack mechanism to implement IPv6 parallel in with IPv4.

IPv6 – Features:

The successor of IPv4 is not designed to be backward compatible. Trying to keep the basic functionalities of IP addressing, IPv6 is redesigned entirely. It offers the following features:

- **Larger Address Space**

  In contrast to IPv4, IPv6 uses 4 times more bits to address a device on the Internet. This much of extra bits can provide approximately 3.4×1038 different combinations of

addresses. This address can accumulate the aggressive requirement of address allotment for almost everything in this world. According to an estimate, 1564 addresses can be allocated to every square meter of this earth.

- **Simplified Header**

  IPv6's header has been simplified by moving all unnecessary information and options (which are present in IPv4 header) to the end of the IPv6 header. IPv6 header is only twice as bigger than IPv4 provided the fact that IPv6 address is four times longer.

- **End-to-end Connectivity**

  Every system now has unique IP address and can traverse through the Internet without using NAT or other translating components. After IPv6 is fully implemented, every host can directly reach other hosts on the Internet, with some limitations involved like Firewall, organization policies, etc.

- **Auto-configuration**

  IPv6 supports both stateful and stateless auto configuration mode of its host devices. This way, absence of a DHCP server does not put a halt on inter segment communication.

- **Faster Forwarding/Routing**

  Simplified header puts all unnecessary information at the end of the header. The information contained in the first part of the header is adequate for a Router to take routing decisions, thus making routing decision as quickly as looking at the mandatory header.

- **IPSec**

  Initially it was decided that IPv6 must have IPSec security, making it more secure than IPv4. This feature has now been made optional.

- **No Broadcast**

  Though Ethernet/Token Ring are considered as broadcast network because they support Broadcasting, IPv6 does not have any broadcast support any more. It uses multicast to communicate with multiple hosts.

- **Anycast Support**

This is another characteristic of IPv6. IPv6 has introduced Anycast mode of packet routing. In this mode, multiple interfaces over the Internet are assigned same Anycast IP address. Routers, while routing, send the packet to the nearest destination.

- **Mobility**

  IPv6 was designed keeping mobility in mind. This feature enables hosts (such as mobile phone) to roam around in different geographical area and remain connected with the same IP address. The mobility feature of IPv6 takes advantage of auto IP configuration and Extension headers.

- **Enhanced Priority Support**

  IPv4 used 6 bits DSCP (Differential Service Code Point) and 2 bits ECN (Explicit Congestion Notification) to provide Quality of Service but it could only be used if the end-to-end devices support it, that is, the source and destination device and underlying network must support it.

  In IPv6, Traffic class and Flow label are used to tell the underlying routers how to efficiently process the packet and route it.

- **Smooth Transition**

  Large IP address scheme in IPv6 enables to allocate devices with globally unique IP addresses. This mechanism saves IP addresses and NAT is not required. So devices can send/receive data among each other, for example, VoIP and/or any streaming media can be used much efficiently.

  Other fact is, the header is less loaded, so routers can take forwarding decisions and forward them as quickly as they arrive.

- **Extensibility**

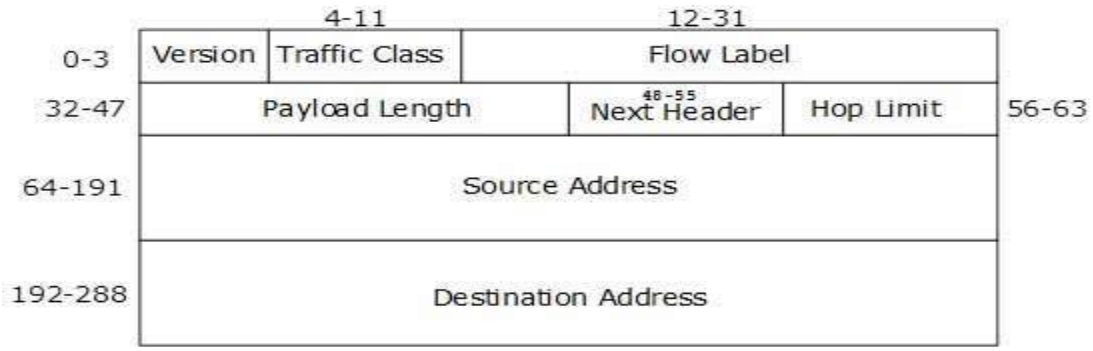
  One of the major advantages of IPv6 header is that it is extensible to add more information in the option part. IPv4 provides only 40-bytes for options, whereas options in IPv6 can be as much as the size of IPv6 packet itself.

IPv6 – Headers:

The wonder of IPv6 lies in its header. An IPv6 address is 4 times larger than IPv4, but surprisingly, the header of an IPv6 address is only 2 times larger than that of IPv4. IPv6 headers have one Fixed Header and zero or more Optional (Extension) Headers. All the necessary

information that is essential for a router is kept in the Fixed Header. The Extension Header contains optional information that helps routers to understand how to handle a packet/flow.

Fixed Header



IPv6 fixed header is 40 bytes long and contains the following information.

S.N.	Field & Description
1	**Version** (4-bits): It represents the version of Internet Protocol, i.e. 0110.
2	**Traffic Class** (8-bits): These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router Known what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN).
3	**Flow Label** (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. This field helps avoid re-ordering of data packets. It is designed for streaming/real-time media.
4	**Payload Length** (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. With 16 bits, up to 65535 bytes can be

	indicated; but if the Extension Headers contain Hop-by-Hop Extension Header, then the payload may exceed 65535 bytes and this field is set to 0.
5	**Next Header** (8-bits): This field is used to indicate either the type of Extension Header, or if the Extension Header is not present then it indicates the Upper Layer PDU. The values for the type of Upper Layer PDU are same as IPv4's.
6	**Hop Limit** (8-bits): This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0 the packet is discarded.
7	**Source Address** (128-bits): This field indicates the address of originator of the packet.
8	**Destination Address** (128-bits): This field provides the address of intended recipient of the packet.

Extension Headers:

In IPv6, the Fixed Header contains only that much information which is necessary, avoiding those information which is either not required or is rarely used. All such information is put between the Fixed Header and the Upper layer header in the form of Extension Headers. Each Extension Header is identified by a distinct value.

When Extension Headers are used, IPv6 Fixed Header's Next Header field points to the first Extension Header. If there is one more Extension Header, then the first Extension Header's 'Next-Header' field points to the second one, and so on. The last Extension Header's 'Next-Header' field points to the Upper Layer Header. Thus, all the headers points to the next one in a linked list manner.

If the Next Header field contains the value 59, it indicates that there are no headers after this header, not even Upper Layer Header.

The following Extension Headers must be supported as per RFC 2460:

Extension Header	Next Header Value	Description
Hop-by-Hop Options header	0	read by all devices in transit network
Routing header	43	contains methods to support making routing decision
Fragment header	44	contains parameters of datagram fragmentation
Destination Options header	60	read by destination devices
Authentication header	51	information regarding authenticity
Encapsulating Security Payload header	50	encryption information

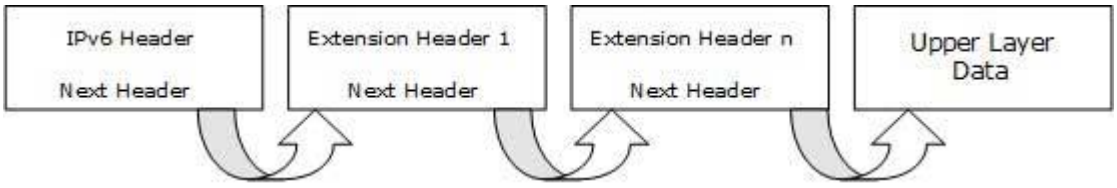
The sequence of Extension Headers should be:

IPv6 header
Hop-by-Hop Options header
Destination Options header[1]
Routing header
Fragment header
Authentication header
Encapsulating Security Payload header
Destination Options header[2]
Upper-layer header

These headers:

- Should be processed by First and subsequent destinations.
- Should be processed by Final Destination.

Extension Headers are arranged one after another in a linked list manner, as depicted in the following diagram:

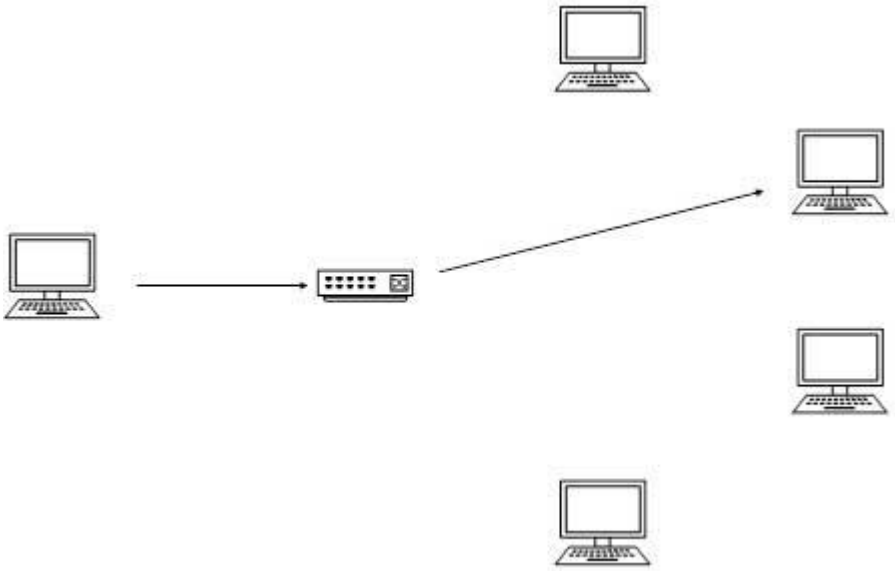


IPv6 - Addressing Modes:

In computer networking, addressing mode refers to the mechanism of hosting an address on the network. IPv6 offers several types of modes by which a single host can be addressed. More than one host can be addressed at once or the host at the closest distance can be addressed.

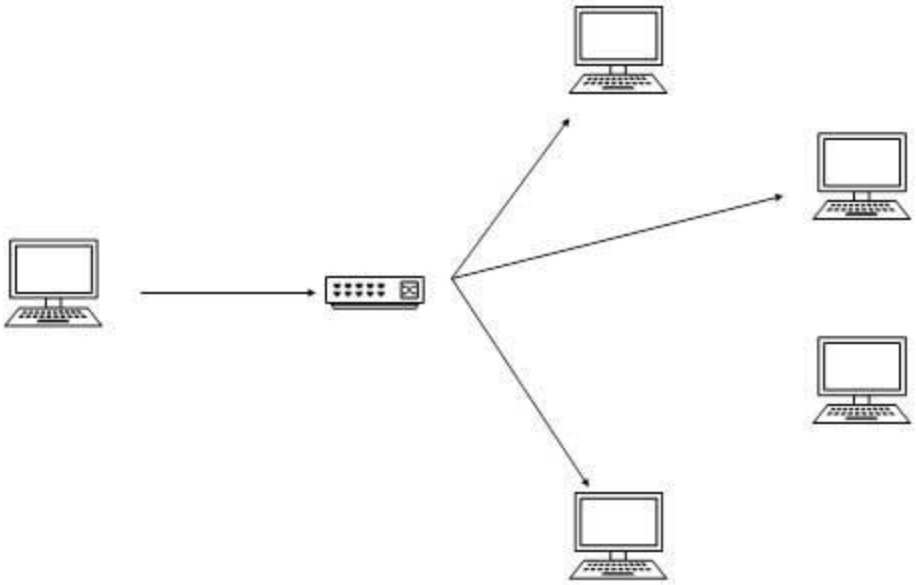
Unicast:

In unicast mode of addressing, an IPv6 interface (host) is uniquely identified in a network segment. The IPv6 packet contains both source and destination IP addresses. A host interface is equipped with an IP address which is unique in that network segment. When a network switch or a router receives a unicast IP packet, destined to a single host, it sends out one of its outgoing interface which connects to that particular host.



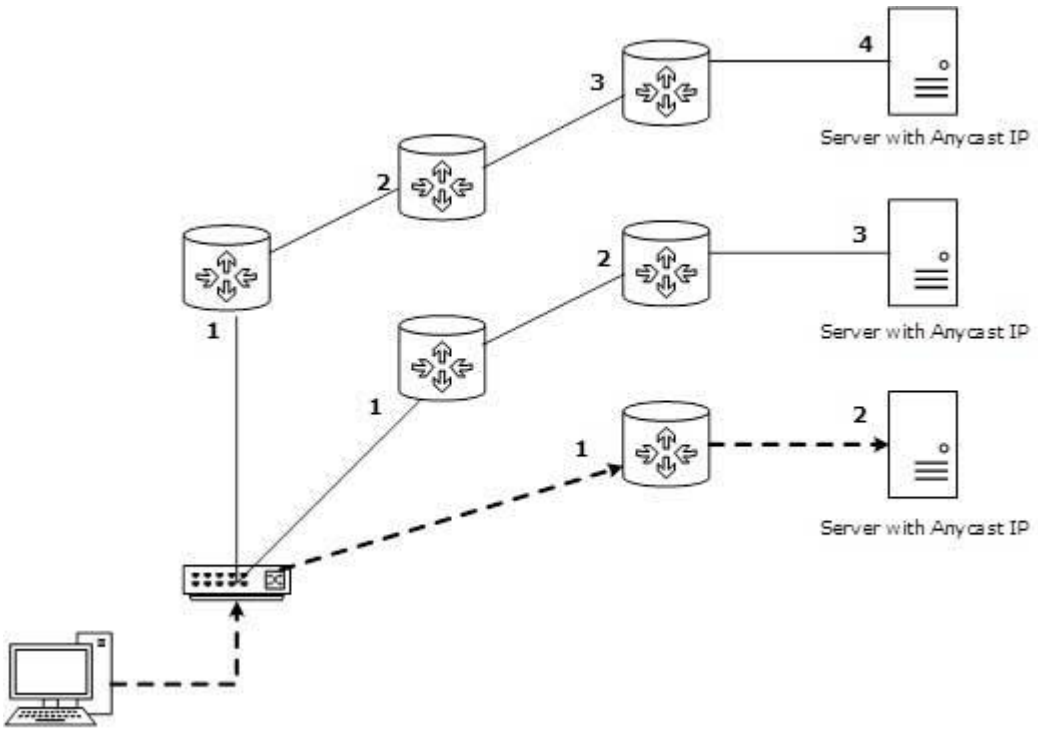
Multicast:

The IPv6 multicast mode is same as that of IPv4. The packet destined to multiple hosts is sent on a special multicast address. All the hosts interested in that multicast information, need to join that multicast group first. All the interfaces that joined the group receive the multicast packet and process it, while other hosts not interested in multicast packets ignore the multicast information.



Any cast:

IPv6 has introduced a new type of addressing, which is called Anycast addressing. In this addressing mode, multiple interfaces (hosts) are assigned same Anycast IP address. When a host wishes to communicate with a host equipped with an Anycast IP address, it sends a Unicast message. With the help of complex routing mechanism, that Unicast message is delivered to the host closest to the Sender in terms of Routing cost.



Let's take an example of TutorialPoints.com Web Servers, located in all continents. Assume that all the Web Servers are assigned a single IPv6 anycast IP Address. Now when a user from Europe wants to reach TutorialsPoint.com the DNS points to the server that is physically located in Europe itself. If a user from India tries to reach Tutorialspoint.com, the DNS will then point to the Web Server physically located in Asia. Nearest or Closest terms are used in terms of Routing Cost.

In the above picture, when a client computer tries to reach a server, the request is forwarded to the server with the lowest Routing Cost.

Address Structure :

An IPv6 address is made of 128 bits divided into eight 16-bits blocks. Each block is then converted into 4-digit Hexadecimal numbers separated by colon symbols.

For example, given below is a 128 bit IPv6 address represented in binary format and divided into eight 16-bits blocks:

```
0010000000000001 0000000000000000 0011001000111000 1101111111100001
0000000001100011 0000000000000000 0000000000000000 1111111011111011
```

Each block is then converted into Hexadecimal and separated by ':' symbol:

2001:0000:3238:DFE1:0063:0000:0000:FEFB

Even after converting into Hexadecimal format, IPv6 address remains long. IPv6 provides some rules to shorten the address. The rules are as follows:

Rule.1: Discard leading Zero(es):

In Block 5, 0063, the leading two 0s can be omitted, such as (5th block):

2001:0000:3238:DFE1:63:0000:0000:FEFB

Rule.2: If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign ::, such as (6th and 7th block):

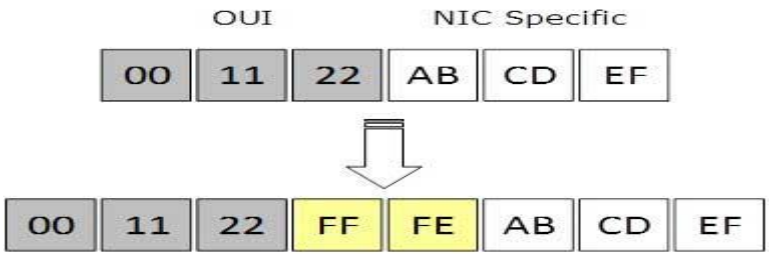
2001:0000:3238:DFE1:63::FEFB

Consecutive blocks of zeroes can be replaced only once by :: so if there are still blocks of zeroes in the address, they can be shrunk down to a single zero, such as (2nd block):

2001:0:3238:DFE1:63::FEFB

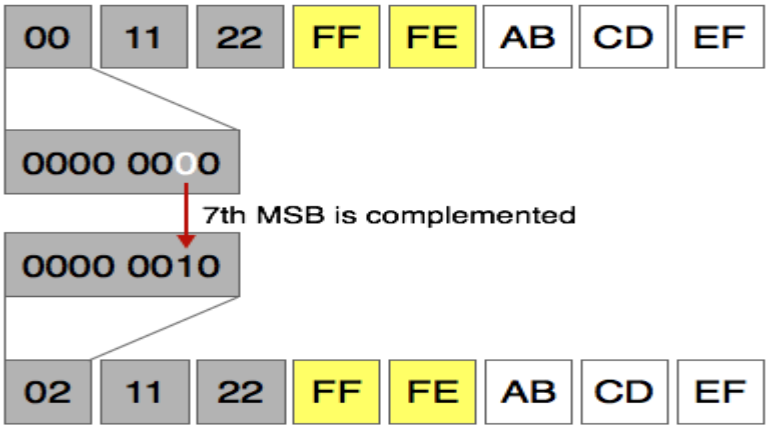
Interface ID

IPv6 has three different types of Unicast Address scheme. The second half of the address (last 64 bits) is always used for Interface ID. The MAC address of a system is composed of 48-bits and represented in Hexadecimal. MAC addresses are considered to be uniquely assigned worldwide. Interface ID takes advantage of this uniqueness of MAC addresses. A host can auto-configure its Interface ID by using IEEE's Extended Unique Identifier (EUI-64) format. First, a host divides its own MAC address into two 24-bits halves. Then 16-bit Hex value 0xFFFE is sandwiched into those two halves of MAC address, resulting in EUI-64 Interface ID.



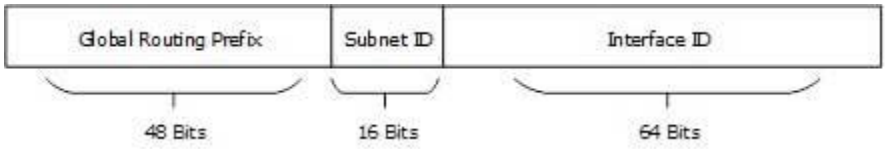
Conversion of EUI-64 ID into IPv6 Interface Identifier

To convert EUI-64 ID into IPv6 Interface Identifier, the most significant 7th bit of EUI-64 ID is complemented. For example:



Global Unicast Address

This address type is equivalent to IPv4's public address. Global Unicast addresses in IPv6 are globally identifiable and uniquely addressable.



Global Routing Prefix: The most significant 48-bits are designated as Global Routing Prefix which is assigned to specific autonomous system. The three most significant bits of Global Routing Prefix is always set to 001.

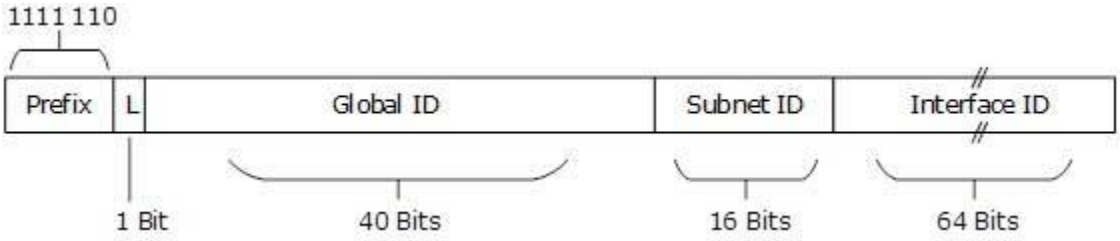
Link-Local Address

Auto-configured IPv6 address is known as Link-Local address. This address always starts with FE80. The first 16 bits of link-local address is always set to 1111 1110 1000 0000 (FE80). The next 48-bits are set to 0, thus:

Link-local addresses are used for communication among IPv6 hosts on a link (broadcast segment) only. These addresses are not routable, so a Router never forwards these addresses outside the link.

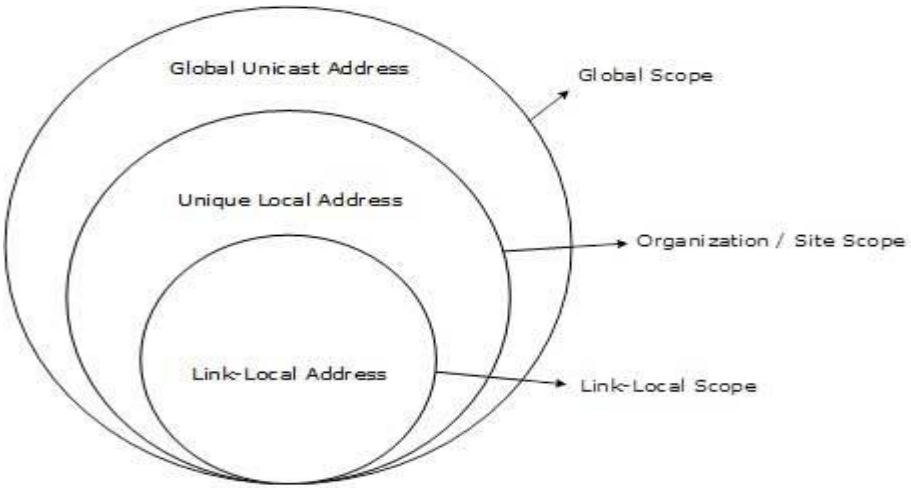
Unique-Local Address

This type of IPv6 address is globally unique, but it should be used in local communication. The second half of this address contain Interface ID and the first half is divided among Prefix, Local Bit, Global ID and Subnet ID.



Prefix is always set to 1111 110. L bit, is set to 1 if the address is locally assigned. So far, the meaning of L bit to 0 is not defined. Therefore, Unique Local IPv6 address always starts with 'FD'.

Scope of IPv6 Unicast Addresses:



The scope of Link-local address is limited to the segment. Unique Local Address are locally global, but are not routed over the Internet, limiting their scope to an organization's boundary. Global Unicast addresses are globally unique and recognizable. They shall make the essence of Internet v2 addressing.

IPv6 - Special Addresses

Version 6 has slightly complex structure of IP address than that of IPv4. IPv6 has reserved a few addresses and address notations for special purposes. See the table below:

IPv6 Address	Meaning
::/128	Unspecified Address
::/0	Default Route
::1/128	Loopback Address

- As shown in the table, the address 0:0:0:0:0:0:0:0/128 does not specify anything and is said to be an unspecified address. After simplifying, all the 0s are compacted to ::/128.
- In IPv4, the address 0.0.0.0 with netmask 0.0.0.0 represents the default route. The same concept is also applied to IPv6, address 0:0:0:0:0:0:0:0 with netmask all 0s represents the default route. After applying IPv6 rule, this address is compressed to ::/0.
- Loopback addresses in IPv4 are represented by 127.0.0.1 to 127.255.255.255 series. But in IPv6, only 0:0:0:0:0:0:0:1/128 represents the Loopback address. After loopback address, it can be represented as ::1/128.

Reserved Multicast Address for Routing Protocols

IPv6 Address	Routing Protocol
FF02::5	OSPFv3
FF02::6	OSPFv3 Designated Routers
FF02::9	RIPng
FF02::A	EIGRP

- The above table shows the reserved multicast addresses used by interior routing protocol.
- The addresses are reserved following the same rules of IPv4.

Reserved Multicast Address for Routers/Node

IPv6 Address	Scope
FF01::1	All Nodes in interface-local
FF01::2	All Routers in interface local
FF02::1	All Nodes in link-local
FF02::2	All Routers in link-local
FF05::2	All Routers in site-local

- These addresses help routers and hosts to speak to available routers and hosts on a segment without being configured with an IPv6 address. Hosts use EUI-64 based auto-configuration to self-configure an IPv6 address and then speak to available hosts/routers on the segment by means of these addresses.

Chapter – 3

ICMPv6 and Neighbor Discovery

ICMPv6 Introduction:

The IP protocol alone provides no direct way to do the following:

- For an end system to learn the fate of IP packets that fail to make it to their destinations.
- For obtaining diagnostic information (e.g., which routers are used along a path or a method to estimate the round-trip time).

To address these deficiencies, a special protocol called the Internet Control Message Protocol (ICMP) is used in conjunction with IP to provide diagnostics and control information related to the configuration of the IP protocol layer and the disposition of IP packets.

ICMP provides for the delivery of error and control messages that may require attention. ICMP messages are usually acted on by:

- The IP layer itself,
- Higher-layer transport protocols (TCP or UDP),
- User applications.

ICMP does not provide reliability for IP; it indicates certain classes of failures and configuration information. The most common cause of packet drops (buffer overrun at a router) does not elicit any ICMP information. Other protocols, such as TCP, handle such situations.

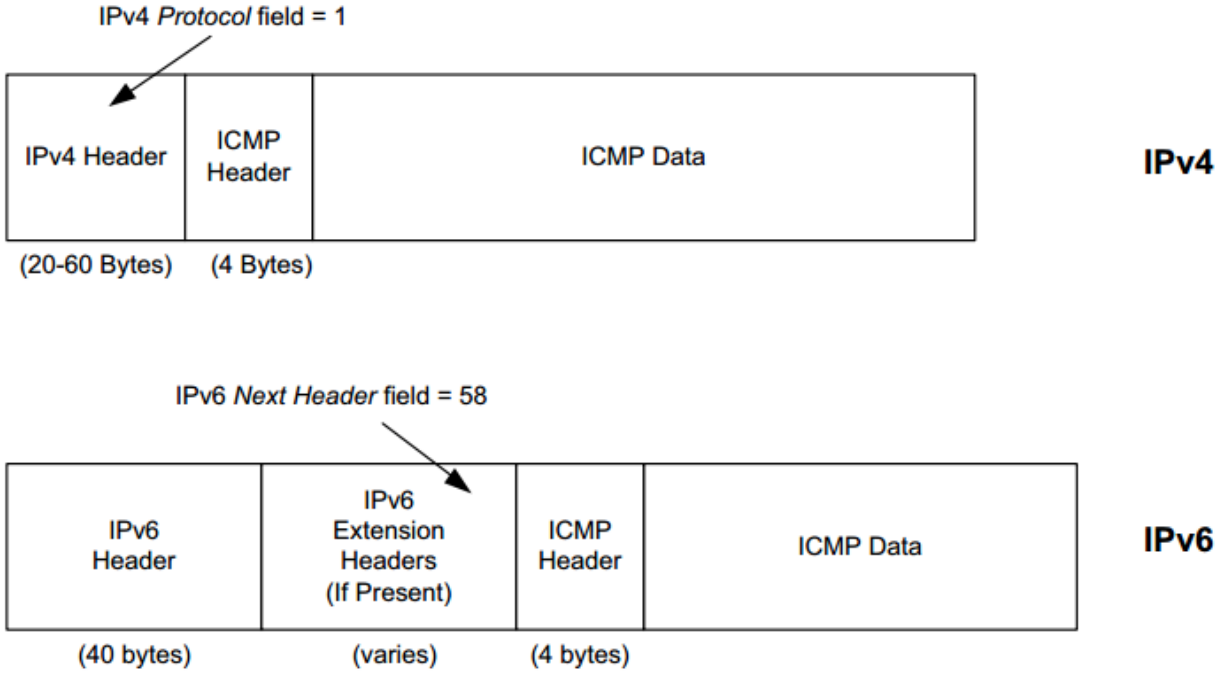
Because of the ability of ICMP to affect the operation of important system functions and obtain configuration information, hackers have used ICMP messages in a large number of attacks. As a result of concerns about such attacks, network administrators often arrange to block ICMP messages with firewalls, especially at border routers. If ICMP is blocked, however, a number of common diagnostic utilities (e.g., ping, trace route) do not work properly.

The term ICMP refers to ICMP in general, and the terms ICMPv4 and ICMPv6 to refer specifically to the versions of ICMP used with IPv4 and IPv6, respectively. ICMPv6 plays a far more important role in the operation of IPv6 than ICMPv4 does for IPv4.

In IPv6, ICMPv6 is used for several purposes beyond simple error reporting and signaling. It is used for:

- **Neighbor Discovery** (ND), which plays the same role as ARP does for IPv4
- **Router Discovery** function used for configuring hosts and multicast address management
- Managing hand-offs in Mobile IPv6.

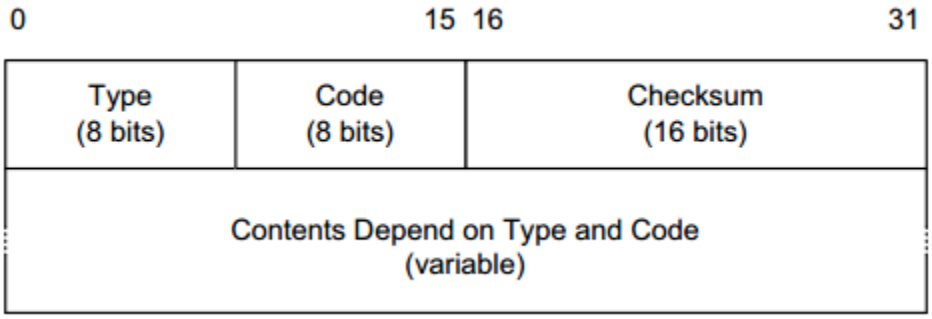
Encapsulation in IPv4 and IPv6



- In IPv4, a Protocol field value of 1 indicates that the datagram caries ICMPv4.
- In IPv6, the ICMPv6 message may begin after zero or more extension headers. The last extension header before the ICMPv6 header includes a Next Header field with value 58.

ICMP messages may be fragmented like other IP datagrams, although this is not common.

The following figure shows the format of both ICMPv4 and ICMPv6 messages. The first 4 bytes have the same format for all messages, but the remainder differ from one message to the next.



In ICMPv4:

- 42 different values are reserved for the **Type** field [], which identify the particular message. Only about 8 of these are in regular use.
- Many types of ICMP messages also use different values of the **Code** field to further specify the meaning of the message.

- The **Checksum** field covers the entire ICMPv4 message; in ICMPv6 it also covers a pseudo-header derived from portions of the IPv6 header.

This is our first example of an *end-to-end* checksum. It is carried all the way from the sender of the ICMP message to the final recipient. In contrast, the IPv4 header checksum is changed at every router hop. If an ICMP implementation receives an ICMP message with a bad checksum, the message is discarded; there is no ICMP message to indicate a bad checksum in a received ICMP message. Recall that the IP layer has no protection on the payload portion of the datagram. If ICMP did not include a checksum, the contents of the ICMP message might not be correct, leading to incorrect system behavior.

ICMPv4 Messages

For ICMPv4, the informational messages include:

- Echo Request (type 8)
- Echo Reply (type 0)
- Router Advertisement (type 9)
- Router Solicitation (type 10)

Router Advertisement and Router Solicitation are together called Router Discovery.

The most common error message types are:

- Destination Unreachable (type 3)
- Redirect (type 5)
- Time Exceeded (type 11)
- Parameter Problem (type 12)

ICMPv6 Messages

ICMPv6 is responsible not only for error and informational messages but also for a great deal of IPv6 router and host configuration.

In ICMPv6, as in ICMPv4, messages are grouped into the informational and error classes. In ICMPv6, however, all the error messages have a 0 in the high-order bit of the **Type** field. Thus, ICMPv6 types 0 through 127 are all errors, and types 128 through 255 are all informational. Many of the informational messages are request/reply pairs.

In comparing the common ICMPv4 messages with the ICMPv6 standard messages, we conclude that some of the effort in designing ICMPv6 was to eliminate the unused messages from the original specification while retaining the useful ones. Following this approach, ICMPv6 also makes use of the **Code** field, primarily to refine the meanings of certain error messages.

In addition to the Type and Code fields that define basic functions in ICMPv6, a large number of standard options are also supported, some of which are required. This distinguishes ICMPv6 from ICMPv4 (ICMPv4 does not have options).

Processing of ICMP Messages

In ICMP, the processing of incoming messages varies from system to system. Generally:

- Incoming informational requests are handled automatically by the operating system.
- Error messages are delivered to user processes or to a transport protocol such as TCP.

The processes may choose to act on them or ignore them. Exceptions to this general rule include:

- The Redirect message. This results in an automatic update to the host's routing table.
- The Destination Unreachable—Fragmentation required messages. This is used in the path MTU discovery (PMTUD) mechanism, which is generally implemented by the transport-layer protocols such as TCP.

In ICMPv6, the following rules are applied when processing incoming ICMPv6 messages:

1. Unknown ICMPv6 error messages must be passed to the upper-layer process that produced the datagram causing the error (if possible).
2. Unknown ICMPv6 informational messages are dropped.
3. ICMPv6 error messages include as much of the original ("offending") IPv6 datagram that caused the error as will fit without making the error message datagram exceed the minimum IPv6 MTU (1280 bytes).
4. When processing ICMPv6 error messages, the upper-layer protocol type is extracted from the original or "offending" packet (contained in the body of the ICMPv6 error message) and used to select the appropriate upper-layer process. If this is not possible, the error message is silently dropped after any IPv6-layer processing.
5. There are special rules for handling errors.
6. An IPv6 node must limit the rate of ICMPv6 error messages it sends. There are a variety of ways of implementing the rate-limiting

ICMP Error Messages

The distinction between the error and informational classes of ICMP messages is important. An ICMP error message is not to be sent in response to any of the following messages:

- Another ICMP error message,
- Datagrams with bad headers (e.g., bad checksum),
- IP-layer broadcast/multicast datagrams,
- Datagrams encapsulated in link-layer broadcast or multicast frames,
- Datagrams with an invalid or network zero source address,
- Any fragment other than the first.

The reason for imposing these restrictions on the generation of ICMP errors is to limit the creation of so-called broadcast storms, a scenario in which the generation of a small number of messages creates an unwanted traffic cascade (e.g., by generating error responses in response to error responses, indefinitely). These rules can be summarized as follows:

ICMPv4 error messages

An ICMPv4 error message is never generated in response to:

- An ICMPv4 error message. (An ICMPv4 error message may, however, be generated in response to an ICMPv4 query message.)
- A datagram destined for an IPv4 broadcast address or an IPv4 multicast address (formerly known as a class D address).
- A datagram sent as a link-layer broadcast.
- A fragment other than the first.
- A datagram whose source address does not define a single host. This means that the source address cannot be any of the following:
    - Zero address,
    - Loopback address,
    - Broadcast address,
    - Multicast address.

ICMPv6 error messages

ICMPv6 error messages are used to report errors in the forwarding or delivery of IPv6 packets. The ICMPv6 "Type field" values for the error message are between 0 and 127. ICMPv6 error messages are Destination Unreachable, Packet Too Big, Time Exceeded, and Parameter Problem.

- ✓ **"Destination Unreachable" ICMPv6 error message:** Destination Unreachable ICMPv6 error message is generated by the source host or a router when an IPv6 datagram packet cannot be delivered for any reason other than congestion.
- ✓ **"Packet Too Big" ICMPv6 error message:** "Packet Too Big" ICMPv6 error messages are generated by the router when a packet cannot be forwarded to the next hop link because the size of the IPv6 datagram is larger than the MTU (Maximum Transmission Unit) of the link. "Packet Too Big" ICMPv6 error message includes the MTU (Maximum Transmission Unit) of the next link also. MTU (Maximum Transmission Unit) is the size of the largest protocol data unit that is supported over the link.
- ✓ **"Time Exceeded" ICMPv6 error message:** Similar to the Time-to-Live field value in IPv4 datagram header, IPv6 header includes a Hop Limit field. The Hop Limit field value in IPv6 header is used to prevent routing loops. Hop Limit field in IPv6 datagram header is decremented by each router that forwards the packet. When the Hop Limit field value in IPv6 header reaches zero, the router discards the IPv6 datagram packet and returns a "Time Exceeded" ICMPv6 error message to the source host.
- ✓ **"Parameter Problem" ICMPv6 error message:** "Parameter Problem" ICMPv6 error message is typically related with the problems and mistakes related with IPv6 header itself. When a problem or mistake with an IPv6 header make a router cannot process the packet, the router stops processing the IPv6 datagram packet, discards the packet and returns a "Parameter Problem" ICMPv6 error message to the source host.

ICMPv6 informational messages

ICMPv6 informational messages are used for network diagnostic functions and additional critical network functions like Neighbor Discovery, Router Solicitation & Router Advertisements, and Multicast Memberships. Echo Request and Echo Reply (used by many commands and utilities like "ping" for network diagnostics and communication trouble shooting) are also ICMPv6 informational messages. The ICMPv6 informational messages have values for the Type field (8 bit binary number) between 128 and 255.

- ✓ **Diagnostic Messages:** ICMPv6 Echo request and Echo reply are the Diagnostic messages. Every IPv6 host must return an ICMPv6 Echo reply when it receives an ICMPv6 Echo request. Echo request and Echo reply messages are used by the ping command to check the network connectivity between two IPv6 hosts.
- ✓ **MLD (Multicast Listener Discovery) Messages:** ICMPv6 MLD (Multicast Listener Discovery) Messages are used by an IPv6 enabled router to discover hosts who are interested in multicast packets, and the multicast addresses they are interested. MLD (Multicast Listener Discovery) messages are used by MLD (Multicast Listener Discovery) Protocol. MLD (Multicast Listener Discovery) Protocol is the IPv6 equivalent of IGMP (Internet Group Management) Protocol in IPv4.
- ✓ **ND (Neighbor Discovery) Messages:** ICMPv6 ND (Neighbor Discovery) Messages are used for the Neighbor Discovery Protocol (NDP). ND (Neighbor Discovery) Messages includes Router Solicitation & Router Advertisement, Neighbor Solicitation and Neighbor Advertisement.

Rate-limiting ICMP messages with token buckets

In addition to the rules governing the conditions under which ICMP messages are generated, there is also a rule that limits the overall ICMP traffic level from a single sender. In [RFC4443], a recommendation for rate-limiting ICMP messages is to use a token bucket.

With a token bucket, a "bucket" holds a maximum number (*B*) of "tokens", each of which allows a certain number of messages to be sent. The bucket is periodically filled with new tokens (at rate *N*) and drained by 1 for each message sent. Thus, a token bucket (or **token bucket filter**, as it is often called) is characterized by the parameters (*B*, *N*). For small or midsize devices, [RFC4443] provides an example token bucket using the parameters (10, 10). Token buckets are a common mechanism used in protocol implementations to limit bandwidth utilization, and in many cases *B* and *N* are in byte units rather than message units.

Copy of offending datagram headers in ICMP error message

When an ICMP error message is sent, it contains a copy of the full IP header from the "offending" or "original" datagram (i.e., the IP header of the datagram that caused the error to be generated, including any IP options), plus any other data from the original datagram's IP payload area such that the generated IP/ ICMP datagram's size does not exceed a specific value. For IPv4 this value is 576 bytes, and for IPv6 it is the IPv6 minimum MTU, which is at least 1280 bytes.

Including a portion of the payload from the original IP datagram lets the receiving ICMP module associate the message with one particular protocol (e.g., TCP or UDP) from the Protocol or Next Header field in the IP header and one particular user process (from the TCP or UDP port numbers that are in the TCP or UDP header contained in the first 8 bytes of the IP datagram payload area).

Destination Unreachable (ICMPv4 Type 3, ICMPv6 Type 1) and Packet Too Big (ICMPv6 Type 2)

Messages of this type are used to indicate that a datagram could not be delivered all the way to its destination because of either a problem in transit or the lack of a receiver interested in receiving it. Although 16 different codes are defined for this message in ICMPv4, only 4 are commonly used. These include:

- Host Unreachable (code 1)
- Port Unreachable (code 3)
- Fragmentation Required/ Don't-Fragment Specified (code 4),
- Communication Administratively Prohibited (code 13).

In ICMPv6, the Destination Unreachable message is type 1 with seven possible code values. In ICMPv6, as compared with IPv4, the Fragmentation required message has been replaced by an entirely different type (type 2), but the usage is very similar to the corresponding ICMP Destination Unreachable message. In ICMPv6 this is called the Packet Too Big (PTB) message. We will use the simpler ICMPv6 PTB terminology from here onward to refer to either the ICMPv4 (type 3, code 4) message or the ICMPv6 (type 2, code 0) message.

ICMPv4 Host Unreachable (Code 1) and ICMPv6 Address Unreachable (Code 3)

This form of the Destination Unreachable message is generated by a router or host when it is required to send an IP datagram to a host using direct delivery but for some reason cannot reach the destination. This situation may arise, for example, because the last-hop router is attempting to send an ARP request to a host that is either missing or down. Which describes ARP.) For ICMPv6, which uses a somewhat different mechanism for detecting unresponsive hosts, this message can be the result of a failure in the ND.

ICMPv4 Port Unreachable (Code 3) and ICMPv6 Port Unreachable (Code 4)

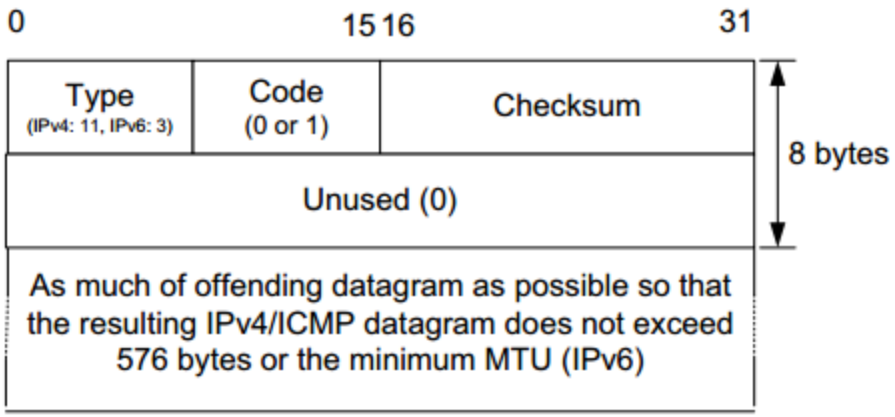
The Port Unreachable message is generated when an incoming datagram is destined for an application that is not ready to receive it. This occurs most commonly in conjunction with UDP, when a message is sent to a port number that is not in use by any server process. If UDP receives a datagram and the destination port does not correspond to a port that some process has in use, UDP responds with an ICMP Port Unreachable message.

ICMP Time Exceeded (ICMPv4 Type 11, ICMPv6 Type 3)

Every IPv4 datagram has a **Time-to-Live** (TTL) field in its IPv4 header, and every IPv6 datagram has a **Hop Limit** field in its header.

As originally conceived, the 8-bit TTL field was to hold the number of seconds a datagram was allowed to remain active in the network before being forcibly discarded (a good thing if forwarding loops are present). Because of an additional rule that said that any router must decrement the TTL field by at least 1, combined with the fact that datagram forwarding times grew to be small fractions of a second, the TTL field has been used in practice as a limitation on the number of hops an IPv4 datagram is allowed to take before it is discarded by a router. This usage was formalized and ultimately adopted in IPv6.

ICMP Time Exceeded (code 0) messages are generated when a router discards a datagram because the TTL or Hop Limit field is too low (i.e., arrives with value 0 or 1 and must be forwarded). This message is important for the proper operation of the trace route tool (called tracert on Windows). Its format, for both ICMPv4 and ICMPv6, is given in the figure below.



Another less common variant of this message is when a fragmented IP datagram only partially arrives at its destination (all its fragments do not arrive after a period of time). In such cases, a variant of the ICMP Time Exceeded message (code 1) is used to inform the sender that its overall datagram has been discarded. Recall that if any fragment of a datagram is dropped, the entire datagram is lost.

Example: The traceroute **Tool**

The traceroute tool is used to determine the routers used along a path from a sender to a destination. This section discusses the operation of the IPv4 version. The approach involves sending datagrams first with an IPv4 TTL field set to 1 and allowing the expiring datagrams to induce routers along the path to send ICMPv4 Time Exceeded (code 0) messages. Each round, the sending TTL value is increased by 1, causing the routers that are one hop farther to expire the datagrams and generate ICMP messages. These messages are sent from the router's primary IPv4 address "facing" the sender.

In this example, traceroute is used to send UDP datagrams from the laptop to the host www.eecs.berkeley.edu. (an Internet host with IPv4 address 128.32.244.172). This is accomplished using the following command:

```
Linux% traceroute –m 2 www.cs.berkeley.edu
traceroute to web2.eecs.berkeley.edu (128.32.244.172), 2 hops max,
52 byte packets
1 gw (192.168.0.1) 3.213 ms 0.839 ms 0.920 ms
2 10.0.0.1 (10.0.0.1) 1.524 ms 1.221 ms 9.176 ms
```

The –m option instructs traceroute to perform only two rounds: one using TTL = 1 and one using TTL = 2. Each line gives the information found at the corresponding TTL. For example, line 1 indicates that one hop away a router with IPv4 address 192.168.0.1 was found and that three independent round-trip-time (RTT) measurements (3.213, 0.839, and 0.920ms) were taken. The difference between the first and subsequent times relates to additional work that is involved in the first measurement (i.e., an ARP transaction).

IPv6 - Communication

In IPv4, a host that wants to communicate with another host on the network needs to have an IP address acquired either by means of DHCP or by manual configuration. As soon as a host is equipped with some valid IP address, it can speak to any host on the subnet. To communicate on layer-3, a host must also know the IP address of the other host. Communication on a link, is established by means of hardware embedded MAC Addresses. To know the MAC address of a host whose IP address is known, a host sends ARP broadcast and in return, the intended host sends back its MAC address.

In IPv6, there are no broadcast mechanisms. It is not a must for an IPv6 enabled host to obtain an IP address from DHCP or manually configured, but it can auto-configure its own IP.

ARP has been replaced by ICMPv6 Neighbor Discovery Protocol.

Neighbor Discovery Protocol

A host in IPv6 network is capable of auto-configuring itself with a unique link-local address. As soon as host gets an IPv6 address, it joins a number of multicast groups. All communications related to that segment take place on those multicast addresses only. A host goes through a series of states in IPv6:

- **Neighbor Solicitation**: After configuring all IPv6's either manually, or by DHCP Server or by auto-configuration, the host sends a Neighbor Solicitation message out to FF02::1/16 multicast address for all its IPv6 addresses in order to know that no one else occupies the same addresses.
- **DAD (Duplicate Address Detection)**: When the host does not listen from anything from the segment regarding its Neighbor Solicitation message, it assumes that no duplicate address exists on the segment.
- **Neighbor Advertisement**: After assigning the addresses to its interfaces and making them up and running, the host once again sends out a Neighbor Advertisement message telling all other hosts on the segment, that it has assigned those IPv6 addresses to its interfaces.

Once a host is done with the configuration of its IPv6 addresses, it does the following things:

- **Router Solicitation**: A host sends a Router Solicitation multicast packet (FF02::2/16) out on its segment to know the presence of any router on this segment. It helps the host to configure the router as its default gateway. If its default gateway router goes down, the host can shift to a new router and makes it the default gateway.

- **Router Advertisement**: When a router receives a Router Solicitation message, it response back to the host, advertising its presence on that link.
- **Redirect**: This may be the situation where a Router receives a Router Solicitation request but it knows that it is not the best gateway for the host. In this situation, the router sends back a Redirect message telling the host that there is a better 'next-hop' router available. Next-hop is where the host will send its data destined to a host which does not belong to the same segment.

IPv6 Neighbor Discovery and Messages

The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and track neighboring devices.

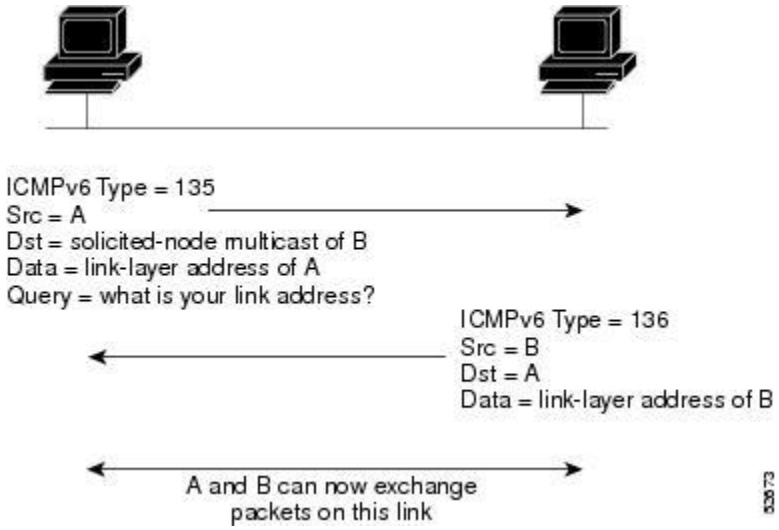
The IPv6 static cache entry for neighbor discovery feature allows static entries to be made in the IPv6 neighbor cache. Static routing requires an administrator to manually enter IPv6 addresses, subnet masks, gateways, and corresponding Media Access Control (MAC) addresses for each interface of each device into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

- IPv6 Neighbor Solicitation Message
- IPv6 Router Advertisement Message
- IPv6 Neighbor Redirect Message

IPv6 Neighbor Solicitation Message

A value of 135 in the Type field of the ICMP packet header identifies a neighbor solicitation message. Neighbor solicitation messages are sent on the local link when a node wants to determine the link-layer address of another node on the same local link. When a node wants to determine the link-layer address of another node, the source address in a neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The destination address in the neighbor solicitation message is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

IPv6 Neighbor Discovery: Neighbor Solicitation Message

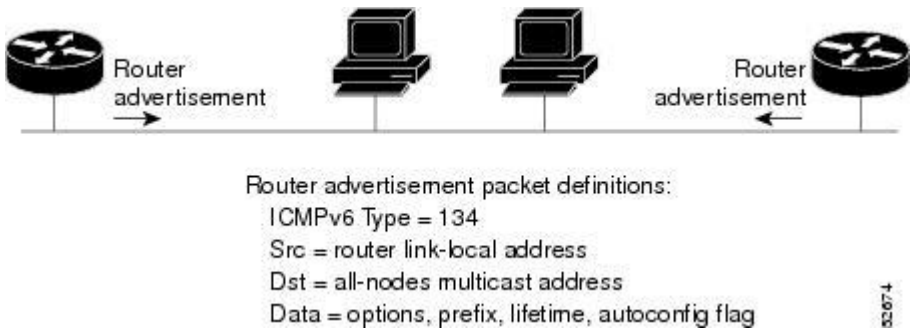


After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node (more specifically, the IPv6 address of the node interface) sending the neighbor advertisement message. The destination address in the neighbor advertisement message is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor advertisement message includes the link-layer address of the node sending the neighbor advertisement message.

IPv6 Router Advertisement Message

Router advertisement (RA) messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out each configured interface of an IPv6 device. For stateless auto configuration to work properly, the advertised prefix length in RA messages must always be 64 bits. The RA messages are sent to the all-nodes multicast address.

IPv6 Neighbor Discovery: RA Message



RA messages typically include the following information:

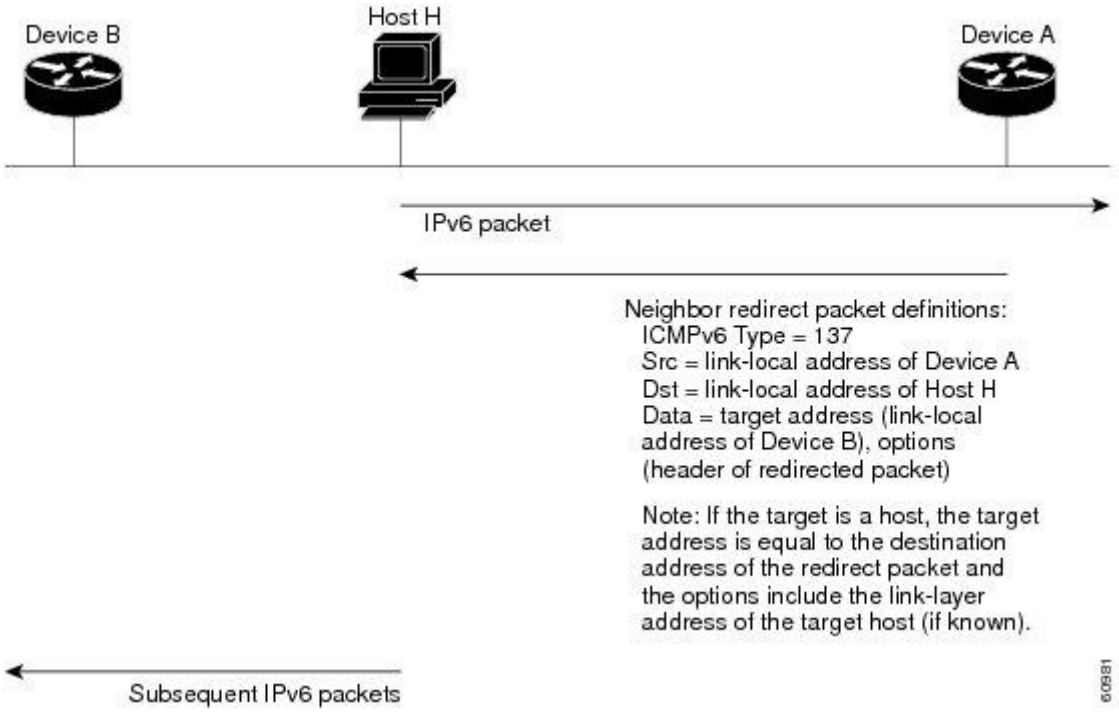
- ✓ One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- ✓ Lifetime information for each prefix included in the advertisement
- ✓ Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed
- ✓ Default device information (whether the device sending the advertisement should be used as a default device and, if so, the amount of time, in seconds, the device should be used as a default device)
- ✓ Additional information for hosts, such as the hop limit and maximum transmission unit (MTU) a host should use in packets that it originates

RAs are also sent in response to device solicitation messages. Device solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message.

IPv6 Neighbor Redirect Message

A value of 137 in the type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Devices send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination.

IPv6 Neighbor Discovery: Neighbor Redirect Message



After forwarding a packet, a device should send a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address.
- The packet was not addressed to the device.
- The packet is about to be sent out the interface on which it was received.
- The device determines that a better first-hop node for the packet resides on the same link as the source of the packet.
- The source address of the packet is a global IPv6 address of a neighbor on the same link, or a link-local address.

Use the **ipv6 icmp error-interval** command to limit the rate at which the device generates all IPv6 ICMP error messages, including neighbor redirect messages, which ultimately reduces link-layer congestion.

IPv6 MTU Path Discovery:

As in IPv4, path MTU discovery in IPv6 allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet

fragmentation saves IPv6 device processing resources and helps IPv6 networks run more efficiently.

With IPv6 path MTU discovery, a device originating IPv6 traffic has an MTU cache that contains MTU values received in ICMPv6 "toobig" messages. In order to prevent an attacker from filling the MTU cache, the device keeps track of the destinations to which it has originated (sent) traffic, and only accepts toobig ICMPv6 messages that have an inner destination matching one of these tracked destinations.

If a malicious device can learn to which destination the device is originating traffic, it could still send a toobig ICMPv6 message to the device for this destination, even if the attacker is not on the path to this destination, and succeeds in forcing his entry into the MTU cache. The device then starts fragmenting traffic to this destination, which significantly affects device performance.

Enabling flow-label marking for locally generated traffic can mitigate this attack. Originated packets are marked with a flow label (which is randomly generated and changed every minute), and toobig messages received are checked against the values sent. Unless an attacker can snoop traffic, the attacker will not know which flow label to use, and its toobig message will be dropped.

MLD Overview:

The Multicast Listener Discovery (MLD) Protocol manages the membership of hosts and routers in multicast groups. IP version 6 (IPv6) multicast routers use MLD to learn, for each of their attached physical networks, which groups have interested listeners. Each router maintains a list of host multicast addresses that have listeners for each sub network, as well as a timer for each address. However, the router does not need to know the address of each listener—just the address of each host. The router provides addresses to the multicast routing protocol it uses, which ensures that multicast packets are delivered to all sub networks where there are interested listeners. In this way, MLD is used as the transport for the Protocol Independent Multicast (PIM) Protocol.

MLD is an integral part of IPv6 and must be enabled on all IPv6 routers and hosts that need to receive IP multicast traffic. The Junos OS supports MLD versions 1 and 2. Version 2 is supported for source-specific multicast (SSM) include and exclude modes.

In include mode, the receiver specifies the source or sources it is interested in receiving the multicast group traffic from. Exclude mode works the opposite of include mode. It allows the receiver to specify the source or sources it is not interested in receiving the multicast group traffic from

For each attached network, a multicast router can be either a querier or a nonquerier. A querier router, usually one per subnet, solicits group membership information by transmitting

MLD queries. When a host reports to the querier router that it has interested listeners, the querier router forwards the membership information to the rendezvous point (RP) router by means of the receiver’s (host's) designated router (DR). This builds the rendezvous-point tree (RPT) connecting the host with interested listeners to the RP router. The RPT is the initial path used by the sender to transmit information to the interested listeners.

Chapter – 4

Security and Quality of Service in IPv6

Types of Threats:

In an attempt to categorize threats both to understand them better and to help in planning ways to resist them, the following four categories are typically used.

- Unstructured threats
- Structured threats
- Internal threats
- External threats

Unstructured Threats

Unstructured threats often involve unfocused assaults on one or more network systems, often by individuals with limited or developing skills. The systems being attacked and infected are probably unknown to the perpetrator. These attacks are often the result of people with limited integrity and too much time on their hands. Malicious intent might or might not exist, but there is always indifference to the resulting damage caused to others. Some of the examples are virus, worms and Torjan Horse.

Structured Threats

Structured threats are more focused by one or more individuals with higher-level skills actively working to compromise a system. The targeted system could have been detected through some random search process, or it might have been selected specifically. The attackers are typically knowledgeable about network designs, security, access procedures, and hacking tools, and they have the ability to create scripts or applications to further their objectives.

Structured attacks are more likely to be motivated by something other than curiosity or showing off to one's peers. Greed, politics, racism (or any intolerance), or law enforcement (ironic) could all be motives behind the efforts. Crimes of all types where the payoff isn't directly tied to the attack, such as identity theft or credit card information theft, are also motivations.

Internal Threats

Internal threats originate from individuals who have or have had authorized access to the network. This could be a disgruntled employee, an opportunistic employee, or an unhappy past employee whose access is still active. In the case of a past network employee, even if their account is gone, they could be using a compromised account or one they set up before leaving for just this purpose.

External Threats

External threats are threats from individuals outside the organization, often using the Internet or dial-up access. These attackers don't have authorized access to the systems.

Common Attacks in Both IPv4 and IPv6

IPv6 Security v1.1 12 IPv6 cannot solve all security problems. Basically it cannot prevent attacks on layers above the network layer in the network protocol stack. Possible attacks that IPv6 cannot address include:

1. Application layer attacks: Attacks performed at the application layer (OSI Layer 7) such as buffer overflow, viruses and malicious codes, web application attacks, and so on.
2. Brute-force attacks and password guessing attacks on authentication modules.
3. Rogue devices: Devices introduced into the network that are not authorized. A device may be a single PC, but it could be a switch, router, DNS server, DHCP server or even a wireless access point.
4. Denial of Service: The problem of denial of service attacks is still present with IPv6.
5. Attacks using social networking techniques such as email spamming, phishing, etc.

IPv6 Security Impact

Many security issues in IPv6 remain the same as in IPv4, but v6 also has new features that affect system and network security, as well as potentially impacting on policies and procedures. IPv6 and IPv4 usually operate completely independently over the same Layer 2 infrastructure, so additional and separate IPv6 security mechanisms must be implemented. Many areas will need overhauling, such as firewalls, monitoring and accounting. It is important to keep in mind that IPv6 is young operationally and may have issues not yet encountered, or even imagined.

IPv4-Only Systems

Many enterprises solely using IPv4 assume IPv6 intrusion cannot happen on their systems. This is quite incorrect – see IPv6 Security Myth No. 1. All sites should now firewall and monitor both IPv4 and IPv6. If IPv6 traffic is not monitored then it is impossible to know how much IPv6 traffic is on networks, and it is almost a certainty that some IPv6 traffic is being carried. At the user level, IPv6 can be accidentally or deliberately employed to bypass usage and security policies. See here for a list of IPv6 monitoring and testing software.

Moving To an IPv6 Frame of Mind

For decades, system and network admins have learnt to conserve and apportion scant IPv4 allocations. To deal with the astonishing abundance of IPv6 addresses takes a complete change of mindset. The standard IPv6 allocation for a single subnet or small enterprise is a /64 prefix, which contains four billion times the total of possible addresses in today's IPv4 Internet. An entirely new approach to addressing must be adopted to use IPv6 optimally, focused on well-designed layouts that reflect service location or function, network growth or potential mergers, or other relevant parameters.

An example of IPv4 thinking that must radically change in an IPv6 setting concerns ICMP (the ping protocol). In IPv6, routers do not fragment too-large packets, which greatly improves throughput. If a packet is too large to forward, the router discards the packet and sends the host an ICMPv6 Packet Too Big message, which includes the MTU of the next hop. The host now uses the lower MTU and successfully retransmits the packet. Many IPv4-experienced admins firmly believe blocking ICMP is a good security practice, but in IPv6 this will cause severe, difficult-to-diagnose problems.

Security Implications

ICMP and Multicast

The common IPv4 practice of blocking ICMP packets as a supposed security measure should not occur, as IPv6 functioning depends on ICMPv6 for error messages, path MTU discovery, multicast group management and Neighbor Discovery. IPv6 also relies upon multicast availability, which will impact on firewalls, intrusion detection and access control rules.

Dual Stacking

Dual stacking means devices have both IPv4 and IPv6 protocol capabilities. It is usually seen as an essential transition method for staged deployment of IPv6, but it means two protocols are in play: security must be maintained for both. This is expensive in terms of time and effort, so some large organisations, e.g. Facebook, are now adopting IPv6 entirely on their internal networks, and using conversion techniques at the network borders.

Automatic Tunnels

Tunneling means packets of one protocol are encapsulated by packets of a second protocol, for transport across a network of the second type. Tunnels are an essential IPv6 transition technique. However, some operating systems out of the box will automatically establish an IPv6 network when a client is connected to a server, e.g. various Windows releases. Potentially unwanted new paths to hosts can be set up, and firewalls may be unprepared.

Auto configuration

Auto configuration in IPv6 is an efficient and economic process, but has potential vulnerabilities. SLAAC (Stateless Address Auto configuration) is the process by which a host configures its own address based on its hardware (MAC) address. But the exposure of MAC addresses may permit host identification via interface ID, NIC vendor, or host vendor. Addresses generated by random, temporary, or cryptographic means can tackle this problem. DHCPv6 (Dynamic Host Configuration Protocol) allows a server to supply addresses to hosts. DHCP in IPv4 needed external support, but in IPv6 it requires nothing but a working router for the connected host to be immediately reachable.

Hosts with Multiple Addresses

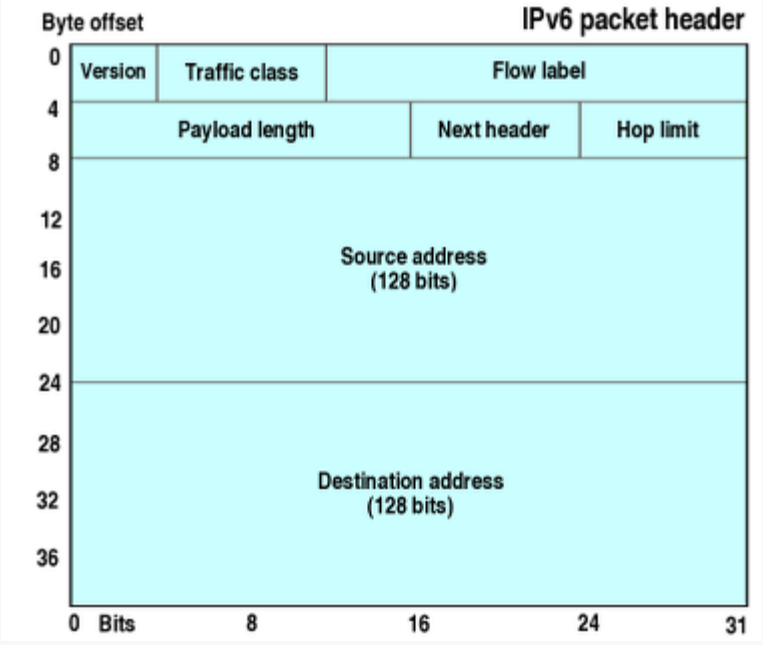
In IPv4, multiple addresses are always possible, but rare. But in IPv6 they are very common, arising from SLAAC, temporary DHCPv6, link-local addresses, multiple prefixes, overlapping lifetimes, as well as IPv4 addresses. Admins must be aware of all possible interface addresses and the capacity of network devices to create their own addresses, e.g. in conjunction with radvd, the Router Advertisement Daemon.

Scans and IPv6

With 18 billion billion addresses in a /64 subnet, sequential scanning is pointless. It would take 500,000 years to scan a single /64 at a million probes per second. However, hinted scanning (using other sources to gain information on address ranges) may still be possible. This can leverage facilities such as Neighbor Discovery, routing table, who is, or reverse DNS to locate vulnerable hosts.

IPv6 Packet Security

Unlike IPv4, IPsec security is mandated in the IPv6 protocol specification, allowing IPv6 packet authentication and/or payload encryption via the Extension Headers. However, IPsec is not automatically implemented, it must be configured and used with a security key exchange.



IPv6 Packet Structure

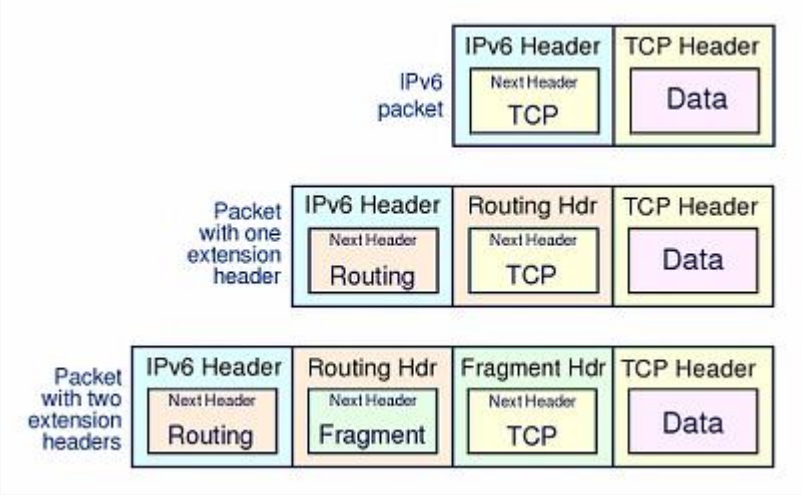
The IPv6 header is not variable, as in IPv4, but has a simple, efficient fixed 40-byte length. Minimum packet size is 1280 bytes, from 40 bytes of header plus 1240 bytes of payload.

1. **Next Header Field**

The Next Header field defines the type of header immediately following the current one. It is usually the payload, but sometimes Extension Headers provide valuable functions. Encryption capabilities are defined by the Authentication and Encapsulated Security headers.

2. **Extension Headers**

It specifies protocol numbers in required order of use as below:
000 Hop-by-hop – must be examined by every node on path to destination
043 Routing header – list of nodes that should be visited on path
060 Destination options – processed by routers along path
044 Fragment header – packet was fragmented at source if too large for path
051 Authentication header – part of IPsec
050 Encapsulated security payload – IPsec



060 Destination options – processed at destination

Right: A simple IPv6 packet (top row) with a TCP header and data payload. The second row shows the packet with an additional Routing header, third row has Routing and Fragment headers.

IPv6 Packet Encryption

IPsec defines cryptography-based security for both IPv4 and IPv6 in RFC 4301. IPsec support is an optional add-on in IPv4, but is a mandatory part of IPv6. It provides two security headers which can be used separately or together: Authentication Header (AH) and Encapsulating Security Payload (ESP), used in conjunction with security key exchange.

1. **Authentication Header**

AH provides connectionless integrity, data-origin authentication and protection against replay attacks. It authenticates with an Integrity Check Value (ICV) calculated over the payload, the header, and unchanging fields of the IPv6 header and options. AH does not provide privacy and confidentiality of packet contents. See RFC 2402.

2. Encapsulating Security Payload

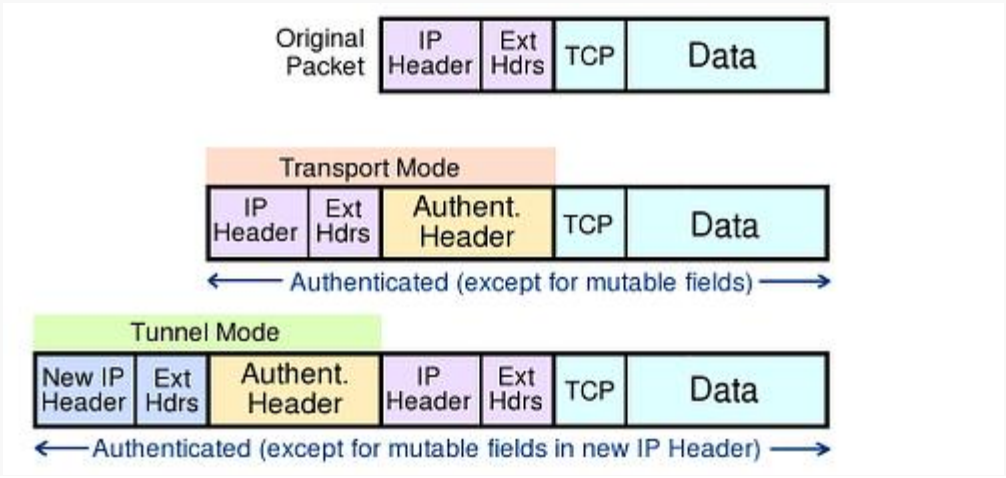
ESP also provides connectionless integrity, data-origin authentication, protection against replay attacks, limited traffic flow confidentiality, but also provides privacy and confidentiality through encryption of the payload. See RFC 2406.

3. IPsec Modes

IPSec operates in two different modes: Transport mode (host-to-host) and Tunnel mode (gateway- to- gateway or gateway-to-host).

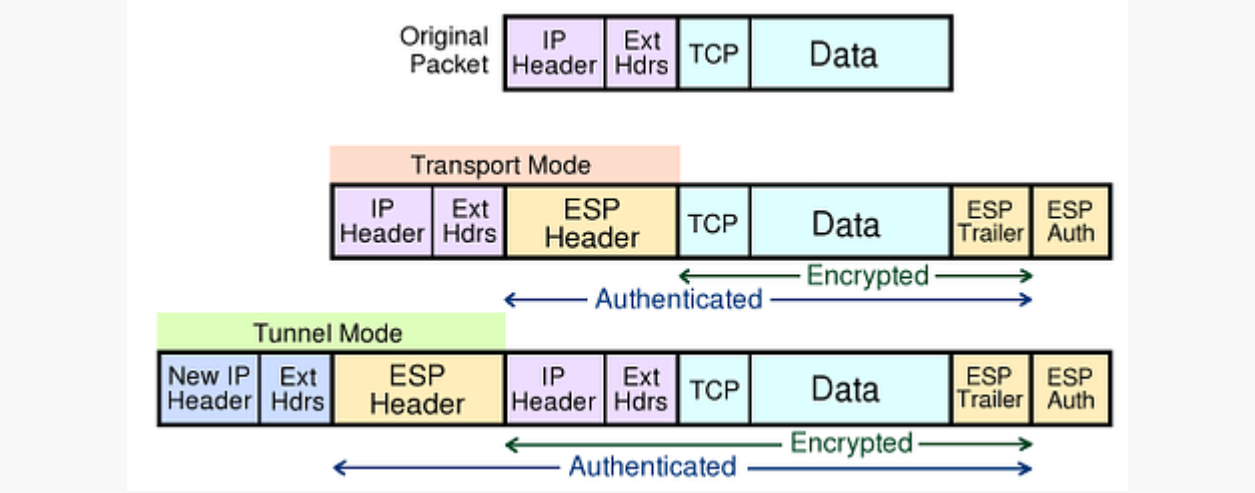
Transport mode: the IPv6 header of the original packet is used, followed by the AH or ESP header, then the payload.
Tunnel mode: a new IPv6 header encapsulates the AH or ESP header and the original IP header and payload.

Extension headers (Hop-by-Hop, Routing, Fragmentation) immediately follow their IP headers, except for Destination Options, which can appear before or after AH or ESP. ('TCP' below indicates any upper layer protocol.)



AH in Transport & Tunnel Modes

AH authenticates the packet and the outermost IPv6 addresses (except for mutable fields), but does not encrypt payloads. AH cannot be used to traverse NATs, as it calculates the integrity check value (ICV) over source and destination addresses: NATs translate addresses, so would



ESP in Transport & Tunnel Modes

ESP authentication does not include the outermost IPv6 headers, but in Tunnel mode it protects the original headers. ESP is used to build virtual private network tunnels between sites. It permits NAT traversal, as it does not use the outermost address values in the ICV calculation. If AH and ESP are used together, ESP is applied first, then AH authenticates the entire new packet.

The Security Association

Security Association is a record of the authentication algorithm, encryption algorithm, keys, mode (transport or tunnel), and sequence number, overflow flag, expiry of the SA, and anti-replay window. The SA is held in a database at each endpoint, indexed by outer destination address, IPsec protocol (AH or ESP), and Security Parameter Index value.

Selection of SA can be manually (pre-shared keys) but preferably is automated with Internet Key Exchange (IKE, IKEv2). IKE uses Daffier-Hellman techniques to create a shared secret encryption key used to negotiate SA data. For key exchange, IKE depends on a Public Key Infrastructure (PKI), which is not yet widespread. The framework and syntax for key exchange is ISAKMP (Internet Security Association and Key Management Protocol). See RFC 2408.

Quality of Service (QOS):

The term quality refers to the delivery of service better than its normal operation. The quality of service measures in terms of data loss, jitter, bandwidth etc in case of computer network. In general it is define as the better uses of resources. It is focused on

- ⇨ End to end communication
- ⇨ Client server application
- ⇨ Data transmission

QoS Paradigms:

It specifies a measure to identify the right approach for providing better QOS in network. It be expressed in terms of following class:

- ⇨ End system-based (client-based) QoS
- ⇨ Service-based QoS
- ⇨ Class/Priority-based QoS
- ⇨ Resource Reservation-based QoS:

End system-based (client-based) QoS:

The end systems must be able to compensate:

- ⇨ Packet loss.
- ⇨ Packet Delay.
- ⇨ Delay jitter, etc.

In this case: QOS is focused to maintain the nominal label of above stated parameter. In end system, there are some mechanisms that use extrapolation of missing data, intelligent playback buffers to compensate for variances in inter arrival times of packets.

Service-based QoS:

In IPv6, different service classes could be formed different groups with different multicast addresses. For example four different classes of the same audio traffic-each encoded with different quality Such as 5.5, 11, 22, and 44 KHz. They form four multicast group (have different queuing and processing by the routers and end systems). The existing IPv6 ICMP multicast feedback control message mechanism in such way that Can be used to shape the traffic characteristics at the sender and intermediate routers (e.g., the highest quality to be generated by the sender, which links need to carry this multicast group because of a multicast member downstream, how to enter/leave an multicast group, etc.). The sender have to provide the same data (but of different quality) multiple times. Hence service based QOS is focused on how better service can be provide.

Class/Priority-based QoS

To handle multimedia traffic requirements, routers need to know how to deal with packets with different service requirements (real-time, non-real-time, etc.). It means IPV6 provides the different class of service. For this purpose, IP packets have to contain corresponding information. IPv6 Routing Extension header can be used and contain information such as the Hop-By-Hop Extension header can transmit required information to all routers on the path. Network basically transmits packets in best-effort mode and may implement a feedback loop to the sender so that the sender

can adapt its multimedia coding. For this purpose, sender sends senders code and label packets with proper priority. For example, to transmit high-quality (44 KHz) audio, hierarchical coding may be used as

- ⇨ Data is encoded as a base audio packet containing the data for 5.5 KHz quality.
- ⇨ A second packet containing the difference between 5.5 and 11 KHz.
- ⇨ A third packet containing the difference between 11 and 22 KHz.
- ⇨ A fourth packet containing the difference between 22 and 44 KHz

Routers may drop packets according to priority. It may be done as:

- ✓ They allow to drop packets in case of congestion: the order is fourth, third, second, and finally first.
- ✓ The audio quality may differ, the audio packet is completely lost only if all four kinds of packets are all dropped.

Hence the Class/Priority-based QoS always focused to provide the better services in case of different class of service with priority.

Resource Reservation-based QoS:

It is the most complex QoS paradigm because the routers must have full knowledge of connections and their QoS requirements to reserve sufficient resources including queues and buffer space, computing time, specific algorithms, etc. With proper resource reservation setup negotiations and signaling, thus QoS can be guaranteed end-to-end between sender and receiver(s). For example, The Resource Reservation Protocol (RSVP).

Quality of Service Quality of Service (QoS) in IPv6 Protocols

QoS features supported for IPv6 environments include packet classification, queueing, traffic shaping, packet marking, and policing of IPv6 packets. These features are available for all FastIron products. The feature set is identical to that in IPv4.

To implement QoS in networks running IPv6, follow the same steps as those to implement QoS in networks running only IPv4. The recommended steps are as follows:

- ✓ Identify applications in your network and understand the characteristics of the applications so that you can make decisions about what QoS features to apply.
- ✓ Depending on network topology, link-layer header sizes are affected by changes and forwarding.
- ✓ Decide the method of classification, marking, and rate limiting. If the same network is carrying IPv4 and IPv6 traffic, decide if you want to treat both the same or differently, and specify match criteria accordingly. If you want to treat them the same, use match statements such as match dscp and set dscp. If you want to treat them differently, add match criteria such as match protocol ip and match protocol ipv6 in the match criteria.

Chapter -5

IPv6 Routing

Routing concepts remain same in case of IPv6 but almost all routing protocols have been redefined accordingly. We discussed earlier, how a host speaks to its gateway. Routing is a process to forward routable data choosing the best route among several available routes or path to the destination. A router is a device that forwards data that is not explicitly destined to it.

There exists two forms of routing protocols:

- **Distance Vector Routing Protocol**: A router running distance vector protocol advertises its connected routes and learns new routes from its neighbors. The routing cost to reach a destination is calculated by means of hops between the source and destination. A router generally relies on its neighbor for best path selection, also known as "routing-by-rumors". RIP and BGP are Distance Vector Protocols.
- **Link-State Routing Protocol**: This protocol acknowledges the state of a Link and advertises to its neighbors. Information about new links is learnt from peer routers. After all the routing information has been converged, the Link-State Routing Protocol uses its own algorithm to calculate the best path to all available links. OSPF and IS-IS are link state routing protocols and both of them use Dijkstra's Shortest Path First algorithm.

Routing protocols can be divided in two categories:

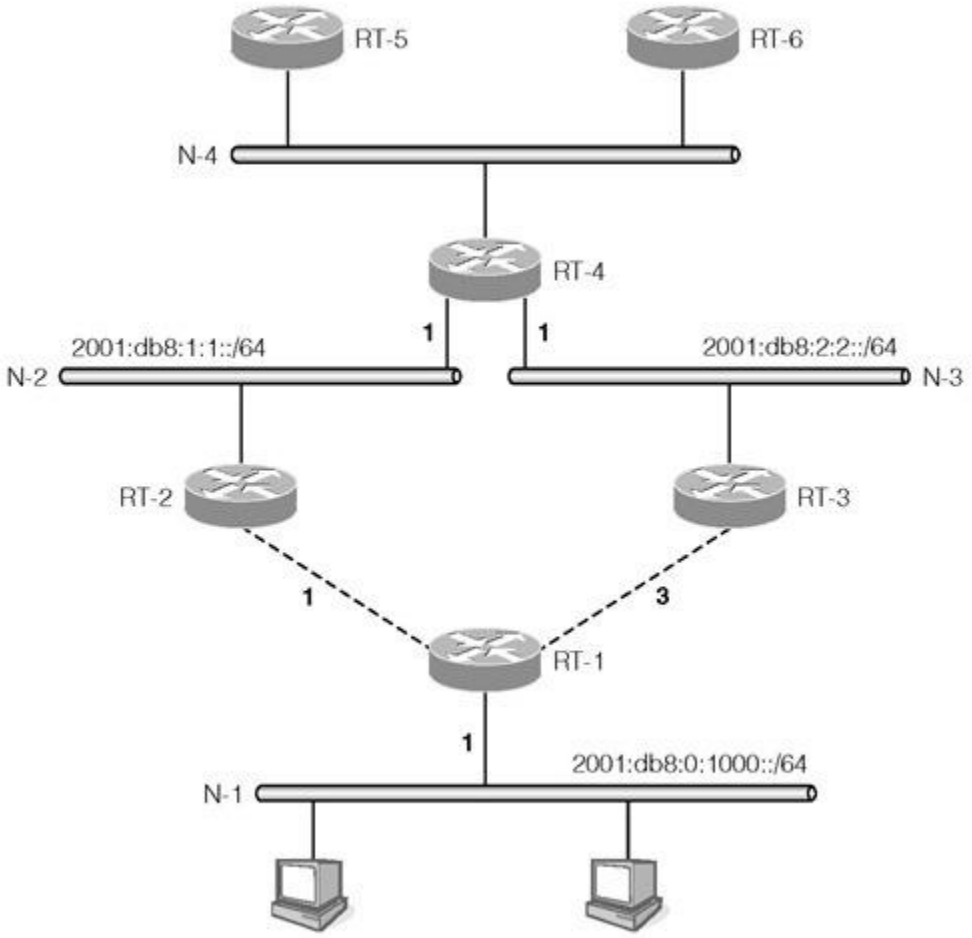
- **Interior Routing Protocol**: Protocols in this categories are used within an autonomous system or organization to distribute routes among all routers inside its boundary. Examples: RIP, OSPF.
- **Exterior Routing Protocol**: An Exterior Routing Protocol distributes routing information between two different autonomous systems or organization. Examples: BGP.

IPV6 Routing protocols:

1. **RIPng**

RIPng stands for Routing Information Protocol Next Generation. This is an Interior Routing Protocol and is a Distance Vector Protocol. RIPng has been upgraded to support IPv6. IPv6 RIP functions the same and offers the same benefits as RIP in IPv4. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes, and the use of the all-RIP-devices multicast group address FF02::9 as the destination address for RIP update messages.

Consider the example in Figure. Router RT-1 advertises its directly connected network N-1 of prefix 2001:db8:0:1000::/64 with a metric of 1 on the point-to-point links to RT-2 and RT-3. The costs of the links from RT-1 to RT-2 and RT-3 are 1 and 3 respectively, which have RT-2 and RT-3 advertise the same prefix on networks N-2 and N-3 (where router RT-4 resides) but with different metrics. RT-2 advertises a metric value of 2 while RT-3 advertises a metric value of 4. After processing the routing messages from RT-2 and RT-3, RT-4 selects the route with the smaller metric and chooses RT-2 as the next hop to reach network 2001:db8:0:1000::/64 with a metric of 2.



2. OSPF

Open Shortest Path First (OSPF) is a routing protocol for IP. It is a link-state protocol, as opposed to a distance-vector protocol. A link-state protocol makes routing decisions based on the states of the links that connect source and destination machines. The state of a link is a description of that interface and the relationship to its neighboring networking devices. The interface information includes the IPv6 prefix of the interface, the network mask, the type of network it is connected to, the routers connected to that network, and so forth. This information is propagated in various type of link-state advertisements (LSAs). Open Shortest Path First version 3 is an Interior Routing Protocol which is modified to support IPv6. This is a Link-State Protocol and uses Djikrasta's Shortest Path First algorithm to calculate best path to all destinations.

OSPF router within a network communicates with other neighboring routers on each connecting interface to establish the states of all adjacencies. Every such communication sequence is a separate conversation identified by the pair of router IDs of the communicating neighbors. RFC 2328 specifies the protocol for initiating these conversations (Hello Protocol) and for establishing full adjacencies (Database Description Packets, Link State Request Packets). During its course, each router conversation transitions through a maximum of eight conditions defined by a state machine.

1. Down: The state down represents the initial state of a conversation when no information has been exchanged and retained between routers with the Hello Protocol.
2. Attempt: The Attempt state is similar to the Down state, except that a router is in the process of concerted efforts to establish a conversation with another router, but is only used on NBMA networks.
3. Init: The Init state indicates that a HELLO packet has been received from a neighbor, but the router has not established a two-way conversation.
4. 2-Way: The 2-Way state indicates the establishment of a bidirectional conversation between two routers. This state immediately precedes the establishment of adjacency. This is the lowest state of a router that may be considered as a Designated Router.
5. ExStart: The ExStart state is the first step of adjacency of two routers.
6. Exchange: In the Exchange state, a router is sending its link state database information to the adjacent neighbor. At this state, a router is capable to exchange all OSPF routing protocol packets.
7. Loading: In the Loading state, a router requests the most recent Link-state advertisements (LSAs) from its neighbor discovered in the previous state. 8. Full:

The Full state concludes the conversation when the routers are fully adjacent, and the state appears in all router- and network-LSAs. The link state databases of the neighbors are fully synchronized.

Protocol messages

Unlike other routing protocols, OSPF does not carry data via a transport protocol, such as the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP). Instead, OSPF forms IP datagrams directly, packaging them using protocol number 89 for the IP Protocol field. OSPF defines five different message types, for various types of communication:

Hello:

Hello messages are used as a form of greeting, to allow a router to discover other adjacent routers on its local links and networks. The messages establish relationships between neighboring devices (called adjacencies) and communicate key parameters about how OSPF is to be used in the autonomous system or area.

Database Description

Database Description messages contain descriptions of the topology of the autonomous system or area. They convey the contents of the link-state database (LSDB) for the area from one router to another. Communicating a large LSDB may require several messages to be sent by having the sending device designated as a master device and sending messages in sequence, with the slave (recipient of the LSDB information) responding with acknowledgements.

Link State Request

These messages are used by one router to request updated information about a portion of the LSDB from another router. The message specifies exactly which link(s) about which the requesting device wants more current information

Link State Update

These messages contain updated information about the state of certain links on the LSDB. They are sent in response to a Link State Request message, and also broadcast or multicast by routers on a regular basis. Their contents are used to update the information in the LSDBs of routers that receive them.

Link State Acknowledgment

These messages provide reliability to the link-state exchange process, by explicitly acknowledging receipt of a Link State Update message.

OSPFV3:

OSPFV3 runs per link No Authentication as it relies on the Authentication Header and Encapsulation security payload capabilities of IPV6 .Router and Network LSA does not carry any IPV6 address. Only independent topology information. 3 different flooding scope in OSPFV3

- Link-local scope: LSA flooded on the local link only
- Area scope: LSA flooded throughout the area, but not beyond
- AS scope: LSAs are flooded throughout the entire OSPF domain

OSPF V3 uses the link-local unicast address as the source of messages sent to neighbors. A routers learns its neighbor's link-local addresses and uses them as the next-hop address for its prefix However Link local address are never carried as prefix information in other LSA. OSPFv3 (OSPF for IPv6) doesn't include any authentication capabilities of its own; instead, it relies entirely on IPsec to secure communications between neighbors. Below are the three different types of authentication supported by standard OSPF.

- **Null Authentication**—This is also called Type 0 and it means no authentication information is included in the packet header. It is the default.
- **Plain Text Authentication**—This is also called Type 1 and it uses simple clear-text passwords.
- **MD5 Authentication**—This is also called Type 2 and it uses MD5 cryptographic passwords.

Authentication does not need to be set. However, if it is set, all peer routers on the same segment must have the same password and authentication method. The examples in this document demonstrate configurations for both plain text and MD5 authentication.

Difference between OSPFV2 and OSPFV2:

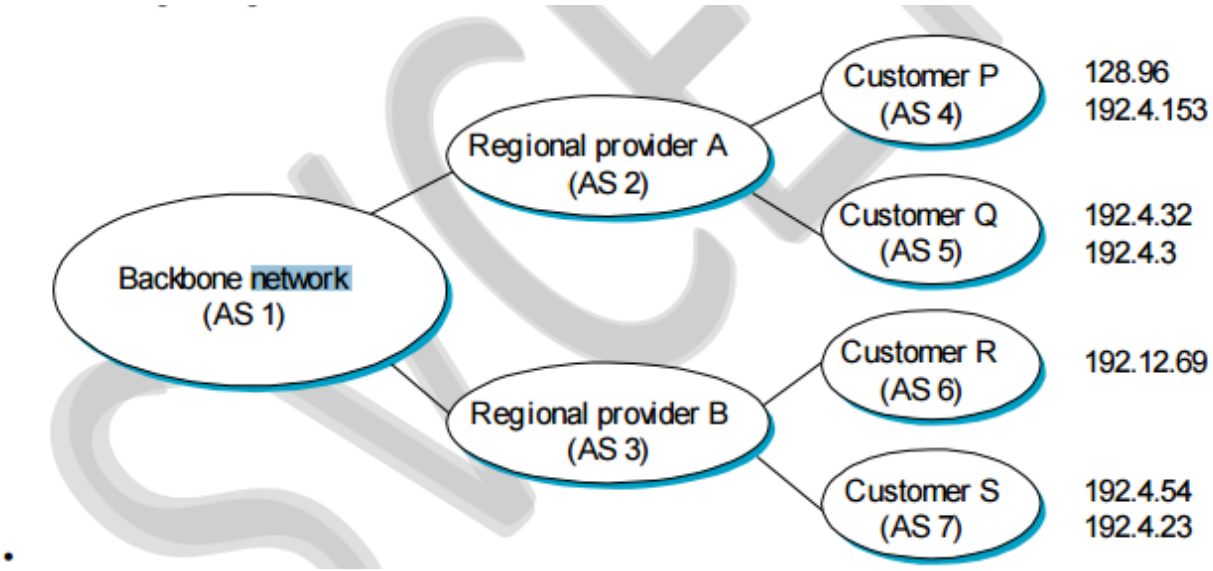
The version of OSPF for IPv6 is OSPFv3. The previous version of OSPF which supports IPv4 is OSPFv2. OSPFv3 is based on OSPFv2. Following table lists the main differences between OSPFv2 and OSPFv3.

Protocol Feature	OSPFv2	OSPFv3
Distance Vector / Link State	Link State	Link State
Routed Protocol Supported	IPv4	IPv6
VLSM Support	Yes	Yes
Router ID	32 bit Binary ID	32 bit Binary ID
Metric Value	Cost (Based on Bandwidth)	Cost (Based on Bandwidth)
How DR and BDR are elected	Based on highest priority value and then highest RID	Based on highest priority value and then highest RID
OSPF multicast all routers IP address	224.0.0.5	FF02::5
OSPF DR and BDR multicast IP address	224.0.0.6	FF02::6
Support for multiple OSPF instances per interface	Not available	Available

3. BGP

BGP stands for Border Gateway Protocol. It is the only open standard Exterior Gateway Protocol available. BGP is a Distance Vector protocol which takes Autonomous System as calculation metric, instead of the number of routers as Hop. BGPv4 is an upgrade of BGP to support IPv6 routing.

- BGP supports flexibility -- paths could be chosen by a provider based on a policy.
- To configure BGP, each AS admin picks at least one node to be the “BGP” speaker - - a spokesperson node for the entire AS.
- In addition, there are border gateways using which packets enter/leave ASes.
- Source advertises complete paths (unlike distance vector or link state routing) -- thus loops are prevented.



- AS 2 says 128.96, 192.4.15, 192.4.32, 192.4.3 can be reached via AS 2.
- AS 1 advertises that these networks can be reached via --note full path description.
- Loops are avoided.

BGP Messages:

BGP has four types of messages

1. OPEN: Establish a connection with a BGP peer
2. UPDATE -- advertise or withdraw routes to a destination.
3. KEEPALIVE: Inform a peer that the sender is still alive but has no information to send.
4. NOTIFICATION: Notify that errors are detected

BGP Extensions for IPv6:

There is no actual BGP for IPv6. The IPv6 support derives from the capability of BGP-4 to exchange information about network layer protocols other than IPv4. These multiprotocol extensions of BGP-4 are defined in RFC 2858, which obsoletes RFC 2283. RFC 2283 is mentioned here because it is the base document for RFC 2545, which defines the IPv6 extensions of BGP-4. It is important to understand BGP-4 fully before looking at its multiprotocol extensions. The following sections start with a short overview of BGP-4 and its operations as defined in RFC 1771. BGP message types are then discussed. The last part covers the implementation of IPv6 information carried within BGP-4.

The BGP-4 protocol [BGP-4] in particular, and path vector routing protocols in general, are mostly independent of the particular Address Family for which the protocol is being used. IPv6 falls under the generic category of protocols for which BGP-4 is suitable and, unless stated otherwise in this document, the BGP-4 procedures to apply when using BGP-4 to carry IPv6 reachability information are those defined in [BGP-4] and in subsequent documents that extend or update the BGP-4 specification. In terms of routing information, the most significant difference between IPv6 and IPv4 (for which BGP was originally designed) is the fact that IPv6 introduces scoped unicast addresses and defines particular situations when a particular address scope must be used.

Multicast routing (DVMRP):

The Distance Vector Multicast Routing Protocol (DVMRP), defined in RFC 1075, is a routing protocol used to share information between routers to facilitate the transportation of IP multicast packets among networks.

DVMRP (Distance Vector Multicast Routing Protocol) is the oldest routing protocol that has been used to support multicast data transmission over networks.

DVMRP (Distance Vector Multicast Routing Protocol) is the oldest routing protocol that has been used to support multicast data transmission over networks. The protocol sends multicast data in the form of unicast packets that are reassembled into multicast data at the destination.

DVMRP can run over various types of networks, including Ethernet local area networks (LANs). It can even run through routers that are not multicast-capable. It has been considered as an intermediate solution while "real" multicast Internet Protocol (IP) routing evolves.

Operation

The protocol is based on the RIP protocol. The router generates a routing table with the multicast group of which it has knowledge with corresponding distances (i.e. number of devices/routers between the router and the destination).

When a multicast packet is received by a router, it is forwarded by the router's interfaces specified in the routing table. DVMRP operates via a reverse path flooding technique, sending a copy of a received packet (specifically IGMP messages for exchanging routing information with other routers) out through each interface except the one at which the packet arrived. If a router (i.e. a LAN which it borders) does not wish to be part of a particular multicast group, it sends a "prune message" along the source path of the multicast.

Criticisms

Like most distance-vector protocols, DVMRP has difficulties with network scaling, primarily due to the periodic re flooding necessary to detect new hosts. This was more prevalent in early versions

of the protocol, prior to the implementation of pruning. DVMRP's flat unicast routing mechanism, which is used to determine the source interface of a data stream, also affects its ability to scale.

DVMRP is the original IP multicast routing protocol. It was designed to run over both multicast capable LANs (like Ethernet) as well as through non-multicast capable routers. In the case of non-multicast capable routers, the IP multicast packets are "tunneled" through the routers as unicast packets. Because DVMRP replicates the packets, it has an effect on performance, but has provided an intermediate solution for IP multicast routing on the Internet while router vendors decide to support native IP multicast routing. When configured, DVMRP defaults to enabling all interfaces that are multicast capable.

Multicast routing PIM:

Protocol-Independent Multicast (PIM) is a family of multicast routing protocols for Internet Protocol (IP) networks that provide one-to-many and many-to-many distribution of data over a LAN, WAN or the Internet. It is termed protocol-independent because PIM does not include its own topology discovery mechanism, but instead uses routing information supplied by other routing protocols. There are four variants of PIM:

- **PIM Sparse Mode (PIM-SM)** explicitly builds unidirectional shared trees rooted at a rendezvous point (RP) per group, and optionally creates shortest-path trees per source. PIM-SM generally scales fairly well for wide-area usage.
- **PIM Dense Mode (PIM-DM)** uses dense multicast routing. It implicitly builds shortest-path trees by flooding multicast traffic domain wide, and then pruning back branches of the tree where no receivers are present. PIM-DM is straightforward to implement but generally has poor scaling properties. The first multicast routing protocol, DVMRP used dense-mode multicast routing. See the PIM Internet Standard RFC 3973.
- **Bidirectional PIM** explicitly builds shared bi-directional trees. It never builds a shortest path tree, so may have longer end-to-end delays than PIM-SM, but scales well because it needs no source-specific state. See Bidirectional PIM Internet Standard RFC 5015.
- **PIM Source-Specific Multicast (PIM-SSM)** builds trees that are rooted in just one source, offering a more secure and scalable model for a limited amount of applications (mostly broadcasting of content). In SSM, an IP datagram is transmitted by a source S to an SSM destination address G, and receivers can receive this datagram by subscribing to channel (S, G). See informational RFC 3569.

PIM-SM is commonly used in IPTV systems for routing multicast streams between VLANs, Subnets or local area networks

Protocol Independent Multicast - Sparse-Mode (PIM-SM)

It is a protocol for efficiently routing Internet Protocol (IP) packets to multicast groups that may span wide-area and inter domain internets. The protocol is named protocol-independent because it is not dependent on any particular unicast routing protocol for topology discovery, and sparse-mode because it is suitable for groups where a very low percentage of the nodes (and their routers)

will subscribe to the multicast session. Unlike earlier dense-mode multicast routing protocols such as DVMRP and dense multicast routing which flooded packets across the network and then pruned off branches where there were no receivers, PIM-SM explicitly constructs a tree from each sender to the receivers in the multicast group.

Multicast clients

- A router receives explicit Join/Prune messages from those neighboring routers that have downstream group members.
- In order to join a multicast group, G, a host conveys its membership information through the Internet Group Management Protocol (IGMP).
- The router then forwards data packets addressed to a multicast group G to only those interfaces on which explicit joins have been received.
- A Designated Router (DR) sends periodic Join/Prune messages toward a group-specific Rendezvous Point (RP) for each group for which it has active members.
- Each router along the path toward the RP builds a wild card (any-source) state for the group and sends Join/Prune messages on toward the RP.

The term route entry is used to refer to the state maintained in a router to represent the distribution tree. A route entry may include such fields as:

- source address
- the group address
- the incoming interface from which packets are accepted
- the list of outgoing interfaces to which packets are sent
- timers, flag bits, etc. o The wild card route entry's incoming interface points toward the RP

The outgoing interfaces point to the neighboring downstream routers that have sent Join/Prune messages toward the RP as well as the directly connected hosts which have requested membership to group G. This state creates a shared, RP-centered, distribution tree that reaches all group members.

Multicast sources:

- When a data source first sends to a group, its Designated Router (DR) unicasts Register messages to the Rendezvous Point (RP) with the source's data packets encapsulated within.
- If the data rate is high, the RP can send source-specific Join/Prune messages back towards the source and the source's data packets will follow the resulting forwarding state and travel encapsulated to the RP. Whether they arrive encapsulated or natively, the RP forwards the source's de-capsulated data packets down the RP-centered distribution tree toward group members.
- If the data rate warrants it, routers with local receivers can join a source-specific, shortest path, distribution tree, and prune this source's packets off the shared RP-centered tree.

- For low data rate sources, neither the RP, nor last-hop routers need join a source-specific shortest path tree nor can data packets be delivered via the shared RP-tree.

Once the other routers which need to receive those group packets have subscribed, the RP will unsubscribe to that multicast group, unless it also needs to forward packets to another router or node. Additionally, the routers will use reverse-path forwarding to ensure that there are no loops for packet forwarding among routers that wish to receive multicast packets.

Chapter -6

IPv4/IPv6 Transition Mechanisms

Migration from IPv4 to IPv6

Migrating from IPv4 to IPv6 in an instant is impossible because of the huge size of the Internet and of the great number of IPv4 users. Moreover, many organizations are becoming more and more dependent on the Internet for their daily work, and they therefore cannot tolerate downtime for the replacement of the IP protocol. As a result, there will not be one special day on which IPv4 will be turned off and IPv6 turned on because the two protocols can coexist without any problems. The migration from IPv4 to IPv6 must be implemented node by node by using auto configuration procedures to eliminate the need to configure IPv6 hosts manually. This way, users can immediately benefit from the many advantages of IPv6 while maintaining the possibility of communicating with IPv4 users or peripherals. Consequently, there is no reason to delay updating to IPv6. The key goals of the migration are as follow:

- IPv6 and IPv4 hosts must interoperate.
- The use of IPv6 hosts and routers must be distributed over the Internet in a simple and progressive way, with a little interdependence.
- Network administrators and end users must think that the migration is easy to understand and implement.

A set of mechanisms called SIT (Simple Internet Transition) has been implemented; it includes protocols and management rules to simplify the migration. The main characteristics of SIT are the following:

1. **Possibility of a progressive and nontraumatic transition:** IPv4 hosts and routers can be updated to IPv6, one at a time, without requiring other hosts or routers to be updated simultaneously.
2. **Minimum requirements for updating:** The only requirement for updating hosts to IPv6 is the availability of a DNS server to manage IPv6 addresses. No requirements are needed for routers.
3. **Addressing simplicity:** When a router or a host is updated to IPv6, it can also continue to use IPv4 addresses.
4. **Low initial cost:** No preparatory work is necessary to begin the migration to IPv6. Mechanisms used by SIT include the following:
5. A structure of IPv6 addresses that allows the derivation of IPv6 addresses from IPv4 addresses.
6. The availability of the dual stack on hosts and on routers during the transition—that is, the presence of both IPv4 and IPv6 stacks at the same time.
7. A technique to encapsulate IPv6 packets inside IPv4 packets (tunneling) to allow IPv6 packets to traverse clouds not yet updated to IPv6.

8. An optional technique that consists of translating IPv6 headers into IPv4 headers and vice versa to allow, in an advanced phase of the migration, IPv4-only nodes to communicate with IPv6-only nodes.

The SIT approach guarantees that IPv6 hosts can interoperate with IPv4 hosts initially on the entire Internet. When the migration is completed, this interoperability will be locally guaranteed for a long time. This capability allows for the protection of investments made on IPv4; simple devices that cannot be updated to IPv6—for example, network printers and terminal servers—will continue to operate with IPv4 until they are no longer used.

Transition from IPv4 to IPv6

Complete transition from IPv4 to IPv6 might not be possible because IPv6 is not backward compatible. This results in a situation where either a site is on IPv6 or it is not. It is unlike implementation of other new technologies where the newer one is backward compatible so the older system can still work with the newer version without any additional changes.

To overcome this short-coming, we have a few technologies that can be used to ensure slow and smooth transition from IPv4 to IPv6.

IPv4-IPv6 Transition Methods

Due to the time that change takes, IETF has been working on specific provisions to allow a smooth transition from version 4 to version 6, and hardware and software interoperability solutions to let newer IPv6 devices access IPv4 hosts. A technique was included in IPv6 to allow administrators to embed IPv4 addresses within IPv6 addresses. Special methods are defined to handle interoperability, including:

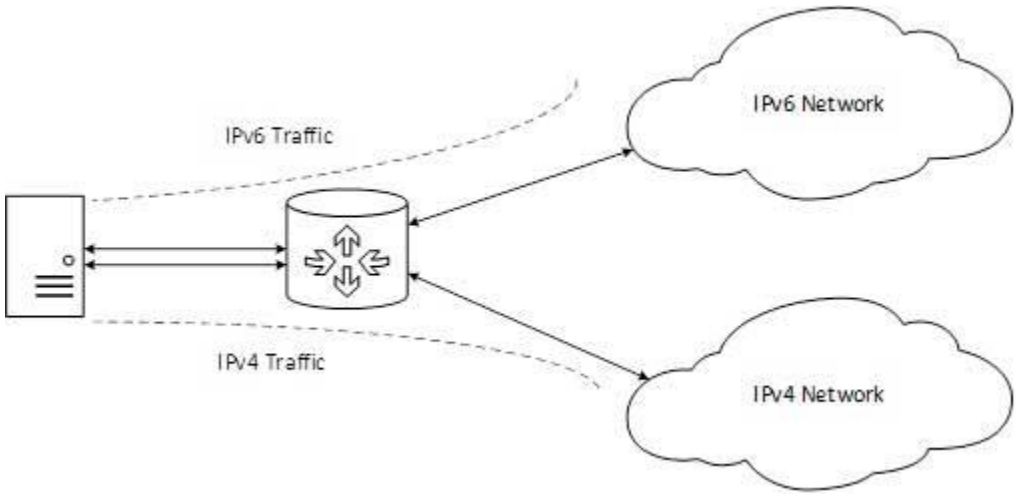
- **"Dual Stack" Devices:** Routers and some other devices may be programmed with both IPv4 and IPv6 implementations to allow them to communicate with both types of hosts.
- **IPv4/IPv6 Translation:** "Dual stack" devices may be designed to accept requests from IPv6 hosts, convert them to IPv4 datagrams, send the datagrams to the IPv4 destination and then process the return datagrams similarly.
- **IPv4 Tunneling of IPv6:** IPv6 devices that don't have a path between them consisting entirely of IPv6-capable routers may be able to communicate by encapsulating IPv6 datagrams within IPv4. In essence, they would be using IPv6 on top of IPv4; two network layers. The encapsulated IPv4 datagrams would travel across conventional IPv4 routers.

Bear in mind that these solutions generally only address backward compatibility, to allow IPv6 devices to talk to IPv4 hardware. Forward compatibility between IPv4 and IPv6 is not possible because IPv4 hosts cannot communicate with IPv6 hosts—they lack the knowledge of how IPv6 works. It is possible that certain special adaptations might be created to allow IPv4 hosts to access IPv6 hosts. But eventually, all IPv4 devices of any importance will want to migrate to IPv6.

The IETF has done such a good job in the past with introducing new technologies, and so much effort has been put into the IPv6 transition, that I am quite confident that the transition to IPv6 will come off with few, if any, problems. One good thing about the transition is that IPv4 is, at the present time, still getting the job done, so there is no big hurry to make the move to version 6. While technologies such as CIDR and NAT are "band-aids" on IPv4, they have been very successful ones in extending the useful life of the aging protocol.

Dual Stack Routers

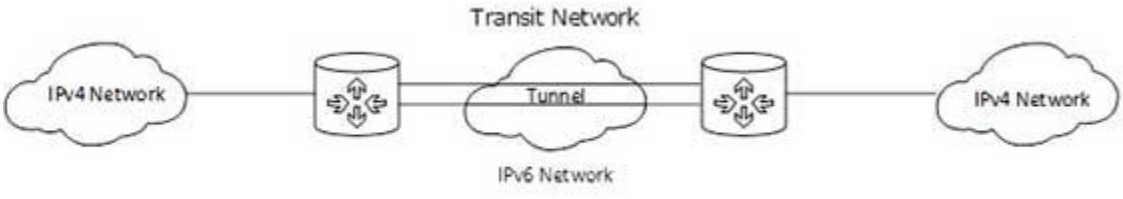
A router can be installed with both IPv4 and IPv6 addresses configured on its interfaces pointing to the network of relevant IP scheme.



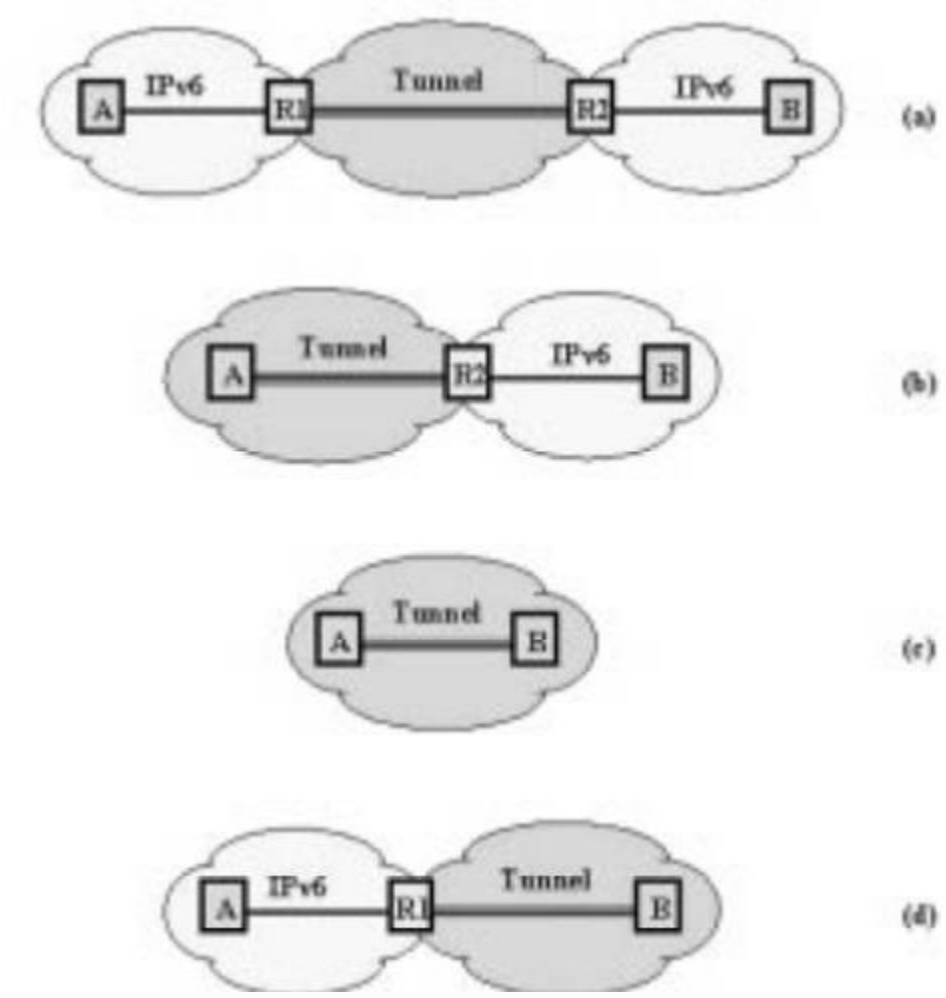
In the above diagram, a server having IPv4 as well as IPv6 address configured for it can now speak with all the hosts on both the IPv4 as well as the IPv6 networks with the help of a Dual Stack Router. The Dual Stack Router, can communicate with both the networks. It provides a medium for the hosts to access a server without changing their respective IP versions.

Tunneling

In a scenario where different IP versions exist on intermediate path or transit networks, tunneling provides a better solution where user's data can pass through a non-supported IP version.



The above diagram depicts how two remote IPv4 networks can communicate via a Tunnel, where the transit network was on IPv6. Vice versa is also possible where the transit network is on IPv6 and the remote sites that intend to communicate are on IPv4.



During the migration, the tunneling technique can be used in the following ways:

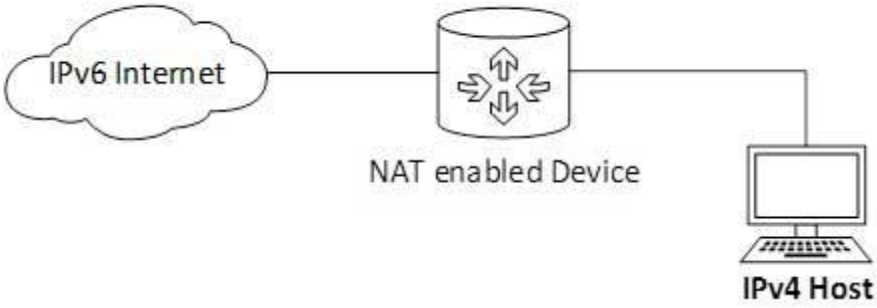
1. **Router-to-router:** IPv6/IPv4 routers interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. See Figure a.
2. **Host-to-router:** IPv6/IPv4 hosts can tunnel IPv6 packets to an intermediary IPv6/IPv4 router that can be reached via an IPv4 infrastructure. See Figure b
3. **Host-to-host:** IPv6/IPv4 hosts that are interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. See Figure c.
4. **Router-to-host:** IPv6/IPv4 routers can use tunnels to reach an IPv6/IPv4 host via an IPv4 infrastructure. See Figure d.

In the first two tunneling methods—router-to-router and host-to-router —the IPv6 packet is tunneled to a router; therefore, the endpoint of this type of tunnel is a router that must decode the IPv6 packet and forward it to its final destination. No relationship exists between the router address and the final destination address. For this reason, the router address that is the tunnel endpoint must be manually configured. This type of tunnel is called a configured tunnel.

In the last two tunneling methods—host-to-host and router-to-host— the IPv6/IPv4 packet is tunneled from a host or from a router to its destination host. In this case, the tunnel endpoint address and the destination host address are the same. If the IPv6 address used for the destination node is an IPv4-compatible address, the tunnel endpoint IPv4 address can be automatically derived from the IPv6 address, and therefore no manual configurations are necessary. These tunnels are also called automatic tunnels.

NAT Protocol Translation

This is another important method of transition to IPv6 by means of a NAT-PT (Network Address Translation – Protocol Translation) enabled device. With the help of a NAT-PT device, actual can take place happens between IPv4 and IPv6 packets and vice versa. See the diagram below:



A host with IPv4 address sends a request to an IPv6 enabled server on Internet that does not understand IPv4 address. In this scenario, the NAT-PT device can help them communicate. When the IPv4 host sends a request packet to the IPv6 server, the NAT-PT device/router strips down the IPv4 packet, removes IPv4 header, and adds IPv6 header and passes it through the Internet. When a response from the IPv6 server comes for the IPv4 host, the router does vice versa.

Chapter -7

IPv6 Network and Server Deployment

IPv6 introduces additional features to an existing network. Therefore, when you first deploy IPv6, you must ensure that you do not disrupt any operations that are working with IPv4. The subjects covered in this section describe how to introduce IPv6 to an existing network in a step-by-step fashion.

Preparing the Network Topology for IPv6 Support

The first step in IPv6 deployment is to assess which existing entities on your network can support IPv6. In most cases, the network topology-wires, routers, and hosts-can remain unchanged as you implement IPv6. However, you might have to prepare existing hardware and applications for IPv6 before actually configuring IPv6 addresses on network interfaces.

Verify which hardware on your network can be upgraded to IPv6. For example, check the manufacturers' documentation for IPv6 readiness regarding the following classes of hardware:

- Routers
- Firewalls
- Servers
- Switches

Preparing Network Services for IPv6 Support

The following typical IPv4 network services in the current Oracle Solaris release are IPv6 ready:

- send mail
- NFS
- HTTP (Apache 2.x or Orion)
- DNS
- LDAP

The IMAP mail service is for IPv4 only.

Nodes that are configured for IPv6 can run IPv4 services. When you turn on IPv6, not all services accept IPv6 connections. Services that have been ported to IPv6 will accept a connection. Services that have not been ported to IPv6 continue to work with the IPv4 half of the protocol stack.

Preparing Servers for IPv6 Support

Because servers are considered IPv6 hosts, by default their IPv6 addresses are automatically configured by the Neighbor Discovery protocol. However, many servers have multiple network

interface cards (NICs) that you might want to swap out for maintenance or replacement. When you replace one NIC, Neighbor Discovery automatically generates a new interface ID for that NIC. This behavior might not be acceptable for a particular server.

How to Prepare Network Services for IPv6 Support

1. Update the following network services to support IPv6:
    - Mail servers
    - NIS servers
    - NFS
2. Verify that your firewall hardware is IPv6 ready.

    Refer to the appropriate firewall-related documentation for instructions.

3. Verify that other services on your network have been ported to IPv6.

    For more information, refer to marketing collateral and associated documentation for the software.

4. If your site deploys the following services, make sure that you have taken the appropriate measures for these services:

    - Firewalls
        - Consider strengthening the policies that are in place for IPv4 to support IPv6. For more security considerations.
    - Mail
        - In the MX records for DNS, consider adding the IPv6 address of your mail server.
    - DNS
        - For DNS-specific considerations.
    - IPQoS
        - Use the same Diffserv policies on a host that were used for IPv4.
5. Audit any network services that are offered by a node prior to converting that node to IPv6

How to Prepare DNS for IPv6 Support

The current Oracle Solaris release supports DNS resolution on both the client side and the server side. Do the following to prepare DNS services for IPv6.

1. Ensure that the DNS server that performs recursive name resolution is dual-stacked (IPv4 and IPv6) or for IPv4 only.
2. On the DNS server, populate the DNS database with relevant IPv6 database AAAA records in the forward zone.
3. Add the associated PTR records for the AAAA records into the reverse zone.

4. Add either IPv4 only data, or both IPv6 and IPv4 data into the NS record that describes zones.

Planning for Tunnels in the Network Topology:

The IPv6 implementation supports a number of tunnel configurations to serve as transition mechanisms as your network migrates to a mix of IPv4 and IPv6. Tunnels enable isolated IPv6 networks to communicate. Because most of the Internet runs IPv4, IPv6 packets from your site need to travel across the Internet through tunnels to destination IPv6 networks.

Here are some major scenarios for using tunnels in the IPv6 network topology:

- The ISP from which you purchase IPv6 service allows you to create a tunnel from your site's boundary router to the ISP network. .
- Sometimes, a router in your infrastructure cannot be upgraded to IPv6. In this case, you can create a manual tunnel over the IPv4 router, with two IPv6 routers as endpoints.

Security Considerations for the IPv6 Implementation

When you introduce IPv6 into an existing network, you must take care not to compromise the security of the site. Be aware of the following security issues as you phase in your IPv6 implementation:

- The same amount of filtering is required for both IPv6 packets and IPv4 packets.
- IPv6 packets are often tunneled through a firewall. Therefore, you should implement either of the following scenarios:
    - Have the firewall do content inspection inside the tunnel.
    - Put an IPv6 firewall with similar rules at the opposite tunnel endpoint.
- Some transition mechanisms exist that use IPv6 over UDP over IPv4 tunnels. These mechanisms might prove dangerous by short-circuiting the firewall.
- IPv6 nodes are globally reachable from outside the enterprise network. If your security policy prohibits public access, you must establish stricter rules for the firewall. For example, consider configuring a stateful firewall.

IPv6 Network Configuration in Linux and Windows Machines:

IPv6 Proxy Server provides authentication free unrestricted Internet access to users. For using IPv6 Proxy Server, users will have to enable IPv6 on their machines because IPv6 Proxy Server serves requests from IPv6 clients only. Users need not configure IPv6 address or any related network configuration. An IPv6 enabled machine will automatically pick IPv6 address and related configurations from the CC Backbone router.

The advantages of using IPv6 Proxy services are:

Unrestricted Access:

Users will get authentication free unrestricted access to all the Internet sites and content.

Mobility:
Users can move their machines/laptops anywhere in the Institute and they will get seamless Internet access without having to reconfigure the network settings of their machine.

Supporting IPv6 in the Linux

IPv6 support can be enabled as a built-in kernel feature or as a loadable module. The support for IPv6 is default in the latest release of Fedora Core 1 and Core 2. For earlier versions, a module has to be loaded to support IPv6.

If you are using Fedora, then you will have to add/modify the line

NETWORKING_IPV6=yes

in /etc/sysconfig/network file and restart network service.

We recommend users to use Fedora Core 1or Core 2 Linux OS to access IPv6 Proxy Server. However if you want to use earlier versions of Linux, then follow the following steps.

Check for IPv6 support in the current running kernel

To check, whether your current running kernel supports IPv6, take a look into your /proc−file−system.

Following entry must exists:

/proc/net/if_inet6

A short automatic test looks like:

[root@ns root]# test -f /proc/net/if_inet6 && echo "Running kernel is IPv6 ready"

Running kernel is IPv6 ready

If this fails, it is quite likely, that the IPv6 module is not loaded.

Loading IPv6 module

You can try to load the IPv6 module by executing

[root@ns root]# modprobe ipv6

If this is successful, this module should be listed, testable with following command:

[root@ns root]# lsmod |grep -w 'ipv6' & echo "IPv6 Module loaded Succesfully"

[1] 32236

IPv6 Module loaded successfully

Note: unloading the module is currently not supported and can result, under some circumstances, in a kernel crash

Automatically loading IPv6 module

Its possible to automatically load the IPv6 module on demand. You only have to add following line in the configuration file of the kernel module loader (normally /etc/modules.conf or /etc/conf.modules):

alias net−pf−10 ipv6 # automatically load IPv6 module on demand

It's also possible to disable automatically loading of the IPv6 module using following line

alias net−pf−10 off # disable automatically load of IPv6 module on demand

Testing your Configuration

After enabling IPv6 support, now you can verify the network interfaces on your Linux machine by typing ifconfig at the command prompt.

[root@ns root]# ifconfig

eth1 Link encap:Ethernet HWaddr 00:08:C7:CF:9D:0A

inet addr:202.141.40.26 Bcast:202.141.40.63 Mask:255.255.255.192

inet6 addr: 2001:e30:1400:1:208:c7ff:fecf:9d0a/64 Scope:Global

inet6 addr: fe80::208:c7ff:fecf:9d0a/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:933227 errors:6 dropped:0 overruns:0 frame:6

```
        TX packets:771601 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:102613930 (97.8 Mb)  TX bytes:213553354 (203.6 Mb)
        Interrupt:21 Base address:0x4000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:10671 errors:0 dropped:0 overruns:0 frame:0
        TX packets:10671 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:1163707 (1.1 Mb)  TX bytes:1163707 (1.1 Mb)
```

Note: The Global IPv6 address (2001:e30:1400:1:208:c7ff:fecf:9d0a/64 in the above example) is only configured if your host is connected to the network.

You can also test by pinging your IPv6 loopback address and your link local address, or some global address of your router interface.

```
[root@ns root]# ping6 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.084 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from ::1: icmp_seq=3 ttl=64 time=0.032 ms
64 bytes from ::1: icmp_seq=4 ttl=64 time=0.030 ms
```

64 bytes from ::1: icmp_seq=5 ttl=64 time=0.044 ms

64 bytes from ::1: icmp_seq=6 ttl=64 time=0.040 ms

--- ::1 ping statistics ---

6 packets transmitted, 6 received, 0% packet loss, time 4998ms

rtt min/avg/max/mdev = 0.030/0.043/0.084/0.020 ms

[root@ns root]# ping6 2001:0e30:1400:1::5

PING 2001:0e30:1400:1::5(2001:e30:1400:1::5) 56 data bytes

64 bytes from 2001:e30:1400:1::5: icmp_seq=1 ttl=64 time=1.08 ms

64 bytes from 2001:e30:1400:1::5: icmp_seq=2 ttl=64 time=0.418 ms

64 bytes from 2001:e30:1400:1::5: icmp_seq=3 ttl=64 time=0.331 ms

64 bytes from 2001:e30:1400:1::5: icmp_seq=4 ttl=64 time=0.343 ms

64 bytes from 2001:e30:1400:1::5: icmp_seq=5 ttl=64 time=0.379 ms

--- 2001:0e30:1400:1::5 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4002ms

rtt min/avg/max/mdev = 0.331/0.510/1.083/0.289 ms

If you try to ssh over IPv6, you'll see something like this:

[root@ns root]# ssh -6 2001:e30:1400:1:210:b5ff:feaa:88d7

root@2001:e30:1400:1:210:b5ff:feaa:88d7's password:

Last login: Wed Sep 1 15:32:02 2004 from 2001:e30:1400:1:208:c7ff:fecf:9d0a

Supporting IPv6 in Windows

IPv6 is best supported on Windows XP. To install the IPv6 Protocol for Windows XP:

1. Log on to the computer running Windows XP as Administrator.
2. Open a command prompt.
3. At the command prompt, type ipv6 install.

```
c:\ > ipv6 install
```

You can check your network configuration using the command **ipconfig** on Command Prompt and see if your machine has got an IPv6 address:

```
c:\ > ipconfig
```

Configuring IPv6 Proxy

IPv6 proxy is supported by Mozilla for Linux and Mozilla & Mozilla Firefox for Windows.

Download and install the browser suitable for your machine and configure the following in the proxy server settings:

Proxy Server: proxy.ipv6.iitk.ac.in

Port: 80

IPv6 Deployment Challenges and Risks

IPv6 deployment faces a number of challenges, including:

1. The IPv6 costs and risks
2. The fact that NAT is required to incrementally deploy IPv6 yet appears to eliminate the need for IPv6, and 3) the inability to really use the IPv6 features effectively during incremental deployment.

One view is that IPv6 should be first deployed at the enterprise level. However, to deploy IPv6 in some portion of the corporate network would require network address translation between the IPv6 portion and the ``legacy'' IPv4 portion, including the rest of the Internet. However, given network address translation, it would be lower risk to instead deploy more IPv4 using a private address domain and thereby gain sufficient addresses for the immediate enterprise needs. This route would eliminate the risks of disrupting the end hosts, routers and multi-layer switches, and network management systems to upgrade to IPv6. NAT-based solutions are widely deployed and well-understood whereas IPv6 support is still largely experimental. Thus, it seems difficult to get IPv6 deployed initially in an enterprise network.

Another view is that IPv6 should be deployed first in the backbone of the Internet. Yet, this appears to expose the ISP to unjustified costs and risks. The backbone has relatively few nodes so it does not have the demand for addresses to compel going to IPv6. Moreover, the ISP would have to support both IPv4 and IPv6 unless its customers simultaneously convert. (Dual-stack mechanisms and tunneling consume extra network and human resources over supporting just IPv4.) Finally, the ISP would have to provide backbone routers with adequate performance. However, there is no existing market for such products and relatively little investment in this direction because there is no significant amount of IPv6 traffic. So, it is not clear where and when an ISP would get these routers from even it decided to convert to IPv6.

Yet another view is that IPv6 might be widely deployed by some wireless service such as cellular phones. However, this move would incur higher packet overhead unless header compression can be very effective. Also, the average packet size with wireless tends to be smaller, both because the technology and because voice uses small packets. Moreover, a key challenge with wireless is dealing with many units collecting in the same cell, whether they are cell phones, wireless appliances in the home or other wireless mobile devices. If IPv6 does increase the packet overhead significantly, it effectively reduces the maximum number of units that can be served per cell in the worst case, thus increasing the cost. Moreover, wireless only has to transmit to the nearest (wired) receiver, which offers an opportunity to translate the packets to another format for the wired infrastructure, as proposed in the widely supported WAP standard [19]. The arguments for IPv6 based on auto configuration may also be less compelling given that wireless devices have to authenticate themselves when entering a realm, giving ample mechanism and opportunity to do DHCP-like address assignment at that point. Finally, having fixed IPv6 addresses for mobile hosts is mostly interesting to support mobile IP, yet mobile IP has received relatively little deployment to date, given the routing difficulties and solutions that exist at the higher level. Until mobile IP is more compelling and excessive header overhead is shown to not be an issue, or until another compelling reason is identified for IPv6 on cellular phones, it is hard to see IPv6 being deployed in this domain.

A final view is that some country such as China will so desperately need addresses that it would deploy IPv6. However, this scenario raises a number of issues. To communicate with the rest of the existing Internet, China would require sufficient (global) IPv4 addresses in any case for NAT-based communication from its internal IPv6 to these IPv4 addresses. However, if it has these addresses, it would be far easier to run the whole country behind a NAT boundary using IPv4 addresses, given that that would allow the use of existing routers, switches and host software. It seems inadvisable for a country with limited Internet expertise and industry to commit to the least proven technology and possibly be forced to largely develop its own products, especially with uncertain prospects of other markets for these products.

IPV6 introduces the following risks:

1. IPv6 introduces a privacy risk because it encodes information in the addresses, making this information externally visible. For instance, with IPv6, one can determine a company's ISP based on the addresses used by its hosts. IPv6 also makes every host that uses multiple ISPs effectively multi-homed. IPv6 addresses can also encode MAC addresses that can

reveal the manufacturers of the Ethernet interfaces in the hosts. These issues have already caught the attention of privacy groups.

2. IPv6 relies on ``renumbering" [6] for efficient routing to keep the mapping of address to topology reasonably compatible. It is reasonably considered a research issue because there is no prior system to the authors' knowledge that has proven this is in fact practical.
3. IPv6 also changes the way that options and IP fragmentation are handled. In particular, IPv6 disallows fragmentation at intermediate hops, making it even more difficult to use multicast efficiently in a highly diverse environment. Some networks impose fragmentation on large packets to provide delay guarantees for latency-sensitive traffic. This fragmentation may only come into play when such applications are running. It seems inappropriate to force a small MTU on a distant multicast source, for all receivers, just because a local low bandwidth link is carrying voice, for instance.
4. The large IPv6 header also introduces significant overhead and risk in some network settings. Besides the overhead in low bandwidth settings and/or risk that header compression will not be effective, the larger header may cause some applications with fixed packet sizes, like those tuned to Ethernet maximum packet size, to incur fragmentation at the IP level because of the larger header, a further deployment risk. IPv6 requires extensive changes to existing end-user host software and the network infrastructure of routers, switches, firewalls and network management. This IPv6 software and equipment is far less tested, less well-supported and far less cost-effective than the comparable IPv4 facilities.

Finally, early adopters risk being orphaned if IPv6 is not be widely deployed soon after they make the move, incurring the cost of backing out of IPv6 as well as the risks and costs of conversion. The lack of IPv6 deployment to date provides empirical support to the above concerns.

IPv6 and the NGN

The NGN is characterised by the following fundamental aspects:

- Packet-based transfer
- Support for a wide range of services, applications and mechanisms based on service building blocks (including real time/streaming/non-real time services and multi-media)
- Broadband capabilities with end-to-end QoS and transparency
- Interworking with legacy networks via open interfaces
- Generalised mobility
- Unfettered access by users to different service providers
- A variety of identification schemes which can be resolved to IP addresses for the purposes of routing in IP networks
- Unified service characteristics for the same service as perceived by the user
- Converged services between Fixed and Mobile networks
- Independence of service-related functions from underlying transport technologies
- Compliant with all Regulatory requirements, for example concerning emergency communications and security/privacy, etc.

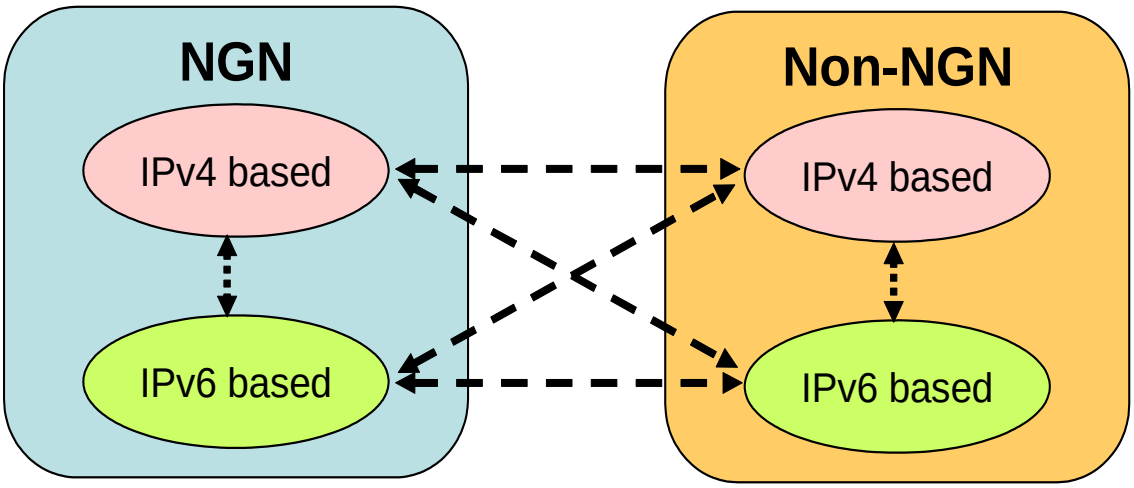
Relationship between NGN and IP

NGN, by definition, is identified an infrastructure using packet technologies.

- There is no specific mention which packet technology NGN should use, but generally assuming IP as a dominant packet technology today.
- There are also no specific statement to specify the version of IP such as 'version 4', 'version 6' or 'version 9,' but most parts of NGN related ITU-T RECs are mainly assumed 'version 4'

Definition of IPv6 based NGN

- IPv6-based NGN: This is a NGN which support addressing, routing protocol and mechanisms of IPv6
- IPv4-based NGN: This is a NGN which support addressing, routing protocol and mechanisms of IPv4
- IPv6-based Non-NGN: This is an IPv6 based packet network which is not comply with NGN
- IPv4-based Non-NGN: This is an IPv4 based packet network which is not comply with NGN



ITU-T Plans for IPv6 based NGN

- ITU-T SG13 has been developed framework recommendations of IPv6 based NGN.
- The concept and requirements of IPv6 based NGN will open one of the important gateway to use IPv6 technology/systems under the umbrella of NGN.
- According to the shortage of IPv4 based address which also impact to the NGN deployment, IPv6 should be used more widely.

- In addition, ‘Ubiquitous Networking’ will require more addresses and flexible but device/object targeted management which would be benefited from IPv6 features .