IPv6

The Next Generation Internet Protocol

Ing. Carlos Barcenilla / Universidad Tecnológica Nacional Facultad Regional La Plata
c.a.barcenilla@ieee.org

IPv6 Motivations

- Address space depletion.
- Router table explosion.
- Other protocol constraints.
  - Fragmentation Inefficiency
  - Control (ICMP useless messages)
  - Checksums

IPv6

Technical Criteria for IPng

- Scale
- Topological flexibility
- Performance
- Robust Service
- Straightforward transition
- Media independence
- Unreliable Datagram Service
- Configuration, Administration and Operation
- Secure Operation
- Unique Naming
- Access and Documentation
- Multicast
- Extensibility
- Network Service
- Mobility
- Control Protocol
- Private Networks

IPv6

Address Space Depletion

- IPv4 Address = 32 bits.
  - Class B addresses are exhausted.
- Short term solution: Supernetting of class C addresses.
  - One network receives several contiguous class C addresses (8*n class C networks).
  - Requires CIDR.
- Long term solution: IPv6 Address = 128 bits
  - Thousands of addresses per square meter of the earth's surface.

IPv6

Router Table Explosion

- Routing requires tables which have grown unmanageably large (more than 50000 entries at the core).
- To solve the problem under IPv4 a technique known as Classless Interdomain Routing is being used (CIDR).
- IPv6 addressing is Classless by nature.

IPv6

Changes from IPv4

- Expanded addressing capabilities.
  - Address size: 128 bits.
  - Improved scalability of multicast (scope field).
  - Anycast addresses.
  - No more broadcast addresses.
- Header format.
  - Some IPv4 fields were dropped or made optional.
  - Improved support for extensions an options.
- Flow labeling (QoS/real-time).
- Authentication and privacy capabilities.

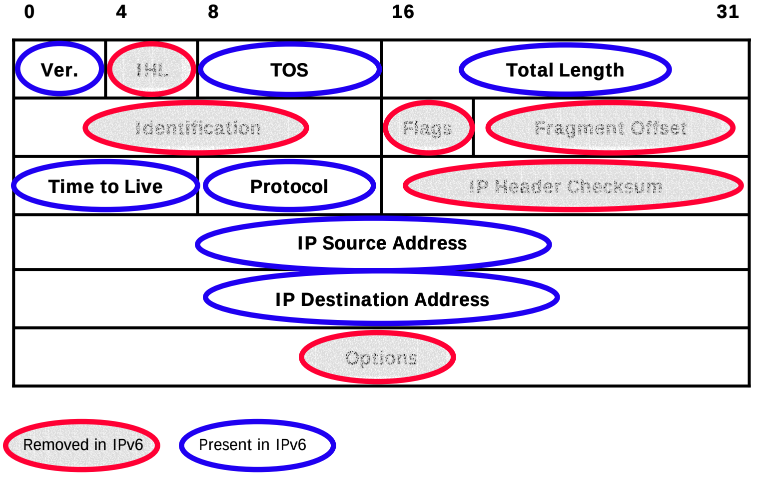
IPv6

IPv6 Terminology

- Node: A device that implements IPv6.
- Router: A node that forwards IPv6 packets not explicitly addressed to itself.
- Host: any node that is not a router.
- Upper layer: a protocol layer immediately above IPv6 (e.g. TCP, UDP, ICMP, OSPF and so on.)
- Link: A communication facility or medium over which nodes can communicate at the link level (e.g. Ethernet, Token Ring, Frame Relay, ATM and so on.)
- Neighbors: nodes attached to the same link.
- Interface: a node's attachment to a link.
- Address: an IPv6-layer identifier for an interface or a set of interfaces.
- Packet: an IPv6 header plus payload.
- Link MTU: the maximum transmission unit (max. packet size in octects) that can be conveyed over a link.
- Path MTU: The minimum link MTU of all the links in a path between a source node and a destination node.

IPv6

IPv4 Header Format



IPv6 Header Format

Version	Traffic Class	Flow Label	
Payload length		Next Header	Hop Limit
Source Address			
Destination Address			

- Version: 4-bit IP version number (6).
- Traffic Class: 8-bit traffic class field.
- Flow Label: 20-bit flow label.
- Payload Length: 16-bit unsigned integer.
- Next Header: 8-bit selector.
- Hop-Limit: 8-bit unsigned integer.
- Source Address: 128-bit address.
- Destination Address: 128-bit address.

Extension Headers

IPv6 header Next Header = TCP	TCP Header + Data

IPv6 header Next Header = Routing	Routing Header Next Header = TCP	TCP Header + Data

IPv6 header Next Header = Routing	Routing Header Next Header = Fragment	Fragment Header Next Header = TCP	Fragment of TCP Header + Data

IPv6

Extension Headers

- Extension headers are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or nodes in case of multicast).

- The exception is the hop-by-hop header which carries info that must be examined and processed by every node along the path, including the source and destination nodes.

IPv6

Extension Headers

- Extension headers must be processed strictly in the order they appear in the packet.

- If a node does not recognize a Next header value, it should discard the packet and send an ICMP Parameter Problem message.

- Each extension header should occur at most once, except for the destination options header which should occur at most twice.

Extension Headers

- Hop-by-hop options.
- Routing.
- Fragment.
- Destination options.
- Authentication.
- Encapsulating security payload.

Hop-by-Hop and Destination Options Headers: Options

- The Hop-by-Hop Options header and the Destination Options header carry a variable number of type-length-value (TLV) encoded “options”.

Option Type	Opt Data Len	Option Data

- Option Type: 8-bit identifier of the type of option.
- Opt Data Len: 8-bit unsigned integer.
- Option Data: Variable-length field.

- The sequence of options within a header must be processed strictly in the order they appear in the header.

IPv6

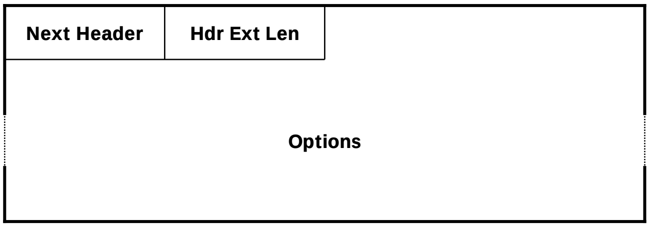
Hop-by-Hop and Destination Options Headers: Options

- For hop-by-hop and destination options headers.
- The two high order bits of option type means:
  - 00 – Skip over this option.
  - 01 – Discard the packet.
  - 10 – Discard the packet and send an ICMP Parameter Problem message.
  - 11 – Discard the packet and send an ICMP Parameter Problem message if the destination address was not multicast.
- The third highest-order bit specifies whether or not the Option Data can change en-route:
  - 0 – Option Data does not change en-route.
  - 1 – Option Data may change en-route.
- There are alignment restrictions.

IPv6

Hop-by-hop Options Header

- Carries additional information that must be examined by every node along a packet's delivery path.



- Next Header: 8-bit selector.
- Hdr Ext Len: 8-bit unsigned integer (Length of the header not including the first 8 octets).
- Options: variable-length field (contains one or more TLV-encoded options, the length of the complete header must be multiple of 8 octets long).
- The only options defined in RFC2460 are Pad1 and PadN (for alignment).

IPv6

Hop-by-hop Option
Router Alert Option

- Alerts transit routers to more closely examine the contents of an IP datagram.
- It is useful for situations where a datagram addressed to a particular destination contains information that may require special processing by routers along the path.

Option Type 000 00101(5)	Opt Data Len 00000010 (2)	Value

- Option Type: 5, means that nodes not recognizing this option type should skip over it and continue processing the header and that the option must not change en route.
- Opt Data Len: 2 bytes.
- Value:
  - 0: Datagram contains a Multicast Listener Discovery message.
  - 1: Datagram contains RSVP message.
  - 2: Datagram contains an Active Networks message.

IPv6

Destination Options Header

- This header is used to carry optional information that need be examined only by a packet's destination node(s).

Next Header	Hdr Ext Len	
Options		

- Next Header: 8-bit selector.
- Hdr Ext Len: 8-bit unsigned integer (Length of the header not including the first 8 octets).
- Options: variable-length field (contains one or more TLV-encoded options, the length of the complete header must be multiple of 8 octets long).
- The only options defined in RFC2460 are Pad1 and PadN (for alignment).

IPv6

Routing Header

- Used by an IPv6 source to list one or more intermediate nodes to be “visited” on the way to a packet’s destination.

Next Header	Hdr Ext Len	Routing Type	Segments Left
type-specific data			

- Next Header: 8-bit selector.
- Hdr Ext Len: 8-bit unsigned integer (Length of the header not including the first 8 octects).
- Routing Type: 8-bit identifier of a particular Routing header variant.
- Segments Left: 8-bit unsigned integer. Number of route segments remaining.
- Type-specific data: Variable-length field, of format determined by the Routing Type, and of length such that the complete Routing header is an integer multiple of 8 octects long.

IPv6

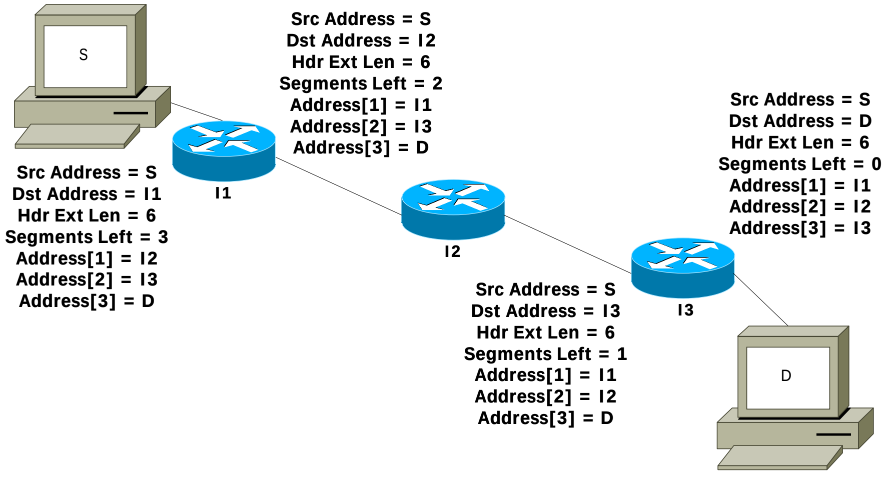
Type 0 Routing Header

Next Header	Hdr Ext Len	Routing Type=0	Segments Left
Reserved			
Address[1]			
Address[2]			
...			
Address[n]			

- Routing Type: 0.
- Segments Left: 8-bit unsigned integer. Number of route segments remaining, I.e., number of explicitly listed intermediate nodes still to be visited before reaching the final destination.
- Reserved: 32-bit reserved field. Initialized to zero for transmission; ignored on reception.
- Address[1..n]: Vector of 128-bit addresses, numbered 1 to n.

IPv6

Routing Header Example



IPv6

Type 0 Routing Header Processing Algorithm

if Segments Left = 0 { process next header in the packet }
else {
  n=Hdr Ext Len / 2 (number of addresses in the Routing Header)
  Segments Left = Segments Left – 1
  i = n - Segments Left (i: index of next address to be visited)
  swap the Destination Address and Address[i]
  if Hop Limit is <= 1 { send ICMP Time Exceeded}
  else {
    Hop Limit = Hop Limit – 1
    resubmit the packet to the IPv6 module for transmission to the new destination
  }
}

- RFC2460 contains a more detailed algorithm.

IPv6 Fragmentation

- The original, unfragmented packet consists of two parts.

Unfragmentable Part	Fragmentable Part

- The Unfragmentable Part consists of the IPv6 header plus any extension headers that must be processed by nodes en route to the destination.
- The Fragmentable Part consists of the rest of the packet.
- The Fragmentable Part of the original packet is divided into fragments, each, except possibly the last one, being an integer multiple of 8 octets long.
- Original packet:

Unfragmentable Part	first fragment	second fragment	......	last fragment

IPv6 Fragmentation

- Fragment packets:

Unfragmentable Part	Fragment Header	first fragment
Unfragmentable Part	Fragment Header	second fragment
⋮		
Unfragmentable Part	Fragment Header	last fragment

- Each fragment packet is composed of:
  - The Unfragmentable Part of the original packet.
  - A Fragment header.
  - The fragment itself.
- The lengths of the fragments must be chosen such that the resulting fragment packets fit within the path MTU.

IPv6 Fragment Header

- Is used by a source to send a packet larger than the path MTU to its destination. Unlike IPv4, fragmentation is only performed by source nodes.

Next Header	Reserved	Fragment Offset	Res	M
Identification				

- Next Header: 8-bit selector.
- Reserved: 8-bit reserved field.
- Fragment Offset: 13-bit unsigned integer. The offset, in 8-octect units, of the data following this header, relative to the start of the Fragmentable Part of the original packet.
- Res: 2-bit reserved field.
- M flag: 1 = more fragments; 0 = last fragment.
- Identification: 32 bits. The Identification must be different than any other fragmented packet sent recently with the same Source Address and Destination Address.

IPv6 Reassembly

- At the destination, fragment packets are reassembled into their original, unfragmented form:

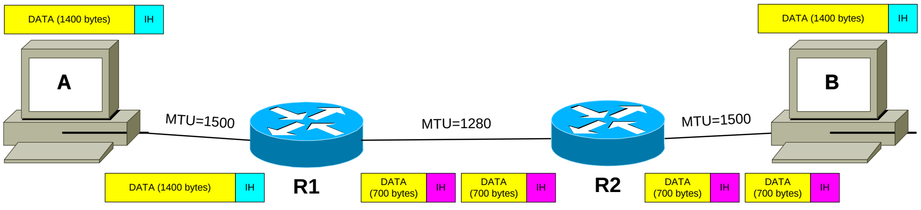
Unfragmentable Part	Fragmentable Part

- An original packet is reassembled only from fragment packets that have the same Source Address, Destination Address, and Fragment Identification.
- The Unfragmentable Part of the reassembled packet consists of all headers up to, but not including, the Fragment Header of the first fragment packet.
- The Fragmentable Part of the reassembled packet is constructed from the fragments following the Fragment headers in each of the fragment packets.

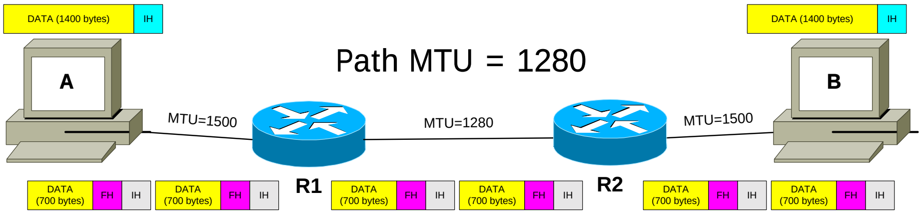
IPv6

IPv4 vs. IPv6 Fragmentation and Reassembly

- IPv4:



- IPv6:



IPv6

Packet Size

- IPv6 requires that every link in the internet have an MTU of 1280 octets or greater.
- From each link to which a node is directly attached, the node must be able to accept packets as large as that link's MTU.
- It is strongly recommended that IPv6 nodes implement Path MTU Discovery, in order to discover and take advantage of path MTUs greater than 1280 octets.
- In order to send a packet larger than a path's MTU, a node may use the IPv6 Fragment header.
- A node must be able to accept a fragmented packet that, after reassembly, is as large as 1500 octets.

Flow Labels

- The 20-bit Flow Label in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or “real-time” service.

- This aspect of IPv6 is still experimental, and may change.

- A flow is a sequence of packets sent from a particular source to a particular destination for which the source desires special handling by the intervening routers.

- There may be multiple active flows from a source to a destination, as well as traffic that is not associated with any flow.

- There is no requirement that all, or even most, packets belong to flows.

Traffic Classes

- The 8-bit Traffic Class field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets (e.g. “differentiated services”).

- General requirements:
  - The service interface to the IPv6 service within a node must provide a means for an upper-layer protocol to supply the value of the Traffic Class bits.
  - Nodes that support a specific use of the Traffic Class bits are permitted to change the value of those bits in packets that they originate, forward, or receive.
  - An upper-layer protocol must not assume that the value of the Traffic-Class bits in a received packet are the same as the value sent by the packet’s source.

Upper-Layer Checksums

- Any transport or other upper-layer protocol that includes the addresses from the IP header in its checksum computation must be modified for use over IPv6. The TCP/UDP “pseudo-header” for IPv6 is:

Source Address	
Destination Address	
Upper-Layer Packet Length	
Zero	Next Header

Upper-Layer Checksums

- If the IPv6 packet contains a Routing Header, the Destination Address in the pseudo-header is that of the final destination.
- The Next Header in the pseudo-header identifies the upper-layer protocol (e.g., 6 for TCP, or 17 for UDP).
- The Upper-Layer Packet Length in the pseudo-header is the length of the upper-layer header and data.
- Unlike IPv4, when UDP packets are originated by an IPv6 node, the UDP checksum is not optional.
- The IPv6 version of ICMP includes this pseudo-header in its checksum computation.

IPv6 Maximum Packet Lifetime

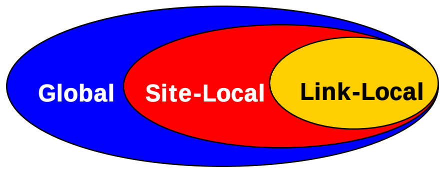
- Unlike IPv4, IPv6 nodes are not required to enforce maximum packet lifetime.
- That is the reason the IPv4 “Time to Live” field was renamed “Hop Limit” in IPv6.

IPv6 Maximum Upper-Layer Payload Size

- When computing the maximum payload size for upper-layer data, an upper-layer protocol must take into account the larger size of the IPv6 header relative to the IPv4 header.
- For example TCP MSS:
  - IPv4: MSS = Max. Packet Size – 40
    (20 octets for the minimum-length IPv4 header and 20 octets for the minimum-length TCP header)
  - IPv6: MSS = Max. Packet Size – 60
    (40 octets for the minimum-length IPv6 header and 20 octets for the minimum-length TCP header)

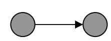
IPv6 Addressing Model

- IPv6 addresses of all types are assigned to interfaces, not nodes.
- All interfaces are required to have at least one link-local unicast address.
- A single interface may be assigned multiple ipv6 addresses of any type or scope.
- A subnet prefix is associated with one link. Multiple subnet prefixes may be assigned to the same link.
- Address size has been expanded to 128 bits.
  - Total: 340.282.366.920.938.463.463.374.607.431.768.211.456 addresses.
- Address scope can be: link-local, site-local or global.

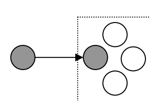


IPv6 Types of addresses

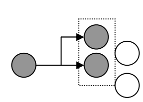
- Unicast.
  - An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.



- Anycast.
  - An Identifier for a set of interfaces. A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the “nearest”).



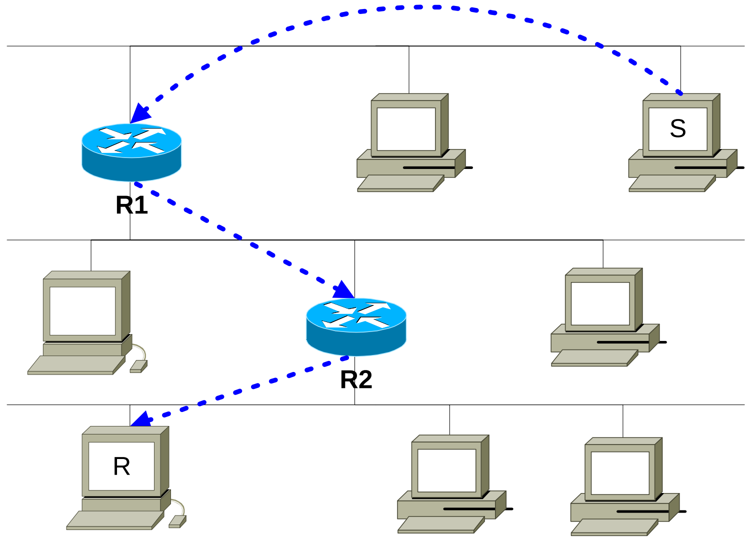
- Multicast.
  - An identifier for a set of interfaces. A packet sent to a multicast address is delivered to all interfaces identified by that address.



- There are no broadcast addresses in IPv6.

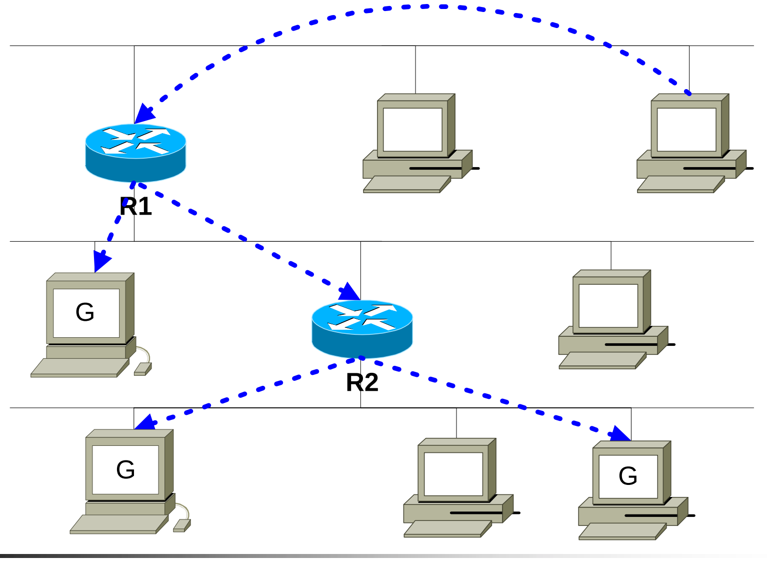
IPv6

Unicast



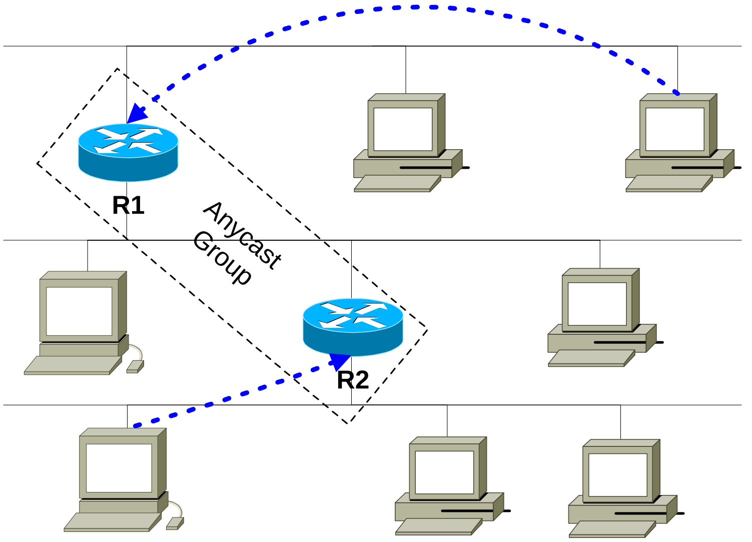
IPv6

Multicast



IPv6

Anycast



IPv6

Text Representation of Addresses

- Preferred form: x:x:x:x:x:x:x:x
  x: hex. Values of the eight 16 bit pieces of the address.

  Ex.: FEDC:ba98:7654:3210:FEDC:ba98:7654:3210
  1080:0:0:0:8:800:200c:417a.
- Syntax for compress the zeros:
  - "::" Indicate multiple groups of 16 bit zeros.
  - The "::" can only appear once in an address.

Ex.: 1080:0:0:0:8:800:200C:417A	1080::8:800:200C:417A
FF01:0:0:0:0:0:0:101	FF01::101
0:0:0:0:0:0:0:1	::1
0:0:0:0:0:0:0:0	::

Text Representation of Addresses

IPv6

- Mixed IPv4 and IPv6 form

x:x:x:x:x:x:d.d.d.d

x: hex d: decimal

Ex.:

0:0:0:0:0:0:13.1.68.3	::13.1.68.3
0:0:0:0:0:FFFF:129.144.52.38	::FFFF:129.244.52.38

Text Representation of Address Prefixes

IPv6

IPv6-address/prefix-length

- Where:
  - IPv6-address: an IPv6 address in any notation.
  - Prefix-length: specifies how many of the leftmost contiguous bits of the address comprise the prefix.

- Examples:
  - Node address: 12AB:0:0:CD31:123:4567:89AB:CDEF
  - Subnet: 12AB:0:0:CD30::/60
  - Node + Subnet: 12AB:0:0:CD31:123:4567:89AB:CDEF/60

IPv6 Address Type Representation

Allocation	Prefix (binary)	Fraction of Address Space
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP Allocation	0000 001	1/128
Reserved for IPX Allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Aggregatable Global Unicast Addresses	001	1/8
Unassigned	010	1/8
Unassigned	011	1/8
Unassigned	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link-Local Unicast Addresses	1111 1110 10	1/1024
Site-Local Unicast Addresses	1111 1110 11	1/1024
Multicast Addresses	1111 1111	1/256

IPv6 Address Type Representation

- Unicast addresses are distinguished from multicast addresses by the value of the high-order octet of the address: a value of FF identifies a multicast address.
- Anycast addresses are taken from the unicast address space, and are not syntactically distinguishable.

Unicast Addresses

IPv6

- Aggregatable with contiguous bitwise mask (like IPv4 CIDR).
- Forms:
  - Global aggregatable unicast address.
  - NSAP address.
  - IPX address.
  - Site-local.
  - Link-local.
  - IPv4-capable host address.

Unicast Addresses

IPv6

- At a minimum, a node may consider that unicast addresses have not internal structure.

128 bits
node address

- A slightly sophisticated host may additionally be aware of subnet prefix(es) for the link(s) it is attached to.

n bits	128-n bits
Subnet prefix	**Interface ID**

- Still more sophisticated hosts may be aware of other hierarchical boundaries in the unicast address.

IPv6

Interface Identifiers

- Interface identifiers in IPv6 unicast addresses
  - Used to identify interfaces on a link.
  - Are required to be unique on that link.
  - In many cases an interface's ID will be the same as that interface's link layer address.

- The same interface identifier may be used on multiple interfaces on a single node.

IPv6

The Unspecified Address

- The address 0:0:0:0:0:0:0:0 is called the *unspecified address*.

- It indicates the absence of an address.

- Ex.: In the Source Address field do any IPv6 packets sent by an initializing host before it has learnt its own address.

- The unspecified address must not be used as the destination address of IPv6 packets or in IPv6 Routing Headers.

The Loopback Address

- The unicast address 0:0:0:0:0:0:0:1 is called the *loopback address*. It may be used by a node to send an IPv6 packet to itself.
- It may never be assigned to any physical interface.
- Must not be used as the Source Address in IPv6 packets that are sent outside of a single node.
- A packet containing this address must never be forwarded by an IPv6 Router.

IPv4-compatible IPv6 Address

- The IPv6 transition mechanisms include a technique for hosts and routers to dynamically tunnel IPv6 packets over IPv4 routing infrastructure. IPv6 nodes that utilize this technique are assigned special IPv6 unicast addresses that carry an IPv4 address in the low-order 32-bits.

80 bits	16 bits	32 bits
0000...0000	0000	IPv4 address

- Example: ::170.210.16.2

IPv6

IPv4-mapped IPv6 address

- This address is used to represent the addresses of IPv4-only nodes as IPv6 addresses.
- For example, an IPv6 host would use an IPv4-mapped IPv6 address to communicate with another host that only supports IPv4.

80 bits	16 bits	32 bits
0000..0000	FFFF	IPv4 address

- Example: ::FFFF:170.210.16.2

IPv6

Link-Local Addresses

- Link-Local addresses are designed to be used for addressing on a single link for purposes such as auto-address configuration, neighbor discovery, or when no routers are present.

10 bits	54 bits	64 bits
1111111010	0	Interface ID

- Routers must not forward any packets with link-local source or destination addresses to other links.
- Example: FE80::1234:5678:9ABC:DEF0

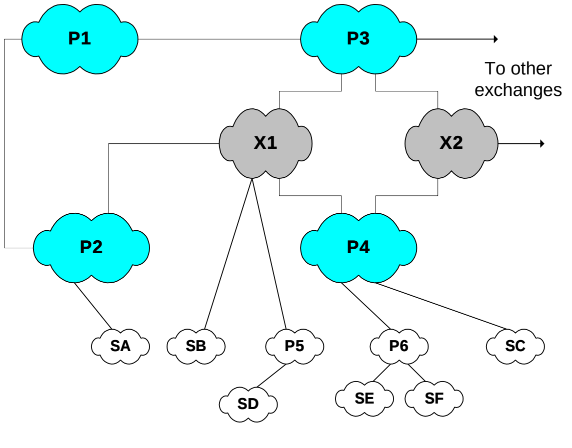
Site-Local Addresses

- Site-Local addresses are designed to be used for addressing inside a site without the need for a global prefix.

10 bits	38 bits	16 bits	64 bits
1111111011	0	Subnet ID	Interface ID

- Routers must not forward any packets with site-local source or destination addresses outside of the site.

Aggregatable Global Unicast Addresses Format



- P1-P4: long-haul providers
- P5-P6: Multiple levels of providers.
- SA-SF: Subscribers
- X1-X2: Exchanges which allocate IPv6 Addresses

- Designed to support both provider based aggregation and exchanges. Sites will have the choice to connect to either type of aggregation point.

IPv6

Aggregatable Global Unicast Addresses Format

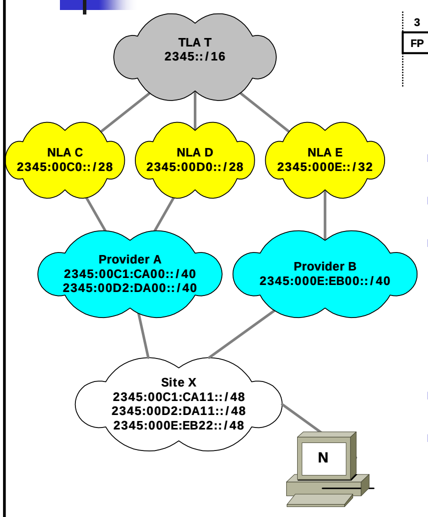
- Aggregatable addresses are organized into a three level hierarchy:
  - Public Topology
  - Site Topology
  - Interface Identifier

3	13	8	24	16	64 bits
FP	TLA ID	RES	NLA ID	SLA ID	Interface ID
Public Topology				Site Topology	Interface Identifier

- FP: Format Prefix (001) for Aggregatable Global unicast Addresses.
- TLA ID: Top-Level Aggregation Identifier
- RES: Reserved for future use.
- NLA ID: Next-Level Aggregation Identifier.
- SLA ID: Site-Level Aggregation Identifier.
- INTERFACE ID: Interface Identified.

IPv6

Aggregatable Global Unicast Addresses Example



2345:00C1:CA11:0001:1234:5678:9ABC:DEF0
2345:00D2:DA11:0001:1234:5678:9ABC:DEF0
2345:000E:EB22:0001:1234:5678:9ABC:DEF0

3	13	8	24	16	64 bits
FP	TLA ID	RES	NLA ID	SLA ID	Interface ID
Public Topology				Site Topology	Interface Identifier

2345:00**C1**:**CA11**:**0001**:**1234**:**5678**:**9ABC**:**DEF0**

- FP: **001** (binary) [2000::/3]
- TLA T ID: **0345** (hex) [2345::/16]
- NLA ID: **C1CA11** (hex) [2345:00C1:CA11::/48]
  - NLA C: C (hex) [2345:00C0::/28]
  - Provider A: 1CA (hex) [2345:00C1:CA00::/40]
  - Site X: 11 (hex) [2345:00C1:CA11::/48]
- SLA ID: **0001** (hex) [2345:00C1:CA11:0001::/64]
- Interface ID: **1234**:**5678**:**9ABC**:**DEF0** (hex)

IPv6

Anycast Addresses

- Are assigned to more than one interface (typically belonging to different nodes).
- A packet sent to an anycast address is routed to the “nearest” interface having the address, according to the routing protocols’ measure of distance.
- Are allocated from the unicast address space.
- Are syntactically indistinguishable from unicast addresses.
- Must not be used as a source address.
- May only be assigned to routers, not hosts.

IPv6

Required Anycast Address

- The Subnet-Router anycast address is predefined.

n bits	128-n bits
subnet prefix	**0000……………..0000**

- The “subnet prefix” in an anycast address is the prefix which identifies a specific link.
- Packets sent to the Subnet-Router anycast address will be delivered to one router on the subnet. All routers are required to support this addresses for the subnets which they have interfaces.

IPv6

Multicast Addresses

- An IPv6 multicast address is an identifier for a group of nodes. A node may belong to any number of multicast groups.

8 bits	4 bits	4 bits	112 bits
11111111	flgs	scop	group ID

- 11111111 at the start of the address identifies the address as being a multicast address.
- flgs is a set of 4 flags: 000T
  - T = 0 indicates a permanently-assigned (“well-known”) multicast address.
  - T = 1 indicates a non-permanently-assigned (“transient”) multicast address.
- scop is a 4-bit multicast scope value to limit the scope of the group:
  - 1: node-local scope
  - 2: link-local scope
  - 5: site-local scope
  - 8: organization-local scope
  - E: global scope
- group ID identifies the multicast group.

IPv6

Multicast Addresses

- The “meaning” of a permanently-assigned multicast address is independent of the scope value. For example, if the “NTP servers group” is assigned a permanent multicast address with a group ID of 101 (hex), then:
  - FF01::101 means all NTP servers on the same node as the sender.
  - FF02::101 means all NTP servers on the same link as the sender.
  - FF05::101 means all NTP servers at the same site as the sender.
  - FF0E::101 means all NTP servers in the internet
- Multicast addresses must not be used as source addresses in IPv6 packets or appear in any routing header.

Pre-Defined multicast Addresses

- Reserved: FF0x:: (x: hex digit)
- All nodes:
  - FF01::1 (node-local)
  - FF02::1 (link-local)
- All routers:
  - FF01::2 (node-local)
  - FF02::2 (link-local)
  - FF05::2 (site-local)
- Solicited-node Address: FF02::1:FFxx:xxxx
  - This address is formed by taking the low-order 24 bits of the address (unicast or anycast) and appending those bits to the prefix FF02::1:FF00:0000/104
  - Example: for the IPv6 address 3FFE:3800:FFFB::BD12:3456 the solicited-node multicast address is: FF02::1:FF12:3456.

Node Required Addresses (Host)

- Link-local address for each interface.
- Assigned Unicast Addresses.
- Loopback Addresses.
- All-nodes Multicast Addresses.
- Solicited-node Multicast Addresses for each of its assigned unicast and anycast addresses.
- Multicast Addresses of all other groups to which the host belongs.

- Example:

Address	Type
fe80::250:56ff:fe8a:0	IPv6 link-local
ff02::1:ff8a:0	Solicited-Node Multicast
::1	Loopback
ff02::1	All-nodes Multicast

IPv6

Node Required Addresses (Router)

All the host required addresses plus:

- Subnet-router anycast addresses for the interfaces it is configured to act as a router on.
- All other anycast addresses with which the router has been configured.
- All-routers multicast addresses.
- Multicast addresses of all other groups to which the router belongs.

- Example:

Address	Type
fe80::260:8ff:fe14:7861 / ff02::1:ff14:7861	IPv6 link-local / Solicited-Node Multicast
3ffe:3800:fffb:2001::1 / ff02::1:ff00:1	IPv6 global / Solicited-Node Multicast
::1	Loopback
ff02::1	All-nodes Multicast
ff02::2	All-routers Multicast

IPv6

IPv6 over Ethernet

- IPv6 packets are transmitted in standard Ethernet frames.
- The Ethernet header contains:
  - Destination and Source Ethernet addresses.
  - Ethernet type code (86DD hexadecimal).
- The data field contains:
  - The IPv6 header.
  - The payload.
  - Padding octets to meet the minimum frame size for the Ethernet link (if needed).
- The Maximum Transmission Unit (MTU) for Ethernet is 1500 octets. This size may be reduced by:
  - A Router Advertisement.
  - Manual configuration of each node.

IPv6 over Ethernet Frame Format

Destination	Source	Type 86DD	Data	FCS	
6	6	2	46-1500	4	Octets

IPv6 Header and Payload

64-1518

- Preamble: 1010...1011
- Destination: Destination Node Address
- Source: Source Node Address
- Type: Higher Layer protocol Type
- Data: Higher Layer Information
- FCS: Frame Check Sequence (CRC-32)

IPv6 over Ethernet Stateless Autoconfiguration

- The Interface Identifier for an Ethernet interface is based on the EUI-64 identifier derived from the interface's built-in 48-bit IEEE 802 address.
- The Interface Identifier is then formed from the EUI-64 by complementing the "Universal/Local" (U/L) bit, which is the next-to-lowest order bit of the first octet of the EUI-64.
- For example, the Interface Identifier for an Ethernet interface whose built-in address is, in hexadecimal,

34-56-78-9A-BC-DE
would be
36-56-78-FF-FE-9A-BC-DE

- An IPv6 address prefix used for stateless autoconfiguration of an Ethernet interface must have a length of 64 bits.

IPv6 over Ethernet Link-local Address

- The IPv6 link-local address for an Ethernet interface is formed by appending the Interface Identifier, to the prefix FE80::/64.

128 bits

1111111010	0	Interface Identifier from Ethernet Address	
10 bits	54 bits	64 bits	Bits

- Example:
  - EUI-48 Ethernet Address: 00:50:56:d9:88:3f
  - EUI-64 Ethernet Address: 00:50:56:**ff:fe**:d9:88:3f
  - Interface Identifier: 0**2**:50:56:ff:fe:d9:88:3f
  - Link-local Address: **fe80**::250:56ff:fed9:883f

IPv6 over Ethernet Unicast address mapping

- The Neigbbor Discovery Source/Target Link-layer Address option has the following form when the link layer is Ethernet.

Type	Lenght	Ethernet Address
8 bits	8 bits	48 bits

- Type:
  - 1 for Source Link-layer address
  - 2 for Target Link-layer address
- Length: 1 (in units of 8 octets)
- Ethernet Address:The 48 bit Ethernet IEEE 802 address, in canonical bit order

IPv6 over Ethernet
Multicast address mapping

- An IPv6 packet with a multicast destination address DST, consisting of the sixteen octets DST[1] through DST[16], is transmitted to the Ethernet multicast address whose first two octets are the value 3333 hexadecimal and whose last four octets are the last four octets of DST.

00110011 (33)	00110011 (33)	DST[13]	DST[14]	DST[15]	DST[16]
8 bits	8 bits	8 bits	8 bits	8 bits	8 bits

- Examples:
  - IPv6 Solicited-Node Multicast Address: ff02:1::1:**ffd9:883f**
    Ethernet Link-Layer Multicast Address: 33:33:**ff:d9:88:3f**
  - IPv6 All nodes Multicast Address: ff02:1::**1**
    Ethernet Link-Layer Multicast Address: 33:33:**00:00:00:1**

Internet Control Message Protocol (ICMPv6)

- ICMPv6 is used by IPv6 nodes to report errors encountered in processing packets, and to perform other internet-layer functions such as diagnostics.

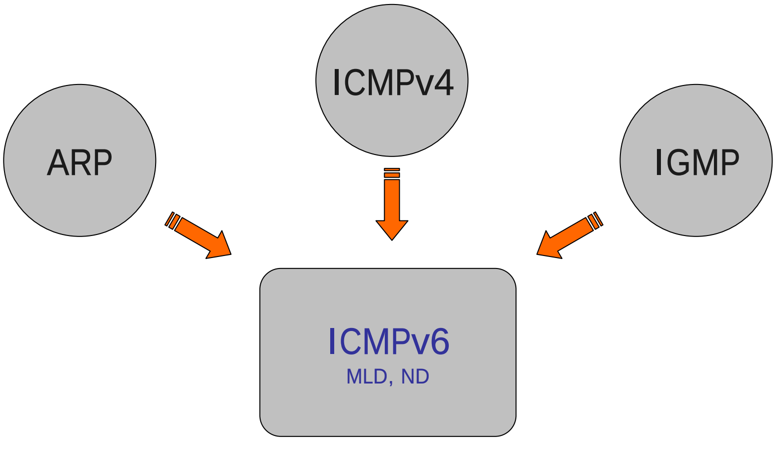
- ICMPv6 is an integral part of IPv6 and must be fully implemented by every IPv6 node.

- ICMPv6 messages are grouped into two classes: error messages and informational messages.

- High-order bit of the message Type:
  - 0: Error messages (Type: 0 to 127).
  - 1: Informational messages (Type: 128 to 255).

IPv6

Integration of protocols in ICMPv6



IPv6

ICMPv6 Messages

- ICMPv6 error messages:
  - 1 Destination Unreachable.
  - 2 Packet Too Big.
  - 3 Time Exceeded.
  - 4 Parameter Problem.
- ICMPv6 informational messages:
  - 128 Echo Request.
  - 129 Echo Reply.

ICMPv6 Message General Format

- Every ICMPv6 message is preceded by an IPv6 header and zero or more IPv6 extension headers. The ICMPv6 header is identified by a Next Header value of 58 in the immediately preceding header.

8	8	16 bits
Type	**Code**	**Checksum**

- Type: indicates the type of the message.
- Code: depends on the message type.
- Checksum: is used to detect data corruption in the ICMPv6 message and parts of the IPv6 header.

ICMPv6 Message Processing Rules

- If an ICMPv6 error message of unknown type is received, it must be passed to the upper layer.
- If an ICMPv6 informational message of unknown type is received, it must be silently discarded.
- Every ICMPv6 error message includes as much of the IPv6 offending packet as will fit without making the error message packet exceed the minimum IPv6 MTU.
- In those cases where the internet-layer protocol is required to pass an ICMPv6 message to the upper layer process, the upper-layer protocol type is extracted from the original packet and used to select the appropriate upper-layer process to handle the error.

IPv6

ICMPv6
Message Processing Rules

- An ICMPv6 error message must not be sent as a result of receiving:
  - An ICMPv6 error message.
  - A packet destined to an IPv6 multicast address*.
  - A packet sent as a link-layer multicast*.
  - A packet sent as a link-layer broadcast*.
  - A packet whose source address does not uniquely identify a single node (e.g. Unspecified Address, Multicast address, Anycast address).

*Exceptions: the Packet too Big message, Parameter Problem message reporting an unrecognized option with Option type highest-order two bits set to 10.

IPv6

ICMPv6
Message Processing Rules

- In order to limit the bandwidth and forwarding costs incurred sending ICMPv6 error messages, an IPv6 node must limit the rate of ICMPv6 error messages it sends.

  There are a variety of ways of implementing this function, e.g.:

  - Timer-based (limiting the rate of transmission of error messages to a given source or to any source to at most once every T milliseconds).

  - Bandwidth-based (for example, limiting the rate at which error messages are sent from a particular interface to some fraction of the attached link's bandwidth)

ICMPv6 Error Message: Destination Unreachable

- A Destination Unreachable message should be generated by a router or by the IPv6 layer in the originating node, in response to a packet that cannot be delivered to its destination address for reasons other than congestion.

8	8	16 bits
Type	**Code**	**Checksum**
Unused		
As much of invoking packet as will fit without ICMPv6 packet exceeding the minimum IPv6 MTU		

- IPv6 Destination Address: Copied from the Source Address field of the invoking packet.
- Type: 1
- Code: 0 – no route to destination 1 – communication with dest. prohibited
  3 – address unreachable 4 – port unreachable
- Unused: Must be initialized to zero by the sender and ignored by the receiver.

ICMPv6 Error Message: Packet Too Big

- Must be sent by a router in response to a packet that it cannot forward because the packet is larger than the MTU of the outgoing link. The information in this message is used as part of the Path MTU Discovery process.

8	8	16 bits
Type	**Code**	**Checksum**
MTU		
As much of invoking packet as will fit without ICMPv6 packet exceeding the minimum IPv6 MTU		

- IPv6 Destination Address: Copied from the Source Address field of the invoking packet.
- Type: 2
- Code: Set to 0 (zero) by the sender and ignored by the receiver.
- MTU: The Maximum Transmission Unit of the next-hop link.

ICMPv6 Error Message: Time Exceeded Message

- If a router receives a packet with a Hop Limit of zero, or a router decrements a packet's Hop Limit to zero, it must discard the packet and send an ICMPv6 Time Exceeded message with Code 0 to the source of the packet. This indicates either a routing loop or too small an initial Hop Limit Value.

8	8	16 bits
Type	**Code**	**Checksum**
Unused		
As much of invoking packet as will fit without ICMPv6 packet exceeding the minimum IPv6 MTU		

- IPv6 Destination Address: Copied from the Source Address field of the invoking packet.
- Type: 3
- Code: 0 – hop limit exceeded in transit 1 – fragment reassembly time exceeded
- Unused: Must be initialized to zero by the sender and ignored by the receiver.

ICMPv6 Error Message: Parameter Problem

- If an IPv6 node finds a problem with a field in the IPv6 header or extension headers such that it cannot complete processing the packet, it must discard the packet and should send an ICMPv6 Parameter problem message to the packet's source, indicating the type and location of the problem.

8	8	16 bits
Type	**Code**	**Checksum**
Pointer		
As much of invoking packet as will fit without ICMPv6 packet exceeding the minimum IPv6 MTU		

- IPv6 Destination Address: Copied from the Source Address field of the invoking packet.
- Type: 4
- Code: 0 – erroneous header field 1 – unrecognized Next Header type
  2 – unrecognized IPv6 option
- Pointer: Identifies the octet offset within the invoking packet where the error was detected.

IPv6

ICMPv6 Informational Message: Echo Request

- Every node must implement an ICMPv6 Echo responder function that receives Echo Requests and sends corresponding Echo Replies.

8	8	16 bits	
Type	**Code**	**Checksum**	
Identifier		**Sequence Number**	
Data...			

- IPv6 Destination Address: Any legal IPv6 address.
- Type: 128
- Code: 0
- Identifier: An identifier to aid in matching Echo Replies to this Echo Request. May be zero.
- Sequence Number: A sequence number to aid in matching Echo Replies to this Echo Request. May be zero.
- Data: Zero or more octets of arbitrary data.

IPv6

ICMPv6 Informational Message: Echo Reply

- Every node must implement an ICMPv6 Echo responder function that receives Echo Requests and sends corresponding Echo Replies. The data received in the ICMPv6 Echo Request message must be returned entirely and unmodified in the ICMPv6 Echo Reply message.

8	8	16 bits	
Type	**Code**	**Checksum**	
Identifier		**Sequence Number**	
Data...			

- IPv6 Destination Address: Copied from the Source Address field of the invoking Echo Request Packet.
- Type: 129
- Code: 0
- Identifier: The identifier from the invoking Echo Request message.
- Sequence Number:The sequence number from the invoking Echo Request message.
- Data: The data from the invoking Echo Request message.

ICMPv6: Security Considerations

- ICMP protocol packet exchanges can be authenticated using the IP Authentication Header. A node should include an Authentication Header when sending ICMP messages if a security association for use with the IP Authentication header exists for the destination address.
- Received Authentication Headers in ICMP packets must be verified for correctness and packets with incorrect authentication must be ignored and discarded.

Neighbor Discovery

- Nodes use Neighbor Discovery (ND) to determine the link-layer addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid.
- Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf.
- Nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed link-layer addresses.
- When a router or the path to a router fails, a host actively searches for functioning alternates.

IPv6 Neighbor Discovery Features

- This protocol solves a set of problems related to the interaction between nodes attached to the same link:
  - Router Discovery: How hosts locate routers that reside on an attached link.
  - Prefix Discovery: How hosts discover the set of address prefixes that define which destinations are on-link for an attached link.
  - Parameter Discovery: How a node learns such link parameters as the link MTU or such Internet parameters as the hop limit value to place in outgoing packets.
  - Address Autoconfiguration: How nodes automatically configure an address for an interface.
  - Address resolution: How nodes determine the link-layer address of an on-link destination (e.g., a neighbor) given only the destination's IP address.

IPv6 Neighbor Discovery Features

- Next-hop determination: The algorithm for mapping an IP destination address into the IP address of the neighbor to which traffic for the destination should be sent. The next-hop can be a router or the destination itself.
- Neighbor Unreachability Detection: How nodes determine that a neighbor is no longer reachable.
- Duplicate Address Detection: How a node determines that an address it wishes to use is not already in use by another node.
- Redirect: How a router informs a host of a better first-hop node to reach a particular destination.

IPv6 Neighbor Discovery Messages

- Neighbor Discovery defines five different ICMP packet types:
  - Router Solicitation: When an interface becomes enabled, hosts may send out Router Solicitations that request routers to generate Router Advertisements immediately rather than at their next scheduled time.
  - Router Advertisement: Routers advertise their presence together with various link and Internet parameters either periodically, or in response to a Router Solicitation message. Router Advertisements contain prefixes that are used for on-link determination and/or address configuration, a suggested hop limit value, etc.

IPv6 Neighbor Discovery Messages

- Neighbor Solicitation: Sent by a node to determine the link-layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link-layer address. Neighbor Solicitations are also used for Duplicate Address Detection.
- Neighbor Advertisement: A response to a Neighbor Solicitation message. A node may also send unsolicited Neighbor Advertisements to announce a link-layer address change.
- Redirect: Used by routers to inform hosts of a better first hop for a destination.

IPv6

Neighbor Discovery (Additional Features)

- Neighbor Discovery also handles the following situations:
  - Link-layer address change: A node that knows its link-layer address has changed can multicast a few (unsolicited) Neighbor Advertisement packets to all nodes to quickly update cached link-layer addresses that have become invalid. The Neighbor Unreachability Detection algorithm ensures that all nodes will reliably discover the new address, though the delay may be somewhat longer.
  - Inbound load balancing: Nodes with replicated interfaces may want to load balance the reception of incoming packets across multiple network interfaces on the same link. Such nodes have multiple link-layer addresses assigned to the same interface. For example, a single network driver could represent multiple network interface cards as a single logical interface having multiple link-layer addresses.

IPv6

Neighbor Discovery (Additional Features)

- Neighbor Discovery also handles the following situations:
  - Anycast addresses: Anycast addresses identify one of a set of nodes providing an equivalent service, and multiple nodes on the same link may be configured to recognize the same Anycast address. Neighbor Discovery handles anycasts by having nodes expect to receive multiple Neighbor Advertisements for the same target. All advertisements for anycast addresses are tagged as being non-Override advertisements. This invokes specific rules to determine which of potentially multiple advertisements should be used.
  - Proxy advertisements: A router willing to accept packets on behalf of a target address that is unable to respond to Neighbor Solicitations can issue non-Override Neighbor Advertisements.

Comparison with IPv4

- The IPv6 Neighbor Discovery protocol corresponds to a combination of the IPv4 protocols ARP, ICMP Router Discovery, and ICMP Redirect. In IPv4 there is no generally agreed upon protocol or mechanism for Neighbor Unreachability Detection.
- Router Discovery is part of the base protocol set.
- Router advertisements and Redirects carry link-layer addresses; no additional packet exchange is needed to resolve the router's link-layer address.
- Router advertisements carry prefixes for a link; there is no need to have a separate mechanism to configure the "netmask".
- Router advertisements enable Address Autoconfiguration.
- Routers can advertise an MTU for hosts to use on the link.
- Address resolution multicasts are "spread" over 4 billion (2^32) multicast addresses greatly reducing address resolution related interrupts on nodes other than the target. Moreover, non-IPv6 machines should not be interrupted at all.

Comparison with IPv4

- Multiple prefixes can be associated with the same link.
- Unlike IPv4, the recipient of an IPv6 redirect assumes that the new next-hop is on-link.
- Neighbor Unreachability Detection (NUD) is part of the base significantly improving the robustness of packet delivery in the presence of failing routers, partially failing or partitioned links and nodes that change their link-layer addresses.
- Unlike ARP, ND detects half-link failures (using NUD) and avoids sending traffic to neighbors with which two-way connectivity is absent.
- Unlike in IPv4 Router Discovery the Router Advertisement messages do not contain a preference field. The NUD will detect dead routers and switch to a working one.
- The use of link-local addresses to uniquely identify routers makes it possible for hosts to maintain the router associations in the event of the site renumbering to use new global prefixes.

IPv6

Router Solicitation Message Format

- Hosts send Router Solicitations in order to prompt routers to generate Router Advertisements quickly.

8	8	16 bits
Type	**Code**	**Checksum**
Reserved		
Options ...		

IP Fields:
- Source Address: An IP address assigned to the sending interface, or the unspecified address if no address is assigned to the sending interface.
- Destination Address: Typically the all-routers multicast address.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender SHOULD include this header.

ICMP Fields:
- Type: 133
- Code: 0
- Valid Options:
  - Source link-layer address: The link-layer address of the sender, if known.

IPv6

Router Advertisement Message Format

- Routers send out Router Advertisement message periodically, or in response to a Router Solicitation.

8	2	6	16 bits
Type	**Code**		**Checksum**
Cur Hop Limit	**M O**	**Reserved**	**Router Lifetime**
Reachable Time			
Retrans Timer			
Options ...			

IP Fields:
- Source Address: Must be the link-local address assigned to the interface from which this message is sent.
- Destination Address: Typically the Source Address of an invoking Router Solicitation or the all-nodes multicast address.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

IPv6

Router Advertisement Message Format

ICMP Fields:

- Type: 134
- Code: 0
- Cur Hop Limit: The default value that should be placed in the Hop Count field of the IP header for outgoing IP packets.
- Managed address configuration flag (M): When set, hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration.
- Other stateful configuration flag (O): When set, hosts use the administered (stateful) protocol for autoconfiguration of other (non-address) information.
- Router Lifetime: The lifetime associated with the default router in units of seconds. A Lifetime of 0 indicates that the router is not a default router and should not appear on the default router list.
- Reachable Time: The time, in milliseconds, that a node assumes a neighbor is reachable after having received a reachability confirmation.
- Retrans Timer: The time, in milliseconds, between retransmitted Neighbor Solicitation messages.
- Valid Options:
  - Source link-layer address: The link-layer address of the sender, if known.
  - MTU: Should be sent on links that have a variable MTU.
  - Prefix Information: These options specify the prefixes that are on-link and/or are used for address autoconfiguration.

IPv6

Neighbor Solicitation Message Format

- Nodes send Neighbor Solicitations to request the link-layer address of a target node while also providing their own link-layer address to the target. Neighbor Solicitations are multicast when the node needs to resolve an address and unicast when the node seeks to verify the reachability of a neighbor.

8	8	16 bits
Type	**Code**	**Checksum**
Reserved		
Target Address (128 bits)		
Options ...		

IP Fields:

- Source Address: Either an address assigned to the interface from which this message is sent or (if Duplicate Address Detection is in progress) the unspecified address.
- Destination Address: Either the solicited-node multicast address corresponding to the target address, or the target address.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

IPv6

Neighbor Solicitation Message Format

ICMP Fields:

- Type: 135
- Code: 0
- Target Address: The IP address of the target of the solicitation. It must not be a multicast address.
- Possible Options:
  - Source link-layer address: The link-layer address for the sender. Must not be included when the source IP address is the unspecified address. Otherwise, on link layers that have addresses this option must be included in multicast solicitations and should be included in unicast solicitations.

IPv6

Neighbor Advertisement Message Format

- A node sends Neighbor Advertisements in response to Neighbor Solicitations and sends unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly.

3	5	8	16 bits
Type		**Code**	**Checksum**
R S O	**Reserved**		
Target Address (128 bits)			
Options ...			

IP Fields:

- Source Address: An address assigned to the interface from which the advertisement is sent.
- Destination Address:
  - For solicited advertisements, the Source Address of an invoking Neighbor Solicitation or, if the solicitation's Source Address is the unspecified address, the all-nodes multicast address.
  - For unsolicited advertisements typically the all-nodes multicast address.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

IPv6

Neighbor Advertisement Message Format

ICMP Fields:

- Type: 136
- Code: 0
- Router flag (R): When set, indicates that the sender is a router. The R-bit is used by NUD to detect a router that changes to a host.
- Solicited flag (S): When set, indicates that the advertisement was sent in response to a NS from the Destination address. The S-bit is used as a reachability confirmation for NUD. It should not be set in solicited advertisements for anycast addresses and in solicited proxy advertisements. It should be set in other solicited advertisements and in unsolicited advertisements.
- Override flag (O): When set, indicates that the advertisement should override an existing cache entry and update the cached link-layer address. When it is not set the advertisement will not update a cached link-layer address though it will update an existing Neighbor Cache entry for which no link-layer address is known.
- Target Address: For solicited advertisements, the Target Address field in the NS message that prompted this advertisement. For an unsolicited advertisement, the address whose link-layer address has changed. The Target Address must not be a multicast address.
- Possible Options:
  - Target link-layer address: The link-layer address for the target, i.e., the sender of the advertisement. This option must be included on link layers that have addresses when responding to multicast solicitations. When responding to a unicast Neighbor Solicitation this option should be included.

IPv6

Redirect Message Format

- Routers send Redirect packets to inform a host of a better first-hop node on the path to a destination. Hosts can be redirected to a better first-hop router but can also be informed by a redirect that the destination is in fact a neighbor. The latter is accomplished by setting the ICMP Target Address equal to the ICMP Destination Address.

8	8	16 bits
Type	**Code**	**Checksum**
Reserved		
Target Address (128 bits)		
Destination Address (128 bits)		
Options ...		

IP Fields:

- Source Address: Must be the link-local address assigned to the interface from which this message is sent.
- Destination Address: The Source Address of the packet that triggered the redirect.
- Hop Limit: 255
- Authentication Header: If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.

IPv6 Redirect Message Format

ICMP Fields:

- Type: 137
- Code: 0
- Target Address: An IP address that is a better first hop to use for the ICMP Destination Address. When the target is the actual endpoint of communication, i.e., the destination is a neighbor, the Target Address field must contain the same value as the ICMP Destination Address field. Otherwise the target is a better first-hop router and the Target Address must be the router's link-local address so that hosts can uniquely identify routers.
- Destination Address: The IP address of the destination which is redirected to the target.
- Possible Options:
  - Target link-layer address: The link-layer address for the target. It should be included (if known). Note that on NBMA links, hosts may rely on the presence of the Target Link- Layer Address option in Redirect messages as the means for determining the link-layer addresses of neighbors. In such cases, the option must be included in Redirect messages.
  - Redirected Header: As much as possible of the IP packet that triggered the sending of the Redirect without making the redirect packet exceed 1280 octets.

IPv6 Conceptual Model of a Host Data Structures

- Neighbor Cache: A set of entries about individual neighbors to which traffic has been sent recently.
  - Neighbor's on-link unicast IP address (key)
  - Link-layer address
  - IsRouter flag
  - Pointer to any queued packets waiting for address resolution to complete
  - Reachability state (NUD)
  - Number of unanswered probes (NUD)
  - Time the next NUD event is scheduled to take place

Conceptual Model of a Host
Data Structures

IPv6

- Destination Cache: A set of entries about destinations to which traffic has been sent recently. Includes both on-link and off-link destinations. IT maps a destination IP address to the IP address of the next-hop neighbor.

- Prefix List: A list of the prefixes that define a set of addresses that are on-link.
  - Entries are created from information received in Router Advertisements.
  - Each entry has an associated invalidation timer value used to expire prefixes when they become invalid.
  - A special "infinity" timer value specifies that a prefix remains valid forever, unless a new (finite) value is received in a subsequent advertisement.

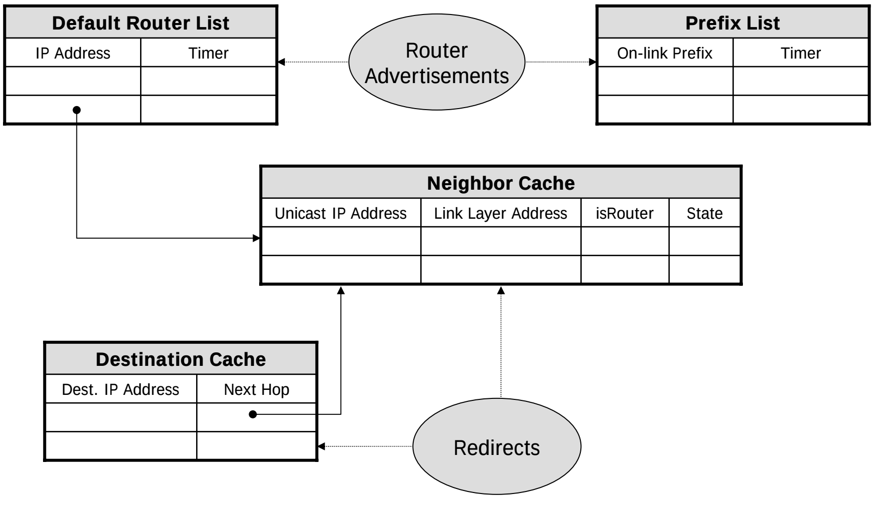
Conceptual Model of a Host
Data Structures

IPv6

- Default Router List: A list of routers to which packets may be sent.
  - Entries point to entries in the Neighbor Cache.
  - The algorithm for selecting a default router favors routers known to be reachable over those whose reachability is suspect.
  - Each entry also has an associated invalidation timer.

IPv6

Conceptual Model of a Host Data Structures



IPv6

Neighbor Cache Neighbor's Reachability State

- A key piece of information in the NC is a neighbor's reachability state, which is one of five possible values.
  - <u>INCOMPLETE:</u> Address resolution is in progress and the link-layer address of the neighbor has not yet been determined.
  - <u>REACHABLE:</u> The neighbor is known to have been reachable recently (within tens of seconds ago).
  - <u>STALE:</u> The neighbor is no longer known to be reachable but until traffic is sent to the neighbor, no attempt should be made to verify its reachability.
  - <u>DELAY:</u> The neighbor is no longer known to be reachable, and traffic has recently been sent to the neighbor. Rather than probe the neighbor immediately, however, delay sending probes for a short while in order to give upper layer protocols a chance to provide reachability confirmation.
  - <u>PROBE:</u> The neighbor is no longer known to be reachable, and unicast Neighbor Solicitation probes are being sent to verify reachability.

IPv6

Host Variables

- In addition the host maintains a number of variables, e.g.:
  - LinkMTU: The MTU of the link.
  - CurHopLimit: The default hop limit to be used when sending unicast IPv6 packets.
  - BaseReachableTime: A base value used for computing the random ReachableTime value.
  - ReachableTime: The time a neighbor is considered reachable after receiving a reachability confirmation.
  - RetransTimer: The time between retransmissions of Neighbor Solicitation messages to a neighbor when resolving the address or when probing the reachability of a neighbor.

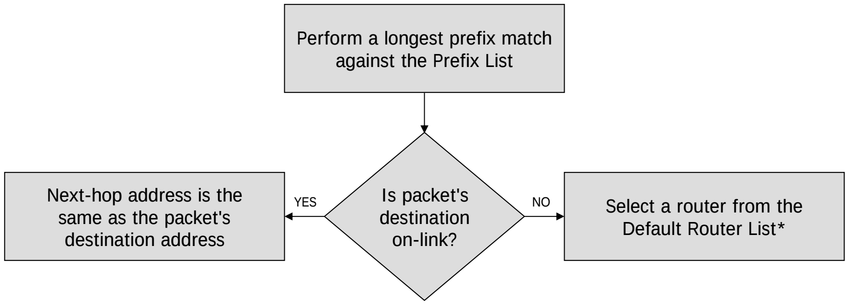
IPv6

Conceptual Sending Algorithm

- When sending a packet to a destination, a node uses a combination of the Destination Cache, the Prefix List, and the Default Router List to determine the IP address of the appropriate next hop, an operation known as "next-hop determination".
- Once the IP address of the next hop is known, the Neighbor Cache is consulted for link-layer information about that neighbor.

IPv6 Conceptual Sending Algorithm Next-Hop Determination

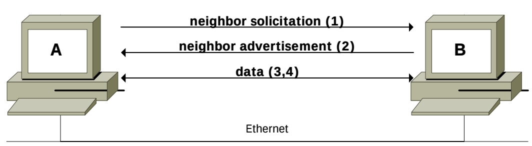
- Next-hop determination at the sender for a given unicast destination operates as follows.



* If the Default Router List is empty, the sender assumes that the destination is on-link.

- For efficiency reasons, next-hop determination is not performed on every packet that is sent. Instead, the results of next-hop determination computations are saved in the Destination Cache. When the sending node has a packet to send, it first examines the Destination Cache. If no entry exists for the destination, next-hop determination is invoked to create a Destination Cache entry.

IPv6 Address Resolution Example



Eth MAC Addr: 0:50:56:8a:0:0
IPv6 link-l Addr: fe80::250:56ff:fe8a:0
IPv6 Sol-Node MA: ff02::1:ff8a:0

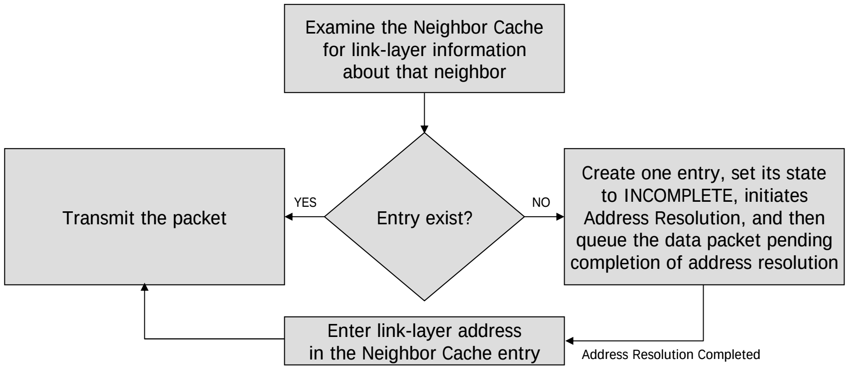
Eth MAC Addr: 0:50:56:d9:88:3f
IPv6 link-l Addr: fe80::250:56ff:fed9:883f
IPv6 Sol-Node MA: ff02::1:ffd9:883f

Src MAC Addr Dst MAC Addr	Src IP Addr Dst IP Addr	ICMP TYPE	Dir
0:50:56:8a:0:0 33:33:ff:d9:88:3f	fe80::250:56ff:fe8a:0 ff02::1:ffd9:883f	neighbor sol: who has fe80::250:56ff:fed9:883f (src lladdr: 0:50:56:8a:0:0)	» 1
0:50:56:d9:88:3f 0:50:56:8a:0:0	fe80::250:56ff:fed9:883f fe80::250:56ff:fe8a:0	neighbor adv: tgt is fe80::250:56ff:fed9:883f (SO) (tgt lladdr: 0:50:56:d9:88:3f)	« 2
0:50:56:8a:0:0 0:50:56:d9:88:3f	fe80::250:56ff:fe8a:0 fe80::250:56ff:fed9:883f	echo request	» 3
0:50:56:d9:88:3f 0:50:56:8a:0:0	fe80::250:56ff:fed9:883f fe80::250:56ff:fe8a:0	echo reply	« 4

IPv6

Conceptual Sending Algorithm Address Resolution

- Once the IP address of the next-hop node is known, the sender follows these steps:



- For multicast-capable interfaces Address Resolution consists of sending a Neighbor Solicitation message and waiting for a Neighbor Advertisement.

IPv6

Conceptual Sending Algorithm

- For multicast packets the next-hop is always the (multicast) destination address and is considered to be on-link.

- Each time a Neighbor Cache entry is accessed while transmitting a unicast packet, the sender checks Neighbor Unreachability Detection related information according to the Neighbor Unreachability Detection algorithm. This unreachability check might result in the sender transmitting a unicast Neighbor Solicitation to verify that the neighbor is still reachable.

IPv6

Address Resolution

- Address resolution is the process through which a node determines the link-layer address of a neighbor given only its IP address.
- Address resolution is performed only on addresses that are determined to be on-link and for which the sender does not know the corresponding link-layer address.
- Address resolution is never performed on multicast addresses.
- When a multicast-capable interface becomes enabled the node must join the all-nodes multicast address on that interface, as well as the solicited-node multicast address corresponding to each of the IP addresses assigned to the interface.

IPv6

Address Resolution
Sending Neighbor Solicitations

- When a node has a unicast packet to send to a neighbor, but does not know the neighbor's link-layer address, it performs address resolution.
- For multicast-capable interfaces this entails creating a Neighbor Cache entry in the INCOMPLETE state and transmitting a Neighbor Solicitation message targeted at the neighbor.
- The solicitation is sent to the solicited-node multicast address corresponding to the target address.
- If the solicitation is being sent to a solicited-node multicast address, the sender must include its link-layer address as a Source Link-Layer Address option.
- Including the source link-layer address in a multicast solicitation is required to give the target an address to which it can send the Neighbor Advertisement.

IPv6

Address Resolution
Sending Neighbor Solicitations

- While waiting for address resolution to complete, the sender must, for each neighbor, retain a small queue of packets waiting for address resolution to complete. Once address resolution completes, the node transmits any queued packets.
- While awaiting a response, the sender should retransmit Neighbor Solicitation messages approximately every RetransTimer milliseconds, even in the absence of additional traffic to the neighbor.
- If no Neighbor Advertisement is received after MAX_MULTICAST_SOLICIT solicitations, address resolution has failed. The sender MUST return ICMP destination unreachable indications with code 3 (Address Unreachable) for each packet queued awaiting address resolution.

IPv6

Address Resolution
Receipt of Neighbor Solicitations

- The recipient should create or update the Neighbor Cache entry for the IP Source Address of the solicitation.
- If an entry does not already exist, the node SHOULD create a new one and set its reachability state to STALE.
- If an entry already exists, and the cached link-layer address differs from the one in the received Source Link-Layer option, the cached address should be replaced by the received address and the entry's reachability state must be set to STALE.
- After any updates to the Neighbor Cache, the node sends a Neighbor Advertisement response.

Address Resolution: Sending Solicited Neighbor Advertisements

IPv6

- A node sends a Neighbor Advertisement in response to a valid Neighbor Solicitation targeting one of the node's assigned addresses.
- The Target Address of the advertisement is copied from the Target Address of the solicitation.
- If the solicitation's IP Destination Address is a multicast address, the Target Link-Layer option must be included in the advertisement.
- If the node is a router, it MUST set the Router flag to one; otherwise it must set the flag to zero.
- If the source of the solicitation is the unspecified address, the node must set the Solicited flag to zero and multicast the advertisement to the all-nodes address. Otherwise, the node must set the Solicited flag to one and unicast the advertisement to the Source Address of the solicitation.

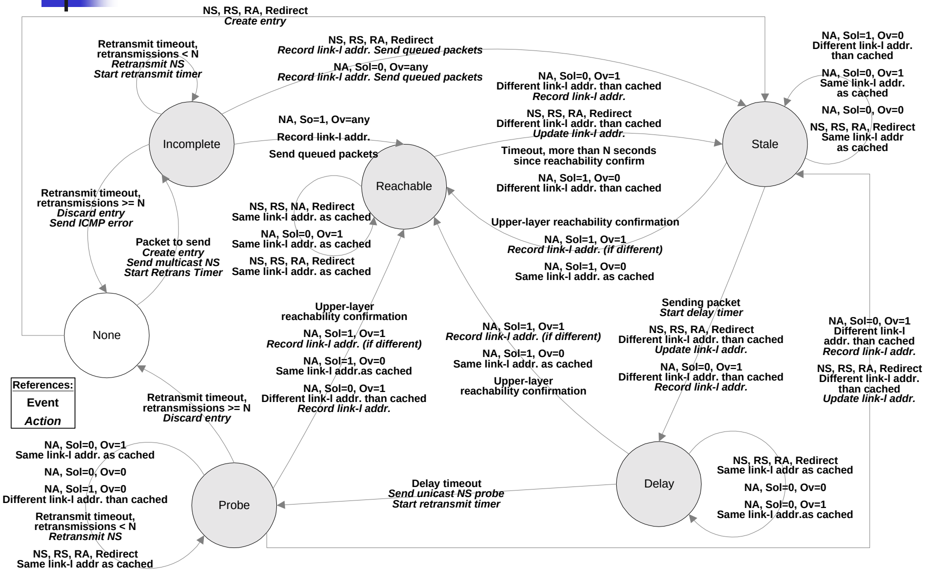
Address Resolution Receipt of Neighbor Advertisements

IPv6

- When a valid Neighbor Advertisement is received, the Neighbor Cache is searched for the target's entry.
- Once the appropriate Neighbor Cache entry has been located, the specific actions taken depend on the state of the Neighbor Cache entry, the flags in the advertisement and the actual link-layer address supplied:
  - State INCOMPLETE:
    - Record the link-layer address in the Neighbor Cache entry.
    - If the advertisement's Solicited flag is set, the state of the entry is set to REACHABLE, otherwise it is set to STALE.
    - It sets the IsRouter flag in the cache entry based on the Router flag in the received advertisement.
    - It sends any packets queued for the neighbor awaiting address resolution.
    - Note that the Override flag is ignored if the entry is in the INCOMPLETE state.
  - If the target's Neighbor Cache entry is in any state other than INCOMPLETE when the advertisement is received, processing becomes quite a bit more complex.

IPv6

State Machine for the Reachability State



IPv6

Address Resolution: Sending Unsolicited Neighbor Advertisements

- In some cases a node may be able to determine that its link-layer address has changed (e.g., hot-swap of an interface card) and may wish to inform its neighbors of the new link-layer address quickly.
- In such cases a node may send unsolicited Neighbor Advertisement messages to the all-nodes multicast address.
- The Target Address field in the unsolicited advertisement is set to an IP address of the interface, and the Target Link-Layer Address option is filled with the new link-layer address. The Solicited flag must be set to zero.
- Neighboring nodes will immediately change the state of their Neighbor Cache entries for the Target Address to STALE, prompting them to verify the path for reachability.
- If the Override flag is set to one, neighboring nodes will install the new link-layer address in their caches. Otherwise, they will ignore the new link-layer address, choosing instead to probe the cached address.

IPv6

Address Resolution: Sending Unsolicited Neighbor Advertisements

- A node that has multiple IP addresses assigned to an interface may multicast a separate Neighbor Advertisement for each address.
- A proxy may multicast Neighbor Advertisements when its link-layer address changes or when it is configured (by system management or other mechanisms) to proxy for an address.
- A node belonging to an anycast address may multicast unsolicited Neighbor Advertisements for the anycast address when the node's link-layer address changes.
- Because unsolicited Neighbor Advertisements do not reliably update caches in all nodes, they should only be viewed as a performance optimization to quickly update the caches in most neighbors.

IPv6

Address Resolution: Anycast Neighbor Advertisements

- From the perspective of Neighbor Discovery, anycast addresses are treated just like unicast addresses in most cases.
- Nodes that have an anycast address assigned to an interface treat them exactly the same as if they were unicast addresses with two exceptions.
  - Neighbor Advertisements sent in response to a Neighbor Solicitation SHOULD be delayed by a random time between 0 and MAX_ANYCAST_DELAY_TIME to reduce the probability of network congestion.
  - Second, the Override flag in Neighbor Advertisements should be set to 0, so that when multiple advertisements are received, the first received advertisement is used rather than the most recently received advertisement.

IPv6

Address Resolution: Proxy Neighbor Advertisements

- A router may proxy for one or more other nodes, that is, through Neighbor Advertisements indicate that it is willing to accept packets not explicitly addressed to itself. For example, a router might accept packets on behalf of a mobile node that has moved off-link.

- All solicited proxy Neighbor Advertisement messages must have the Override flag set to zero. This ensures that if the node itself is present on the link its Neighbor Advertisement will take precedence of any advertisement received from a proxy.

- Finally, when sending a proxy advertisement in response to a Neighbor Solicitation, the sender should delay its response by a random time between 0 and MAX_ANYCAST_DELAY_TIME seconds.

IPv6

Router and Prefix Discovery

- Router Discovery is used to:
  - Locate neighboring routers.
  - Learn prefixes.
  - Learn configuration parameters related to address autoconfiguration.

- Prefix Discovery is the process through which hosts learn the ranges of IP addresses that reside on-link and can be reached directly without going through a router.

- Routers send Router Advertisements that indicate whether the sender is willing to be a default router.

- Router Advertisements also contain Prefix Information options that list the set of prefixes that identify on-link IP addresses.

IPv6

Neighbor Unreachability Detection

- Communication to or through a neighbor may fail for numerous reasons at any time, including hardware failure, hot-swap of an interface card, etc.
- Thus, a node actively tracks the reachability "state" for the neighbors to which it is sending packets.
- NUD is used for all paths between hosts and neighboring nodes, including host-to-host, host-to-router, and router-to-host.
- When a path to a neighbor appears to be failing, the specific recovery procedure depends on how the neighbor is being used.
  - If the neighbor is the ultimate destination, address resolution should be performed again.
  - If the neighbor is a router, attempting to switch to another router would be appropriate.
- Neighbor Unreachability Detection signals the need for next-hop determination by deleting a Neighbor Cache entry.

IPv6

Reachability Confirmation

- A neighbor is considered reachable if the node has recently received a confirmation that packets sent recently to the neighbor were received by its IP layer.
- Positive confirmation can be gathered in two ways:
  - Hints from upper layer protocols that indicate a connection is making “forward progress”.
  - Receipt of a Neighbor Advertisement message that is a response to a Neighbor Solicitation message.
- In TCP, for example, receipt of a (new) acknowledgement indicates that previously sent data reached the peer. It is a confirmation that the next-hop neighbor is reachable.
- For off-link destinations, forward progress implies that the first-hop router is reachable.

Redirect Function

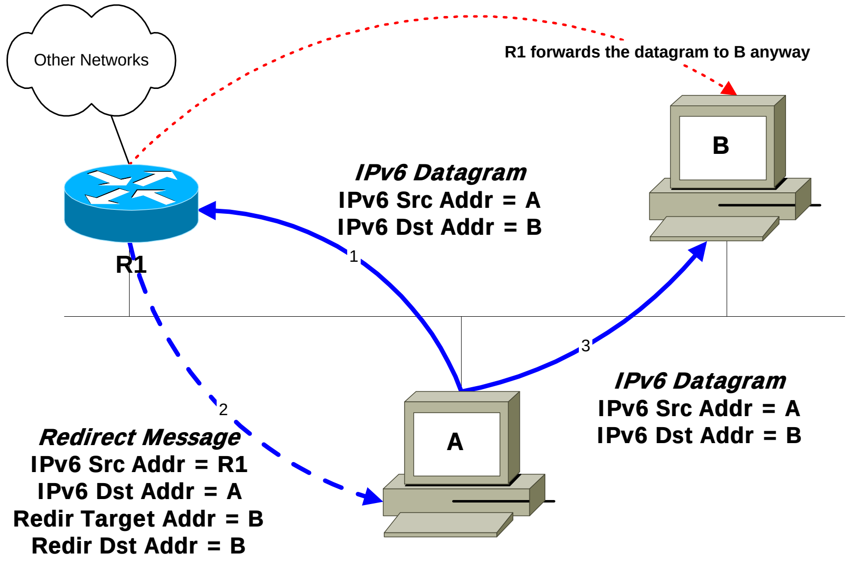
- Redirect messages are sent by routers to redirect a host to a better first-hop router for a specific destination or to inform hosts that a destination is in fact a neighbor (i.e., on-link).
- A router should send a redirect message, subject to rate limiting, whenever it forwards a packet that is not explicitly addressed to itself in which:
  - The Source Address field of the packet identifies a neighbor, and
  - the router determines that a better first-hop node resides on the same link as the sending node for the Destination Address of the packet being forwarded, and
  - the Destination Address of the packet is not a multicast address.

Redirect Function

- The transmitted redirect packet:
  - In the Target Address field: the address to which subsequent packets for the destination should be sent.
  - In the Destination Address field: the destination address of the invoking IP packet.
  - In the options:
    - Target Link-Layer Address option: link-layer address of the target, if known.
    - Redirected Header: as much of the forwarded packet as can fit without the redirect packet exceeding 1280 octets in size.
- A router must limit the rate at which Redirect messages are sent.
- A router must not update its routing tables upon receipt of a Redirect.

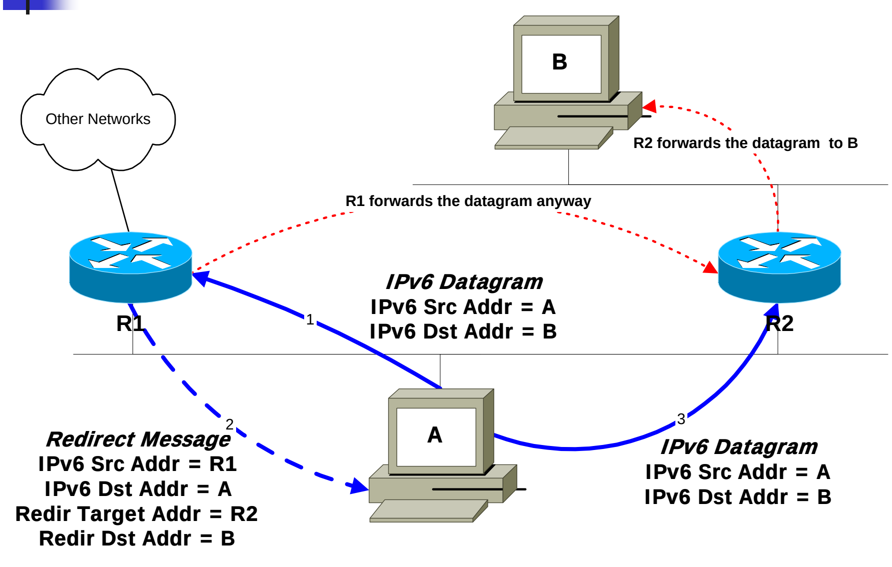
IPv6

Redirect Function
On-link destination



IPv6

Redirect Function
Off-link destination



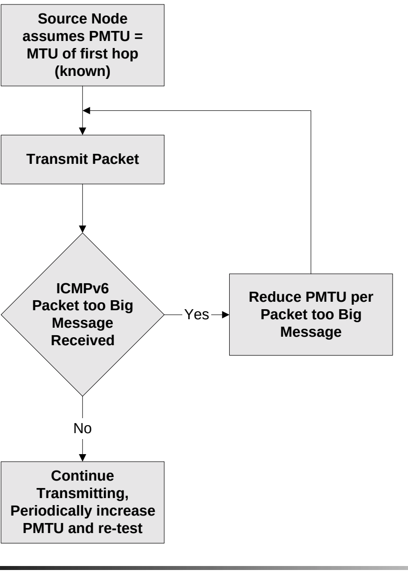
IPv6

Path MTU Discovery

- The Path MTU is the minimum link MTU of all the links in a path between a source node and a destination node.
- Path MTU Discovery is the process by which a node learns the PMTU of a path.
- The basic idea is that a source node initially assumes that the PMTU of a path is the (known) MTU of the first hop in the path.
- If any of the packets sent on that path are too large to be forwarded by some node along the path, that node will discard them and return ICMPv6 Packet Too Big messages.
- Upon receipt of such a message, the source node reduces its assumed PMTU for the path based on the MTU of the constricting hop as reported in the Packet Too Big message.
- The Path MTU Discovery process ends when the node's estimate of the PMTU is less than or equal to the actual PMTU.

IPv6

Path MTU Discovery



- The PMTU of a path may change over time, due to changes in the routing topology.
  - Reductions of the PMTU are detected by Packet Too Big messages
  - To detect increases in a path's PMTU, a node periodically increases its assumed PMTU.
- In the case of a multicast destination, the PMTU is the minimum PMTU value across the set of paths in use.
- The TCP layer must track the PMTU for the path(s) in use by a connection; it should not send segments that would result in packets larger than the PMTU.

IPv6 Stateless Address Autoconfiguration Overview

- IPv6 defines both a stateful and stateless address autoconfiguration mechanism.
- Stateless autoconfiguration requires:
  - no manual configuration of hosts
  - minimal (if any) configuration of routers
  - no additional servers
- The stateless mechanism allows a host to generate its own addresses using a combination of:
  - Locally available information (interface identifier)
  - Information advertised by routers (link prefixes)
- In the absence of routers, a host can only generate link-local addresses. Link-local addresses are sufficient for allowing communication among nodes attached to the same link.

IPv6 Stateless Address Autoconfiguration Overview

- In the stateful autoconfiguration model hosts obtain from a server:
  - Interface addresses
  - Configuration information and parameters.
  - Both.
- Stateful servers maintain a database that keeps track of which addresses have been assigned to which hosts.
- Stateless and stateful autoconfiguration complement each other.
- The stateless approach is used when a site is not particularly concerned with the exact addresses hosts use, so long as they are unique and properly routable.
- The stateful approach is used when a site requires tighter control over exact address assignments.
- Both stateful and stateless address autoconfiguration may be used simultaneously.

IPv6

Stateless Address Autoconfiguration Design Goals

- Stateless autoconfiguration is designed with the following goals in mind:
  - Manual configuration of individual machines before connecting them to the network should not be required.
  - Small sites consisting of a set of machines attached to a single link should not require the presence of a stateful server or router as a prerequisite for communicating.
  - A large site with multiple networks and routers should not require the presence of a stateful address configuration server.
  - Address configuration should facilitate the graceful renumbering of a site's machines.
  - System administrators need the ability to specify whether stateless autoconfiguration, stateful autoconfiguration, or both should be used.

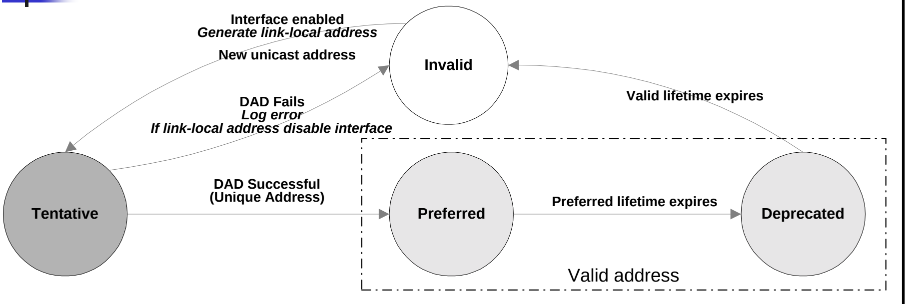
IPv6

Stateless Address Autoconfiguration Address Leasing

- IPv6 addresses are leased to an interface for a fixed (possibly infinite) length of time.
- Each address has an associated lifetime that indicates how long the address is bound to an interface.
- When a lifetime expires, the address may be reassigned to another interface elsewhere.
- To handle the expiration of address bindings gracefully, an address goes through two distinct phases while assigned to an interface:
  - Initially, it is “preferred” (its use in arbitrary communication is unrestricted).
  - Later, an address becomes "deprecated" in anticipation that its current interface binding will become invalid.
- A deprecated address should be used only by applications that have been using it and would have difficulty switching to another address without a service disruption.

IPv6

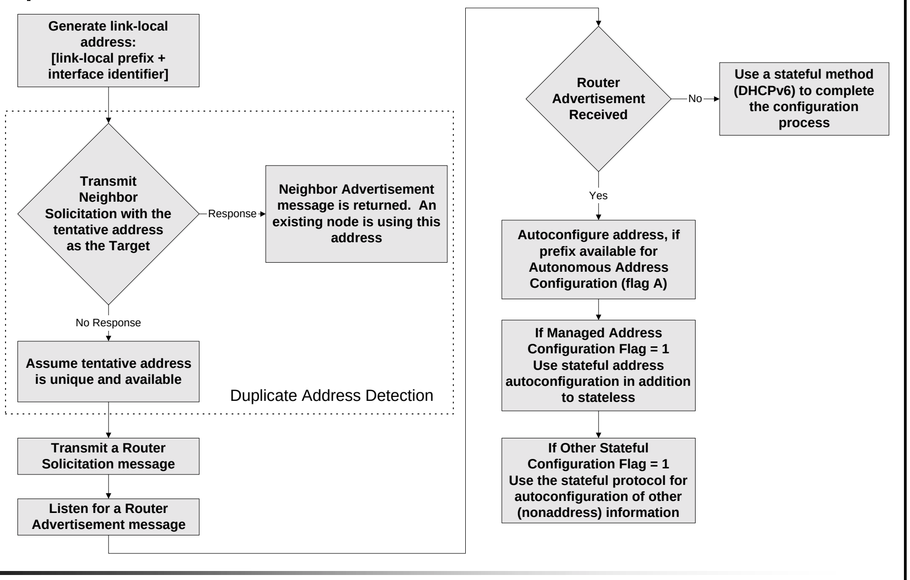
Stateless Address Autoconfiguration Address States



- Invalid address: an Address that is not assigned to any interface.
- Tentative address: An address whose uniqueness on a link is being verified, prior to its assignment to an interface.
- Preferred address: An address assigned to an interface whose use by upper layer protocols is unrestricted.
- Deprecated address: An address assigned to an interface whose use is discouraged, but not forbidden. A deprecated address should no longer be used as a source address in new communications.
- Valid address: A preferred or deprecated address.

IPv6

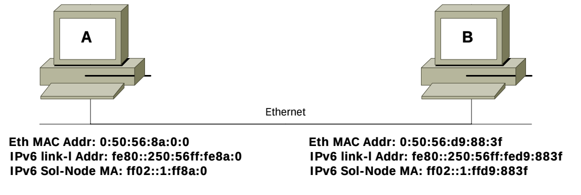
Stateless Address Autoconfiguration



Stateless Address Autoconfiguration
Site Renumbering

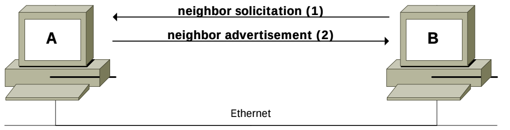
- Address leasing facilitates site renumbering by providing a mechanism to time-out addresses assigned to interfaces in hosts.
- Dividing valid addresses into preferred and deprecated categories provides a way of indicating to upper layers that a valid address may become invalid shortly and that future communication using the address will fail.
- To avoid this scenario, higher layers should use a preferred address to increase the likelihood that an address will remain valid for the duration of the communication.
- The deprecation period should be long enough that most, if not all, communications are using the new address at the time an address becomes invalid.

Duplicate Address Detection
Example: Successful Assignment



Src MAC Addr Dst MAC Addr	Src IP Addr Dst IP Addr	ICMP TYPE	Dir
Add IP address to host A: ip addr add 3ffe:3800:fffb:8001::1/64			
0:50:56:8a:0:0 33:33:ff:0:0:1	:: ff02::1:ff00:1	icmp6: neighbor sol: who has 3ffe:3800:fffb:8001::1 (src lladdr: 0:50:56:8a:0:0)	»
A's address state: inet6 fe80::250:56ff:fe8a:0/10 scope link inet6 3ffe:3800:fffb:8001::1/64 scope global **tentative**			
A few seconds later: inet6 fe80::250:56ff:fe8a:0/10 scope link inet6 3ffe:3800:fffb:8001::1/64 scope global			

Duplicate Address Detection Example: Duplicated Assignment

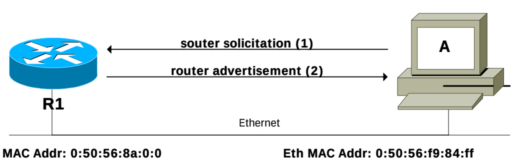


Eth MAC Addr: 0:50:56:8a:0:0
IPv6 link-l Addr: fe80::250:56ff:fe8a:0
IPv6 global Addr: 3ffe:3800:fffb::1
IPv6 Sol-Node MA: ff02::1:ff8a:0

Eth MAC Addr: 0:50:56:d9:88:3f
IPv6 link-l Addr: fe80::250:56ff:fed9:883f
IPv6 Sol-Node MA: ff02::1:ffd9:883f

Src MAC Addr Dst MAC Addr	Src IP Addr Dst IP Addr	ICMP TYPE	Dir
Add IP address to host B: ifconfig lnc0 inet6 3ffe:3800:fffb:8001::1 prefixlen 64			
0:50:56:d9:88:3f 33:33:ff:0:0:1	:: ff02::1:ff00:1	icmp6: neighbor sol: who has 3ffe:3800:fffb:8001::1	«
0:50:56:8a:0:0 33:33:ff:0:0:1	3ffe:3800:fffb:8001::1 ff02::1	icmp6: neighbor adv: tgt is 3ffe:3800:fffb:8001::1 (O) (tgt lladdr: 0:50:5 6:8a:0:0)	»
B's address state: inet6 fe80::250:56ff:fecc:a8ec prefixlen 64 scopeid 0x1 inet6 3ffe:3800:fffb:8001::1 prefixlen 64 **duplicated**			

Stateless Address Autoconfiguration Example



Eth MAC Addr: 0:50:56:8a:0:0
IPv6 link-l Addr: fe80::250:56ff:fe8a:0
IPv6 Sol-Node MA: ff02::1:ff8a:0

Eth MAC Addr: 0:50:56:f9:84:ff
IPv6 link-l Addr: fe80::250:56ff:fef9:84ff
IPv6 Sol-Node MA: ff02::1:fff9:84ff

Src MAC Addr Dst MAC Addr	Src IP Addr Dst IP Addr	ICMP TYPE	Dir
0:50:56:f9:84:ff 33:33:0:0:0:2	fe80::250:56ff:fef9:84ff ff02::2	icmp6: router solicitation (src lladdr: 0:50:56:f9:84:ff)	«
0:50:56:8a:0:0 33:33:0:0:0:1	fe80::250:56ff:fe8a:0 ff02::1	icmp6: router advertisement(chlim=64, router_ltime=1800, reachable_time=0, retrans_time=0)[ndp opt] Prefix: 3ffe:3800:fffb:8001::/64	»
New A Unicast Address: 3ffe:3800:fffb:8001:250:56ff:fef9:84ff			

IPv6 Multicast Listener Discovery

- Enables each IPv6 router to discover the presence of multicast listeners on its attached links.
- Discovers specifically which multicast addresses are of interest to those neighboring nodes.
- This information is then provided to whichever multicast routing protocol is being used.
- There are three types of MLD Messages:
  - Multicast Listener Query (ICMPv6 Type 130)
    - General
    - Multicast-Address-Specific Query
  - Multicast Listener Report (ICMPv6 Type 131)
  - Multicast Listener Done (ICMPv6 Type 132)

IPv6 Multicast Listener Discovery Message Format

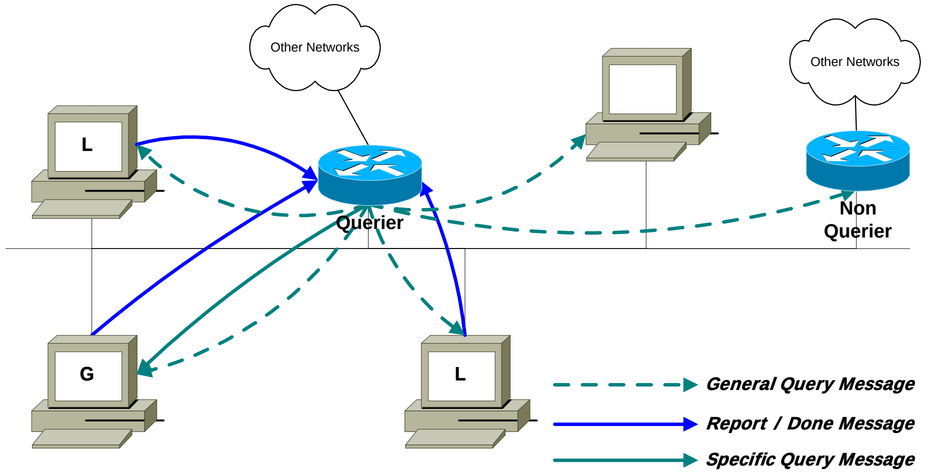
- MLD is a sub-protocol of ICMPv6, messages have the following format:

8	8	16 bits
Type	**Code**	**Checksum**
Maximum Response Delay		**Reserved**
Multicast Address		

IP Fields:
- Source Address: Must be a link-local address assigned to the interface from which this message is sent.
- Hop Limit: 1
- Router Alert in a Hop-by-Hop Options header.

ICMPv6 Fields:
- Maximum Response Delay: Meaningful only in Query messages. Specifies the maximum allowed delay before sending a responding Report (milliseconds).
- Multicast Address: Set to a specific IPv6 multicast address in a Multicast-Address-Specific Query, Report or Done Message.

IPv6 Multicast Listener Discovery



IPv6 Multicast Listener Discovery Node Behavior

- A node may be in one of three possible states with respect to any single IPv6 multicast address on any single interface:
  - Non-Listener: when the node is not listening to the address on the interface.
  - Delaying Listener: when the node is listening to the address on the interface and has a report delay timer running for that address.
  - Idle Listener: when the node is listening to the address on the interface and does not have a report delay timer running for that address.

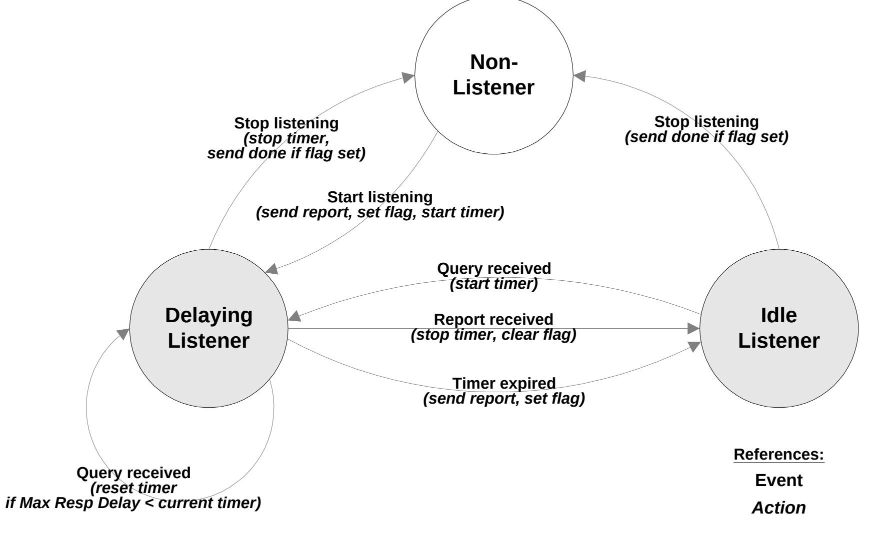
IPv6

Multicast Listener Discovery Node Behavior

- There are seven possible actions that may be taken in response to the events:
  - Send report: for the address on the interface. The Report message is sent to the address being reported.
  - Send done: for the address on the interface. If the flag saying we were the last node to report is cleared, this action may be skipped. The Done message is sent to the link-scope all-routers address (FF02::2).
  - Set flag: that we were the last node to send a report for this address.
  - Clear flag: since we were not the last node to send a report for this address.
  - Start timer: for the address on the interface, using a delay value chosen uniformly from the interval [0, Maximum Response Delay], where Maximum Response Delay is specified in the Query. If this is an unsolicited Report, the timer is set to a delay value chosen uniformly from the interval [0, [Unsolicited Report Interval]].
  - Reset timer: for the address on the interface to a new value, using a delay value chosen uniformly from the interval [0, Maximum Response Delay], as described in "start timer".
  - Stop timer: for the address on the interface.

IPv6

Multicast Listener Discovery Node State Transition Diagram



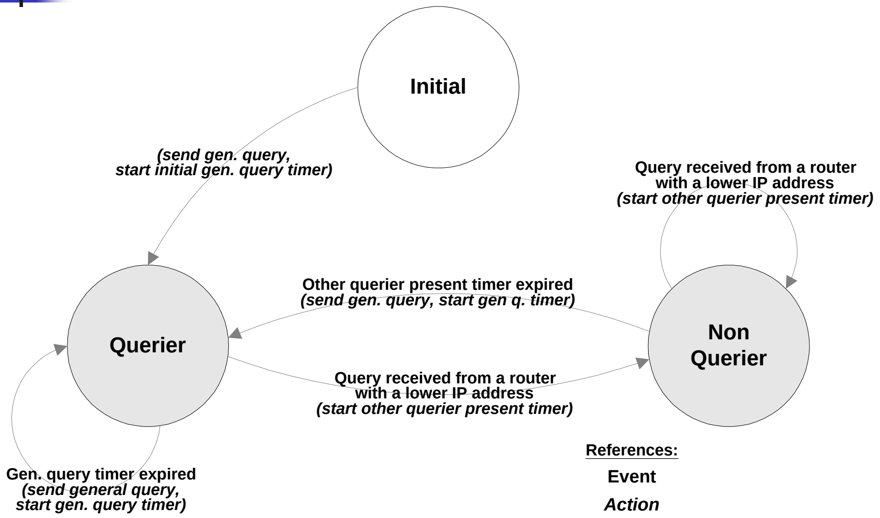
Multicast Listener Discovery Router Behavior

- A router may be in one of two possible states with respect to any single attached link:
  - Querier: when this router is designated to transmit MLD Queries on this link.
  - Non-Querier: when there is another router designated to transmit MLD Queries on this link.
- The following three events can cause the router to change states:
  - Query timer expired: occurs when the timer set for query transmission expires. This event is significant only when in the Querier state.
  - Query received from a router with a lower IP address: occurs when a valid MLD Query is received from a router on the same link with a lower IPv6 Source Address.
  - Other querier present timer expired: occurs when the timer set to note the presence of another querier with a lower IP address on the link expires. This event is significant only when in the Non-Querier state.

Multicast Listener Discovery Router Behavior

- There are three actions that may be taken in response to the events:
  - Start general query timer: for the attached link to [Query Interval].
  - Start other querier present timer: for the attached link to [Other Querier Present Interval].
  - Send general query: on the attached link. The General Query is sent to the link-scope all-nodes address (FF02::1), and has a Maximum Response Delay of [Query Response Interval].

Multicast Listener Discovery Router State Transition Diagram (Link State)



Multicast Listener Discovery Router Behavior

- To keep track of which multicast addresses have listeners, a router may be in one of three possible states with respect to any single IPv6 multicast address on any single attached link:
  - No Listeners Present: when there are no nodes on the link that have sent a Report for this multicast address. This is the initial state for all multicast addresses on the router.
  - Listeners Present: when there is a node on the link that has sent a Report for this multicast address.
  - Checking Listeners: when the router has received a Done message but has not yet heard a Report for the identified address.

IPv6

Multicast Listener Discovery Router Behavior

- There are five significant events that can cause router state transitions:
  - Report received: occurs when the router receives a Report for the address from the link.
  - Done received: occurs when the router receives a Done message for the address from the link.
  - Multicast-address-specific query received: occurs when a router receives a Multicast-Address-Specific Query for the address from the link.
  - Timer expired: occurs when the timer set for a multicast address expires.

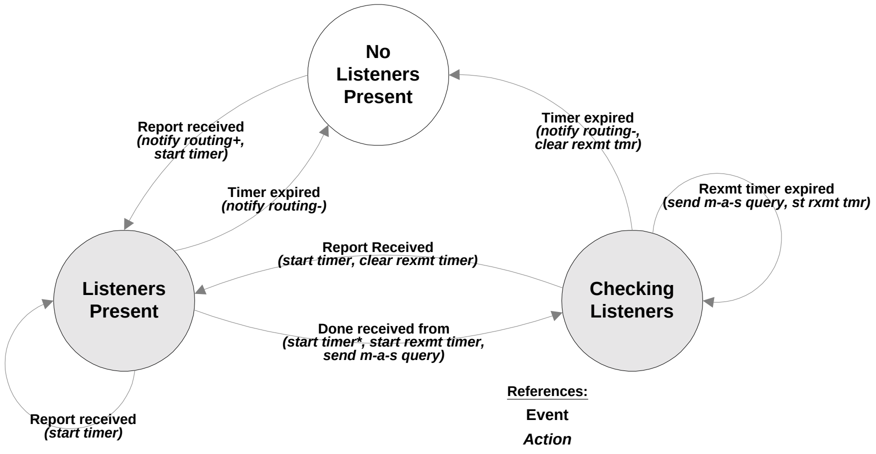
IPv6

Multicast Listener Discovery Router Behavior

- There are seven possible actions that may be taken in response to the events:
  - Start timer for the address on the link.
  - Start timer* for the address on the link - this alternate action sets the timer to the minimum of its current value and either [Last Listener Query Interval] * [Last Listener Query Count] if this router is a Querier, or the Maximum Response Delay in the Query message * [Last Listener Query Count] if this router is a non-Querier.
  - Start retransmit timer for the address on the link.
  - Clear retransmit timer for the address on the link.
  - Send multicast-address-specific query for the address on the link.
  - Notify routing + internally notify the multicast routing protocol that there are listeners to this address on this link.
  - Notify routing - internally notify the multicast routing protocol that there are no longer any listeners to this address on this link.

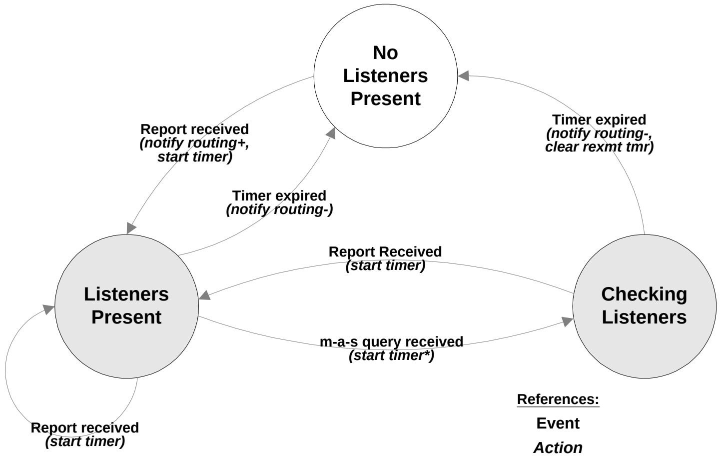
Multicast Listener Discovery Router State Transition Diagram (Group State)

- The following state diagram apply per group per link in Querier state.



Multicast Listener Discovery Router State Transition Diagram (Group State)

- The following state diagram apply per group per link in Non-Querier state.



IPv6

Multicast Listener Discovery Message Destinations

Message Type	IPv6 Destination Address
General Query	Link-scope all-nodes (FF02::1)
Multicast-Address-Specific Query	The multicast address being queried
Report	The multicast address being reported
Done	Link-scope all-routers (FF02::2)

IPv6

IPv6 over PPP

- The Point-to-Point Protocol (PPP) has three main components:
  - A method for encapsulating datagrams over serial links.
  - A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection.
  - A family of Network Control Protocols (NCPs) for establishing and configuring different network-layer protocols.
- The NCP for establishing and configuring the IPv6 over PPP is called the IPv6 Control Protocol (IPv6CP).

IPv6

IPv6 over PPP

- Exactly one IPv6 packet is encapsulated in the Information field of PPP Data Link Layer frames where the Protocol field indicates type hex 0057 (Internet Protocol Version 6).

Flag	Address	Control	Protocol	Information (IPv6 or IPv6CP)	FCS	Flag	
1	1	1	2	n	2	1	Octets

- Protocol:
  - 0057 hex: IPv6
  - 8057 hex: IPv6 Control Protocol

IPv6

IPv6 over PPP Link-local Address

- The interface identifier may be selected using one of the following methods:
  - If an IEEE global identifier is available anywhere on the node, then that address should be used.
  - If an IEEE global identifier is not available, then a different source of uniqueness, such as a machine serial number, should be used.
  - If a good source of uniqueness cannot be found, a random number should be generated.
- The resulting link-local address is:

128 bits

1111111010	0	Interface Identifier from EUI-64 Address, Unique Number or Random Number	
10 bits	54 bits	64 bits	Bits

IPv6 over PPP
IPv6CP Configuration Options

- IPV6CP Configuration Options allow negotiation of desirable IPv6 parameters.
- Current values are assigned as follows:
  - 1: Interface-Identifier.
    This Configuration Option provides a way to negotiate a unique 64-bit interface identifier to be used for the address autoconfiguration at the local end of the link.
  - 2: IPv6-Compression-Protocol.
    This Configuration Option provides a way to negotiate the use of a specific IPv6 packet compression protocol. The IPv6-Compression-Protocol Configuration Option is used to indicate the ability to receive compressed packets. Each end of the link must separately request this option if bi-directional compression is desired. By default, compression is not enabled.

IPv6 DNS Extensions

- To support the storage of IPv6 addresses the following extensions are defined:
  - A new resource record type is defined to map a domain name to an IPv6 address.
  - A new domain is defined to support lookups based on address.
  - Existing queries that perform additional section processing to locate IPv4 addresses are redefined to perform additional section processing on both IPv4 and IPv6 addresses.

IPv6 DNS Extensions
AAAA Record Type

IPv6 **Deprecated**

- The AAAA resource record type is a new record specific to the Internet class that stores a single IPv6 address.
- A 128 bit IPv6 address is encoded in the data portion of an AAAA resource record in network byte order (high-order byte first).
- An AAAA query for a specified domain name in the Internet class returns all associated AAAA resource records in the answer section of a response.
- A type AAAA query does not perform additional section processing.
- The textual representation of the data portion of the AAAA resource record used in a master database file is the textual representation of a IPv6 address.

IPv6 DNS Extensions
IP6.INT Domain

IPv6 **Deprecated**

- A special domain is defined to look up a record given an address.
- The domain is rooted at IP6.INT.
- An IPv6 address is represented as a name in the IP6.INT domain by a sequence of nibbles separated by dots with the suffix ".IP6.INT". The sequence of nibbles is encoded in reverse order, i.e. the low-order nibble is encoded first, followed by the next low-order nibble and so on. Each nibble is represented by a hexadecimal digit.
- Example:
  - The inverse lookup domain name corresponding to the address

    4321:0:1:2:3:4:567:89ab

    would be

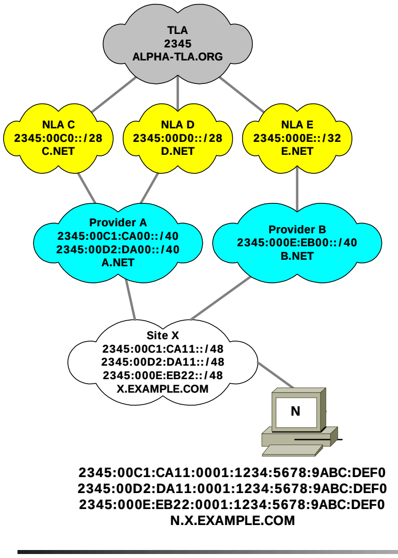
    b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.INT.

IPv6 DNS Extensions
A6 Resource Record

- The A6 RR contains two or three fields:
  - A prefix length.
  - An IPv6 address suffix.
  - The name of the prefix.
- The domain name component shall not be present if the prefix length is zero.
- The address suffix component shall not be present if the prefix length is 128.
- It is suggested that an A6 record intended for use as a prefix for other A6 records have all the insignificant trailing bits in its address suffix field set to zero.
- Example (of textual representation):

```
$ORIGIN example2.net.
subnet5     A6    48 0:0:0:1:: ipv6net2.example2.net.
ipv6net2    A6    0  6666:5555:4::
```


IPv6 DNS Extensions
A6 Record Chains Example



X's DNS:

```
$ORIGIN X.EXAMPLE.COM.
N             A6  64  ::1234:5678:9ABC:DEF0  SUBNET-1.IP6
SUBNET-1.IP6  A6  48  0:0:0:1::  IP6
IP6           A6  48  0::0  SUBSCRIBER-X.IP6.A.NET.
IP6           A6  48  0::0  SUBSCRIBER-X.IP6.B.NET.
```

Elsewhere:

```
SUBSCRIBER-X.IP6.A.NET. A6 40 0:0:0011:: A.NET.IP6.C.NET.
SUBSCRIBER-X.IP6.A.NET. A6 40 0:0:0011:: A.NET.IP6.D.NET.

SUBSCRIBER-X.IP6.B.NET. A6 40 0:0:0022:: B-NET.IP6.E.NET.

A.NET.IP6.C.NET. A6 28 0:0001:CA00:: C.NET.ALPHA-TLA.ORG.

A.NET.IP6.D.NET. A6 28 0:0002:DA00:: D.NET.ALPHA-TLA.ORG.

B-NET.IP6.E.NET. A6 32 0:0:EB00::    E.NET.ALPHA-TLA.ORG.

C.NET.ALPHA-TLA.ORG. A6 0 2345:00C0::
D.NET.ALPHA-TLA.ORG. A6 0 2345:00D0::
E.NET.ALPHA-TLA.ORG. A6 0 2345:000E::
```

IPv6 DNS Extensions
Binary Labels

- A "Bit-String Label" may appear within domain names.
- Represents a sequence of "One-Bit Labels".
- Enables RRs to be stored at any bit-boundary in a binary-named section of the domain name tree.
- Are intended to efficiently solve the problem of storing data and delegating authority on arbitrary boundaries (for reverse zones).
- Textual Representation example:
  - \[b11010000011101]
  - \[o64072/14]
  - \[xd074/14]
  - \[208.116.0.0/14]
  - \[b11101].\[o640]
  - \[x1d].\[o64]
  - \[o35].\[208.0.0.0/8]

IPv6 DNS Extensions
Non-Terminal DNS Name Redirection

- A new RR called "DNAME" provides the capability to map an entire subtree of the DNS name space to another domain.
- It's a solution to the problem of maintaining address-to-name mappings in a context of network renumbering.
- Renumbering Example:

```
$ORIGIN \[x20aa00bbcccc/48].ip6.arpa.
\[xdddd/16]          DNAME          ipv6-rev.example.com.

$ORIGIN \[x266655550004/48].ip6.arpa.
\[x0001/16]          DNAME          ipv6-rev.example.com.

$ORIGIN ipv6-rev.example.com.
\[x1234567812125675/64]        PTR          host.example.com.
```

IPv6

IPv6 DNS Extensions
IP6.ARPA Domain

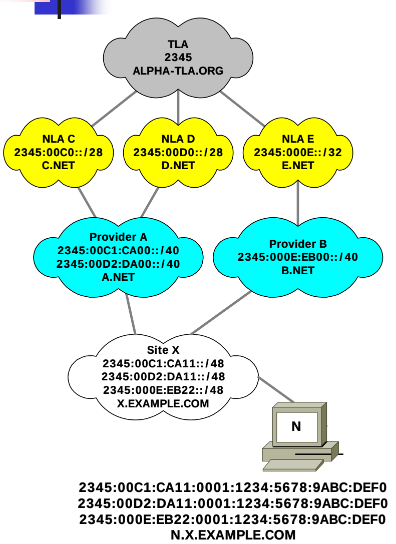
- A new special domain is defined to look up a record given an address.
- The domain is rooted at IP6.ARPA.
- This new scheme for reverse lookups relies on Bynary Labels.
- The inverse lookup domain name corresponding to the address

  4321:0:1:2:3:4:567:89ab
  would be
  \[x4321000000010002000300040567 89ab].IP6.ARPA.

- DNS address space delegation is implemented not by zone cuts and NS records, but by the new DNAME resource record.

IPv6

IPv6 DNS Extensions
IP6.ARPA Domain Example



IP6.ARPA Level:
```
$ORIGIN IP6.ARPA.
\[x234500/24]   DNAME   IP6.ALPHA-TLA.ORG.
\[x267800/24]   DNAME   IP6.BRAVO-TLA.ORG.
\[x29AB00/24]   DNAME   IP6.CHARLIE-TLA.XY.
```

TLA Level (ALPHA-TLA):
```
\[xC/4].IP6.ALPHA-TLA.ORG.   DNAME  IP6.C.NET.
\[xD/4].IP6.ALPHA-TLA.ORG.   DNAME  IP6.D.NET.
\[x0E/8].IP6.ALPHA-TLA.ORG.  DNAME  IP6.E.NET.
```

ISP Level (A, B, C, D, and E):
```
\[x1CA/12].IP6.C.NET.  DNAME  IP6.A.NET.
\[x2DA/12].IP6.D.NET.  DNAME  IP6.A.NET.
\[xEB/8].IP6.E.NET.    DNAME  IP6.B.NET.
\[x11/8].IP6.A.NET.    DNAME  IP6.X.EXAMPLE.COM.
\[x22/8].IP6.B.NET.    DNAME  IP6.X.EXAMPLE.COM.
```

The Site Level (X.EXAMPLE.COM):
```
$ORIGIN IP6.X.EXAMPLE.COM.
\[x0001/16]                  DNAME  SUBNET-1
\[x123456789ABCDEF0].SUBNET-1  PTR    N.X.EXAMPLE.COM.
```

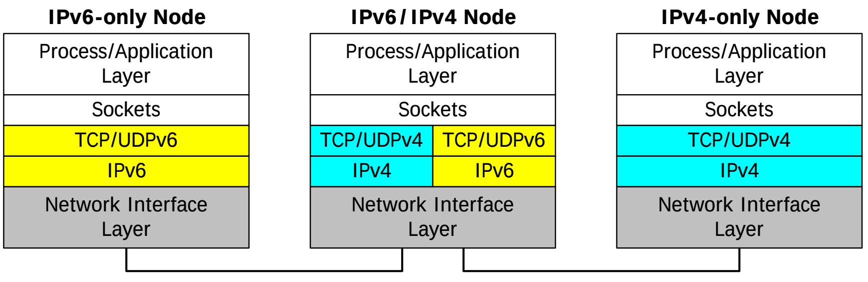
IPv6 DNS Extensions
Modifications to existing Query Types

- All existing query types that perform type A additional section processing, must be redefined to perform type A, A6 and AAAA additional section processing, i.e.:
  - Name server (NS)
  - Mail exchange (MX)
  - Mailbox (MB)

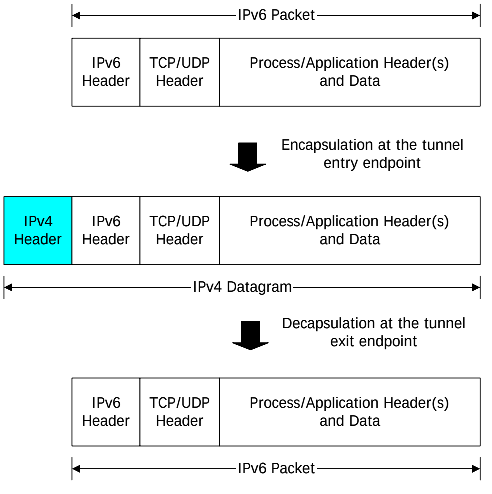
- These new definitions mean that a name server must add any relevant IPv4 addresses and any relevant IPv6 addresses available locally to the additional section of a response when processing any one of the above queries.

Transition Mechanisms
Dual IP Stacks

- Is the simplest mechanism for IPv4 and IPv6 coexistence.
- Node has both IPv4 and IPv6 stacks and addresses.
- DNS Resolver returns IPv6, IPv4 or both to application.
- IPv6 applications can communicate with IPv4 nodes.



Transition Mechanisms
Tunneling IPv6 in IPv4



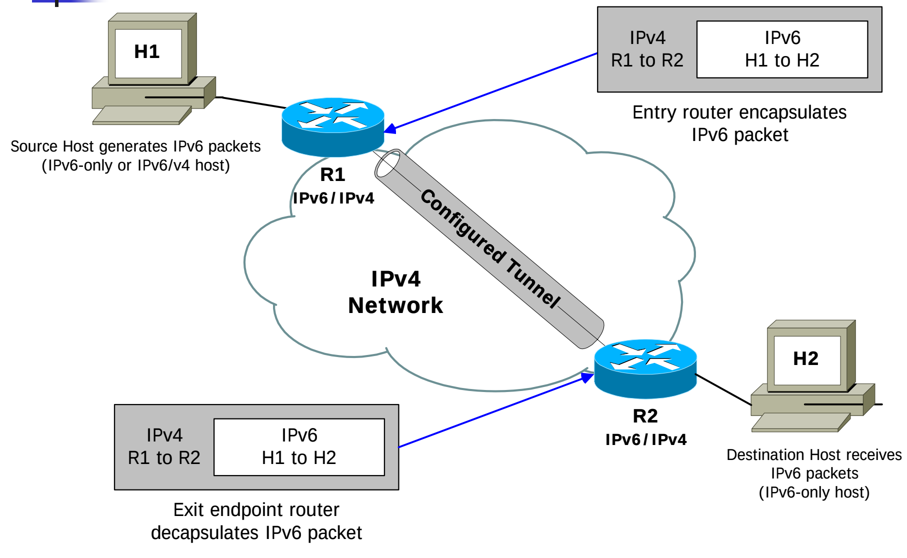
- IPv6 encapsulated in IPv4
- Four possible configurations:
  - Router-to-Router
  - Host-to-Router
  - Host-to-Host
  - Router-to-Host
- The tunnel endpoints takes care of the encapsulation. This process is “transparent” to the other nodes.
- The manner in which endpoints addresses are determined defines:
  - Configured tunnels
  - Automatic tunnels
  - Multicast tunnels

Transition Mechanisms
Configured Tunneling

- Tunnel endpoints are fixed (manually configured).
- Tunnel endpoints must be dual-stack nodes.
  - The IPv4 address is the endpoint for the tunnel.
  - Require reachable IPv4 addresses.
- The tunnels can be either unidirectional or bidirectional.
- Bidirectional configured tunnels behave as virtual point-to-point links.

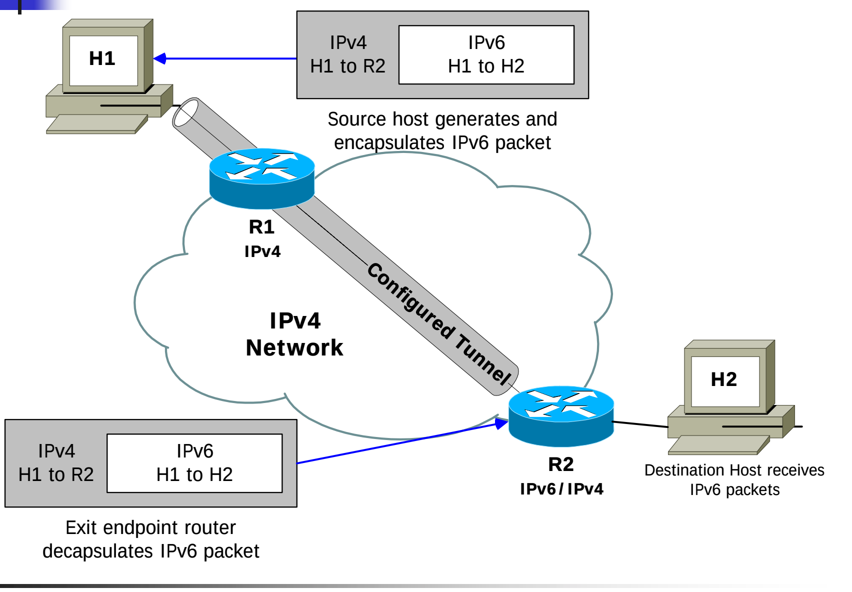
IPv6

Transition Mechanisms
Configured Tunneling: Router-to-Router



IPv6

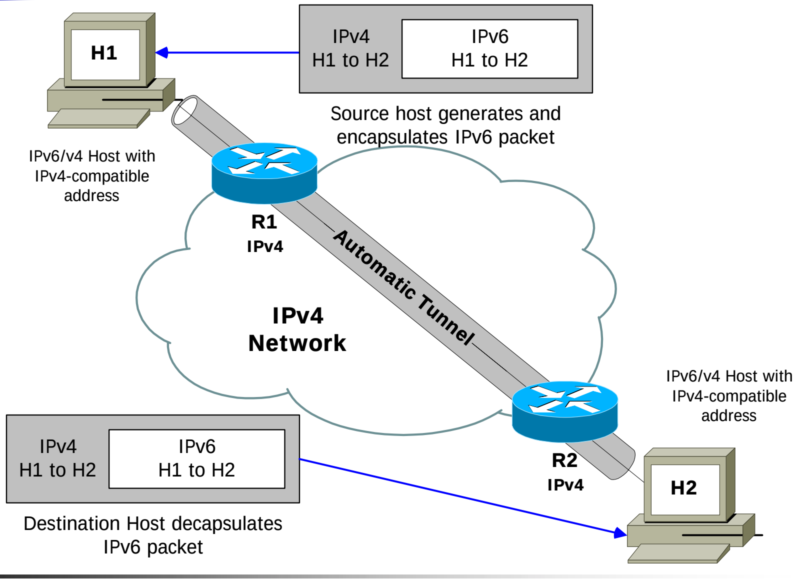
Transition Mechanisms
Configured Tunneling: Host-to-Router



Transition Mechanisms
Automatic Tunneling

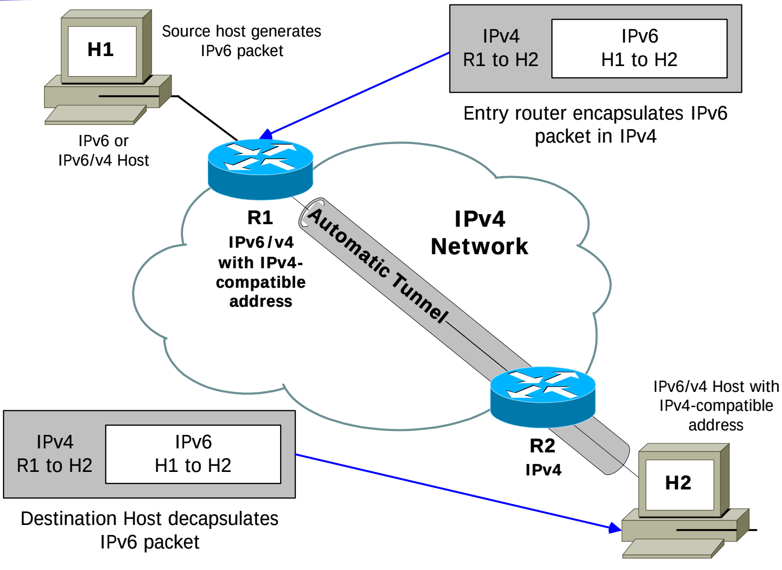
- IPv4 tunnel endpoint address is determined from the IPv4-compatible destination IPv6 address.
  - Example: ::170.210.79.4
- Terminates on a host.
- Routing table redirects ::/96 to automatic tunneling interface.
- If two hosts have IPv4-compatible IPv6 addresses, they can communicate acoss an IPv4 infrastructure using automatic tunneling.
- A dual router, upon receiving an IPv6 packet destined for a host with an IPv4-compatible address, can automatically tunnel that packet to its endpoint.

Transition Mechanisms
Automatic Tunneling: Host-to-Host



IPv6

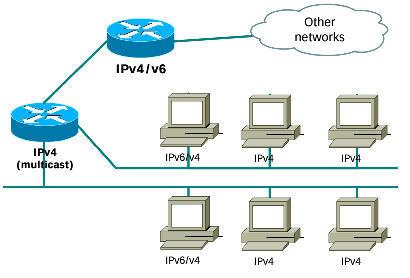
Transition Mechanisms
Automatic Tunneling: Router-to-Host



IPv6

Transition Mechanisms
Multicast Tunneling: 6over4

- Interconnection of isolated IPv6 domains in an IPv4 world.
- No explicit tunnels.
- The egress router must:
  - Have a dual stack
  - Have a globally routable IPv4 address
  - Have an IPv4 multicast infrastructure
  - Implement 6over4 on an external interface
- Uses IPv4 as a link layer for IPv6, that’s why IPv4 multicast is needed.



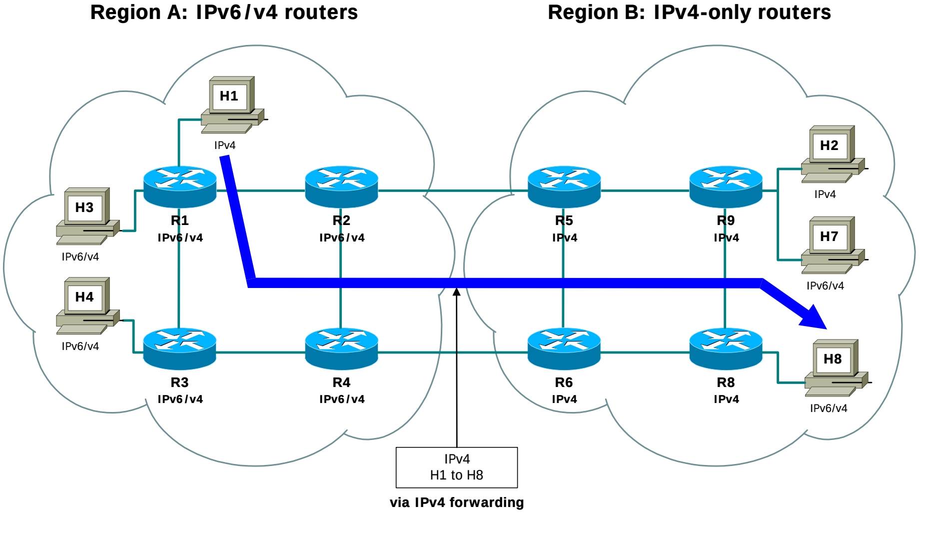
IPv6

Transition Routing

- Terms related to transition routing architecture:
  - Border router: A router that forwards packets across routing domain boundaries.
  - Routing domain: A collection of routers that coordinate routing knowledge using a single protocol.
  - Routing region: Collection of routers, interconnected by a single Internet protocol, that coordinate their routing knowledge using routing protocols from a single IP stack. A routing region may be a superset of a routing domain.
  - Reachability information: Information describing the set of reachable destinations that can be used for packet forwarding decisions.
  - Route leaking: Advertisement of network layer reachability information across routing boundaries.

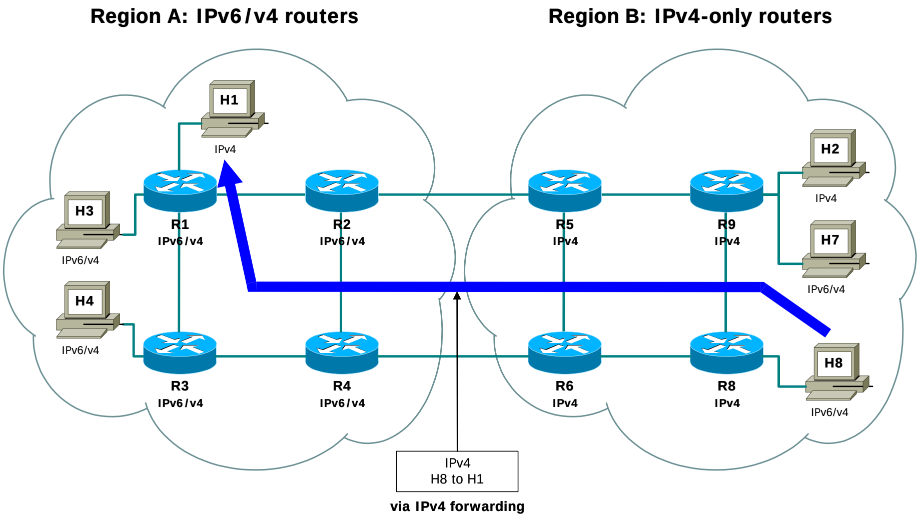
IPv6

Transition Routing
Routing Example (1)



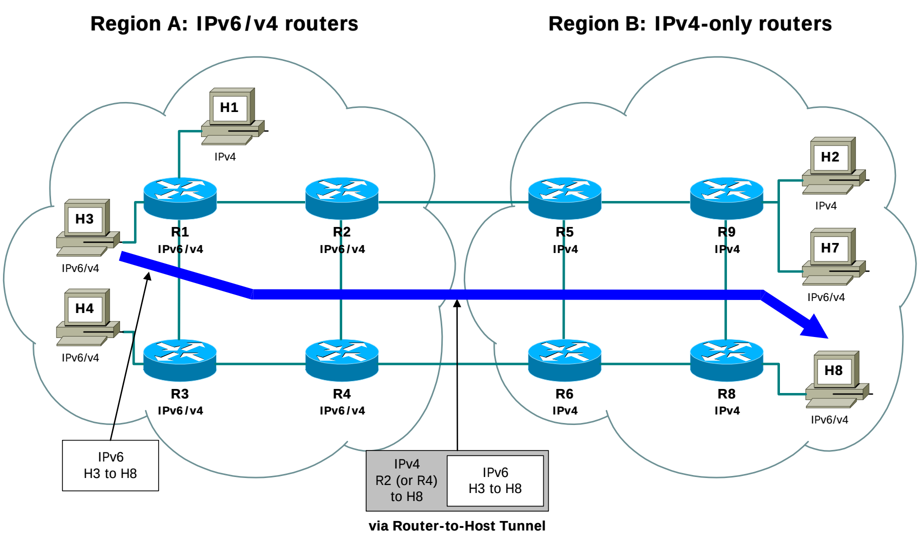
IPv6

Transition Routing
Routing Example (2)



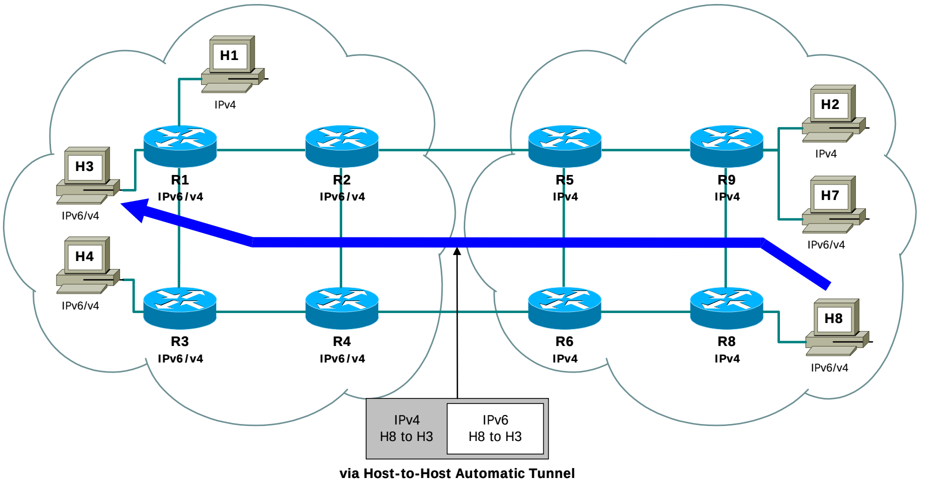
IPv6

Transition Routing
Routing Example (3)



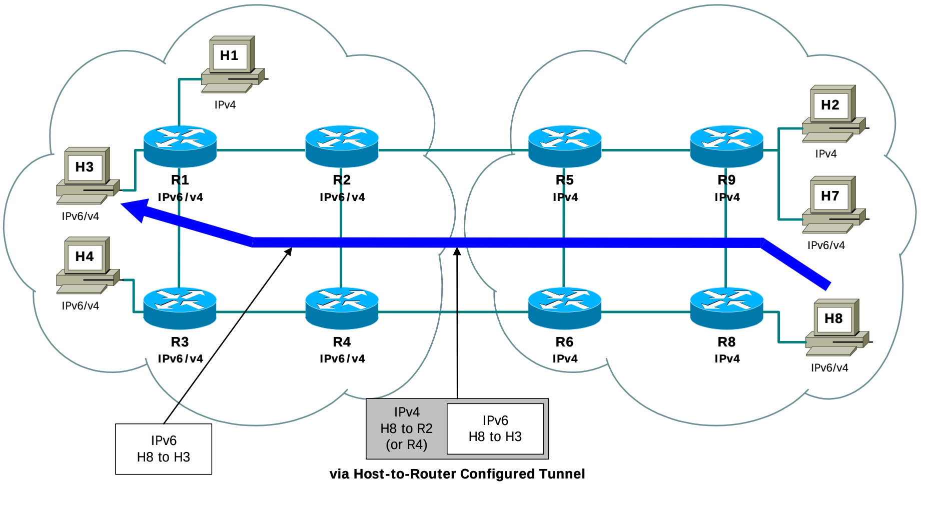
Transition Routing
Routing Example (4)

Region A: IPv6/v4 routers **Region B: IPv4-only routers**



Transition Routing
Routing Example (5)

Region A: IPv6/v4 routers **Region B: IPv4-only routers**



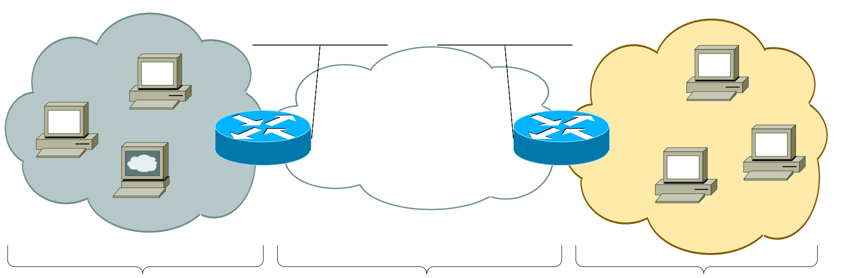
IPv6

6to4

- Mechanism for IPv6 sites to communicate with each other over the IPv4 network without explicit tunnel setup.
- Allows communication with native IPv6 domains.
- Assigns an interim unique IPv6 address prefix to any site that currently has at least one globally unique IPv4 address.
- Not requires:
  - IPv4-compatible IPv6 addresses
  - configured tunnels
- Uses the prefix 2002::/16 to form 6to4 prefixes derived from the IPv4 Address.

IPv6

6to4 – Terminology



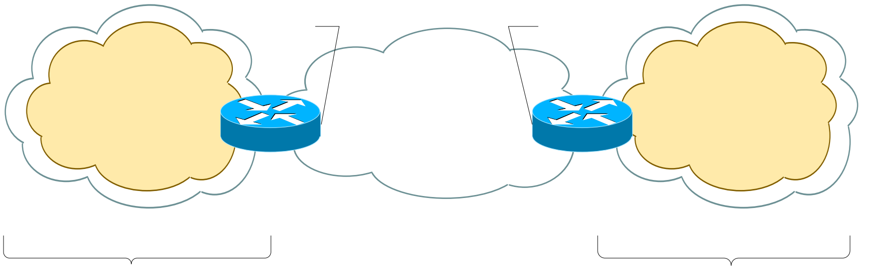
- Requires an IPv4 network communicating both 6to4 routers.
- 6to4 prefix: a prefix derived from an IPv4 address.

  Ex.: 170.210.16.2 → 2002:acd2:1002::/48
- 6to4 address: an IPv6 address constructed using a 6to4 prefix.

IPv6

6to4 – Scenario: All sites work the same

- Requires an IPv4 network communicating both 6to4 routers.
- Each site has an IPv6 prefix in the form 2002:WWXX:YYZZ::/48
- Outgoing packets are encapsulated into IPv4 at the 6to4 router.
- Incoming packets are decapsulated and sent to the internal IPv6 network.
- Any number of 6to4 sites can interoperate with **no tunnel configuration**.



IPv6

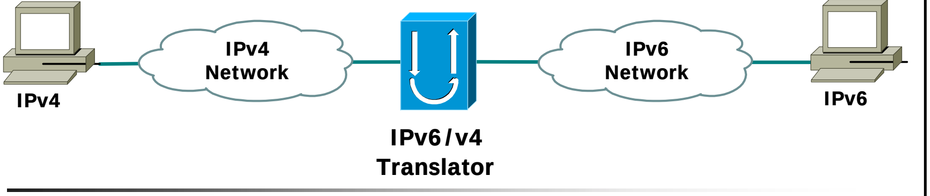
Transition Routing Summary

Host A	Host B	Result
v4-compatible address no local v6 router	v4-compatible address no local v6 router	host to host tunneling in both directions
v4-compatible address no local v6 router	v4-compatible address local v6 router	A->B: host to host tunnel B->A: v6 forwarding plus router->host tunnel
v4-compatible address no local v6 router	incompatible address local v6 router	A->B: host to router tunnel plus v6 forwarding B->A: v6 forwarding plus router to host tunnel
6to4 address	Any IPv6 address	6to4
v4-compatible address local v6 router	v4-compatible address local v6 router	end to end native v6 in both directions
v4-compatible address local v6 router	incompatible address local v6 router	
incompatible address local v6 router	incompatible address local v6 router	

IPv6

SIIT: Stateless IP/ICMP Translation

- Allows IPv6-only hosts to talk to IPv4 hosts.
- Header translator maps corresponding header fields of IPv4⇔IPv6.
- Requires one temporary IPv4 address per host.
- Problem: if no corresponding fields/infos in both headers => no translation possible.
- Conclusion: except segmentation no usage of IPv6 extension headers.
- Requires IPv4-mapped IPv6 address ::FFFF:d.d.d.d



IPv6

NAT-PT: Network Address Translation – Protocol Translation

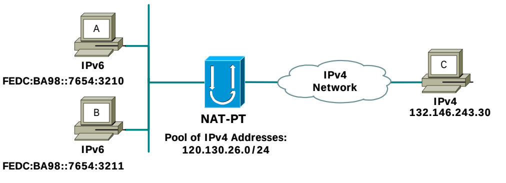
- Enables communication between pure IPv6 and IPv4 nodes.
- Combines two techniques: NAT (Network Address Translation) and SIIT Protocol Translation. Uses Stateful Translation.
- Requires at least one IPv4 address per site.
- Operation:
  - IPv6 node sends packet to NAT-PT server with special destination address..

96 bits	32 bits
NAT-PT Prefix	**IPv4 address**

  - NAT-PT server manages pool of IPv4 addresses, translates headers: IPv4 ⇔ IPv6. Assigns IPv4 address to IPv6 address, forwards packet to IPv4 node.
  - IPv4 node: first IPv4 address of IPv6 node has to be received from DNS. DNS server requests NAT-PT to assign and delivers reserved IPv4 address of IPv6 node.

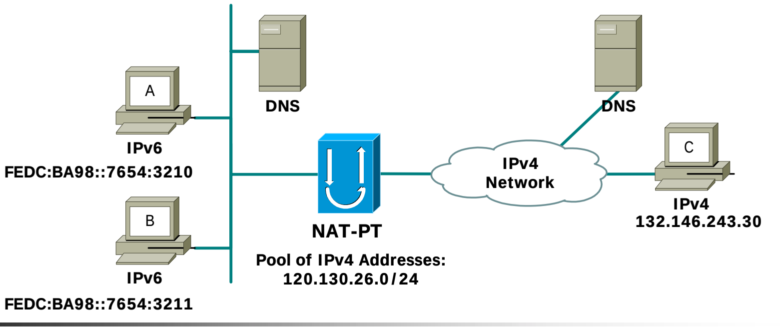
Traditional NAT-PT

- Traditional-NAT-PT would allow hosts within a V6 network to access hosts in the V4 network.
- In a traditional-NAT-PT, sessions are unidirectional, outbound from the V6 network.
- This is in contrast with`Bi-directional-NAT-PT, which permits sessions in both inbound and outbound directions.
- There are two variations to traditional-NAT-PT
  - Basic-NAT-PT: a block of V4 addresses are set aside for translating addresses of V6 hosts as they originate sessions to the V4 hosts in external domain.
  - NAPT-PT, which stands for "Network Address Port Translation + Protocol Translation", would allow V6 nodes to communicate with the V4 nodes transparently using a single V4 address.



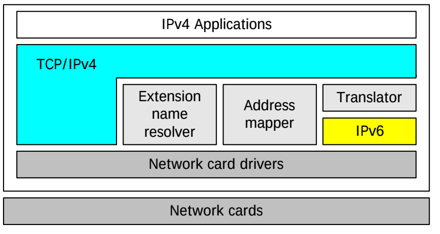
Bi-directional NAT-PT

- Sessions can be initiated from hosts in V4 network as well as the V6 network.
- V6 network addresses are bound to V4 addresses:
  - Statically
  - Dynamically
- Hosts in V4 realm access V6-realm hosts by using DNS for address resolution.
- A DNS-Application-Level-Gateway must be employed to facilitate name to address mapping.



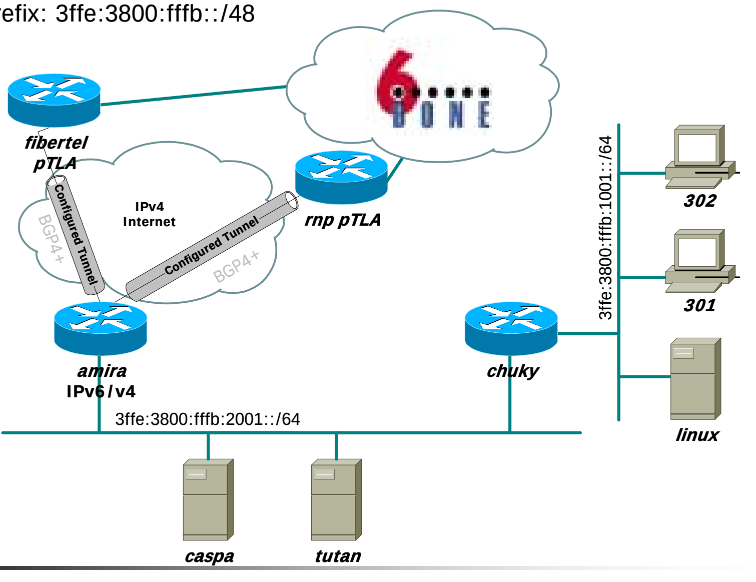
IPv6 Bump in the Stack

- IPv4 applications can transparently communicate via an IPv6 net (if application uses logical names and DNS service).
- Inserts 3 additional modules into IPv4 protocol stack (Dual Stack Host).
- Operation:
  - Translator maps IPv4 packets into IPv6 packets using protocol translation (SIIT).
  - Extension name resolver creates DNS requests (A-rec. + AAAA-rec.) upon appl. DNS request.
  - If DNS server replies A-rec., this is guided directly to IPv4 application.
  - If DNS server replies only AAAA-record, address mapper reserves IPv4 address.
  - Then A-rec. is derived (from reserved IPv4 address and AAAA-rec.) and given to application.
  - Address mapper manages pool of IPv4 addresses and assigned IPv6 addresses.



IPv6 6bone at UTN FRLP

- Prefix: 3ffe:3800:fffb::/48



IPSec – Network Security

- IPSec is designed to provide interoperable, high quality, criptographically-based security for IPv4 and IPv6.
- IPSec provides security at the IP layer, transparent to applications.
- IPSec offers services for:
  - Authentication: Authenticate the sender.
  - Confidentially: Encrypt data before transmission.
  - Data Integrity: Detect altered data in packets.
  - Anti-Replay: Detect replayed packets.
- Open standard, published by the IETF.

IPSec – Protocols

- IPSec uses two protocols to provide traffic security.
  - Authentication Header (AH)
    - Connectionless integrity.
    - Data origin authentication.
    - Anti-replay service (optional).
  - Encapsulating Security Payload (ESP)
    - Connectionless integrity.
    - Data origin authentication.
    - Anti-replay service (optional).
    - Confidentiality (encryption).
    - Limited traffic flow confidentiality
- Two modes of use: transport or tunnel.

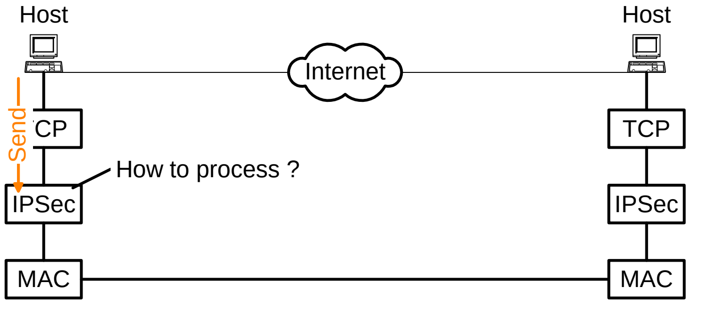
IPSec – Security Associations

- The concept of a "Security Association" is fundamental to IPSec.
- A Security Association (SA) is a simplex "connection" that affords security services to the traffic carried by it.
- A Security Association is unidirectional.
- A Security Association is identified by a triple consisting of:
  - Security Parameter Index (SPI).
  - IP Destination Address.
  - Security protocol identifier (AH or ESP).

IPSec – Security Databases

- There are two nominal databases in this model:
  - Security Policy Database
    - Specifies the policies that determine the disposition of all IP traffic inbound or outbound from an IPSec implementation.
    - An SPD must discriminate among traffic that is afforded IPSec protection and traffic that is allowed to bypass IPSec.
  - Security Association Database
    - Contains parameters that are associated with each security association.
- Selector: a set of IP and upper layer protocol field values that is used by the SPD to map traffic to a policy, i.e., an SA.

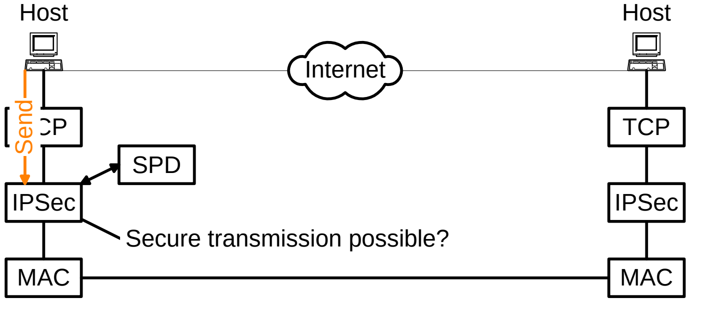
IPSec – Basic Overview



IPSec packet processing:

- Look up in the Security Policy Database (SPD) how to handle the packet:
  - Discard
  - Bypass IPSec -> use IP
  - Apply IPSec

IPSec – Basic Overview



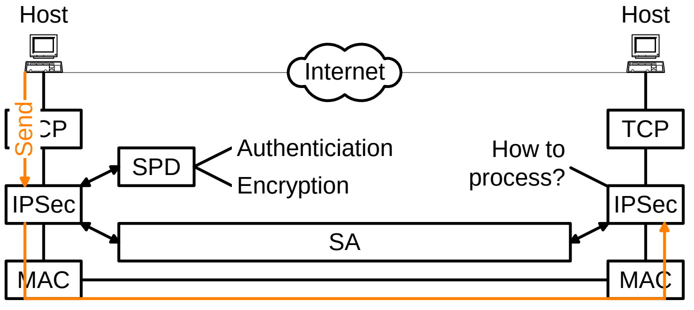
IPSec packet processing:

- Lookup in the Security Association List (SA List) if a Security Association (SA) is available, i.e. if a secure transmission is possible.

- SA stores information about Authentication and / or Encryption algorithm and symmetric, shared keys.

IPv6

IPSec – Basic Overview
Case 1: SA already available.

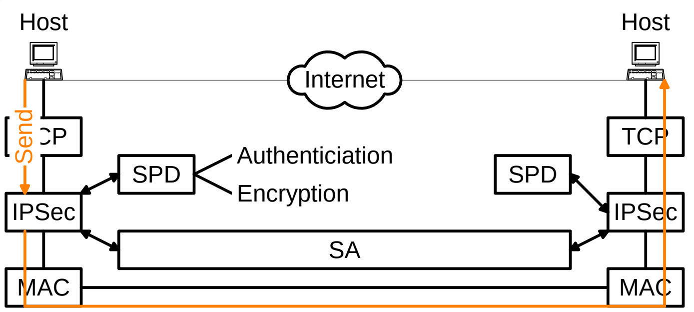


<u>IPSec packet processing:</u>

- Look up in the Security Policy Database (SPD) how to handle the packet:
  - Discard
  - Bypass IPSec -> use IP
  - Apply IPSec

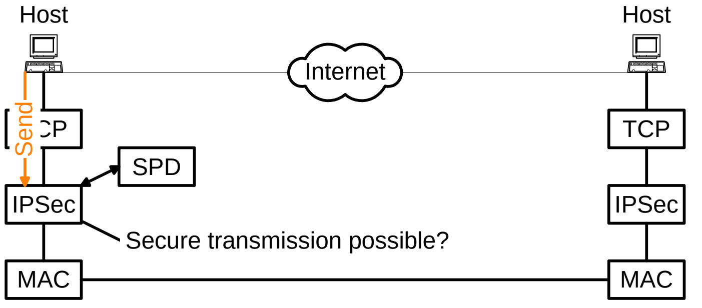
IPv6

IPSec – Basic Overview
Case 1: SA already available.



IPv6

IPSec – Basic Overview
Case 2: SA not available.

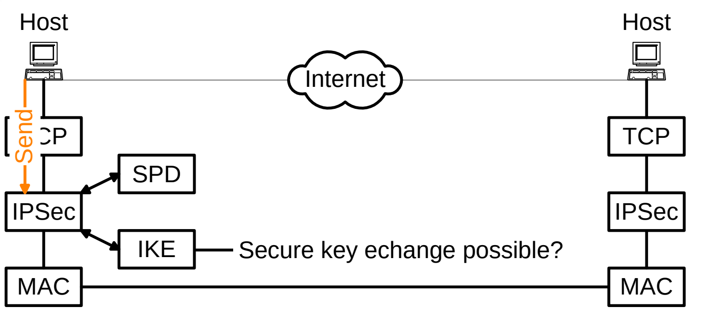


IPSec packet processing:

- Dynamically create a SA using Internet Key Exchange (IKE)
  - Exchange shared keys for IP Sec

IPv6

IPSec – Basic Overview
Case 2: SA not available.

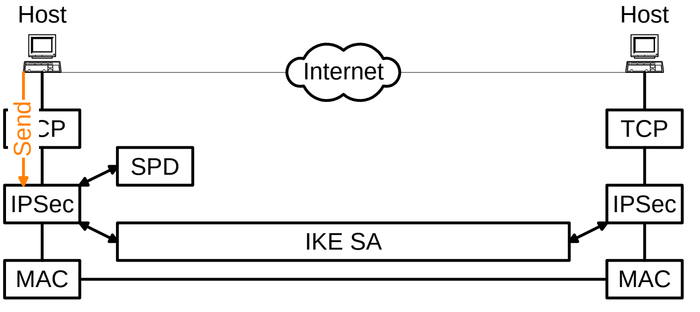


IPSec packet processing:

- Internet Key Exchange (IKE)
  - Create a IKE SA using public keys
  - Exchange shared keys for IP Sec

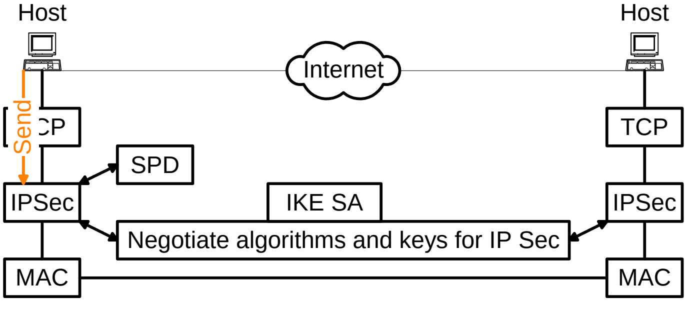
IPv6

IPSec – Basic Overview
Case 2: SA not available.



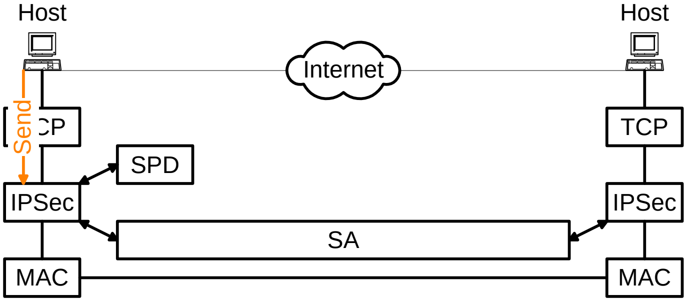
IPv6

IPSec – Basic Overview
Case 2: SA not available.



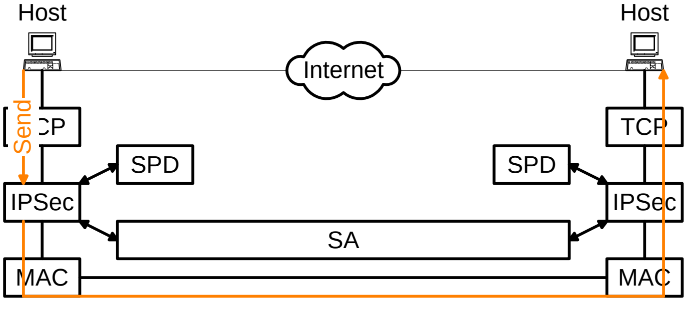
IPv6

IPSec – Basic Overview
Case 2: SA not available.

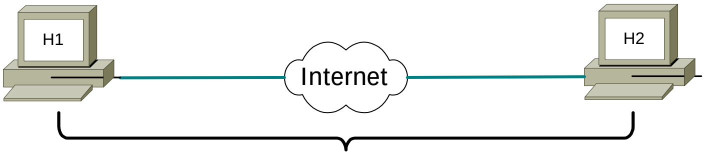


IPv6

IPSec – Basic Overview
Case 2: SA not available.

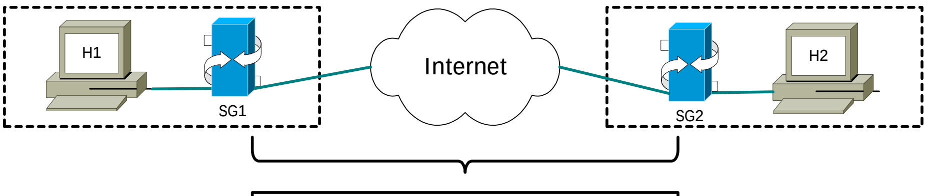


IPSec – Supported Combinations Host to Host



IP_AH_payload (transport)
IP_ESP_payload (transport)
IP_AH_ESP_payload (transport)
IP (host)_AH_IP_payload (tunnel)
IP (host)_ESP_IP_payload (tunnel)

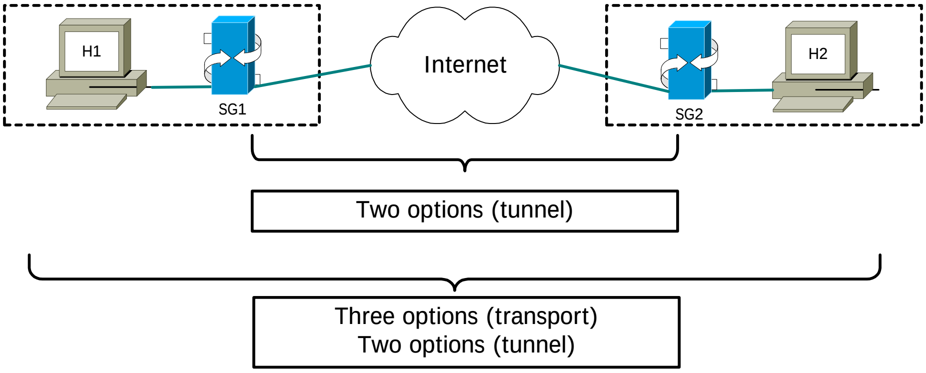
IPSec – Supported Combinations Security Gateway to Security Gateway



IP (SG)_AH_IP_payload (tunnel)
IP (SG)_ESP_IP_payload (tunnel)

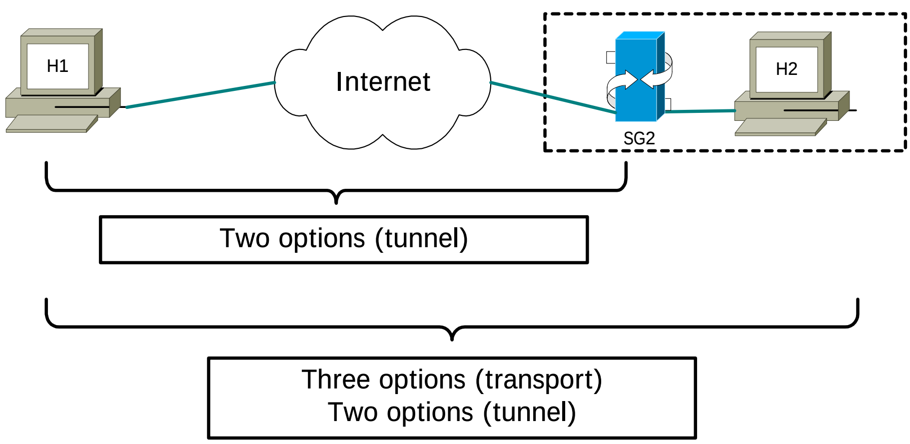
IPv6

IPSec – Supported Combinations
Combination of Cases



IPv6

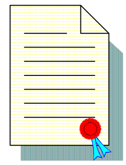
IPSec – Supported Combinations
Remote Access



IPv6

IPSec – Authentication header

- Authentication of data origin
- Data integrity
- Anti-replay (optional)



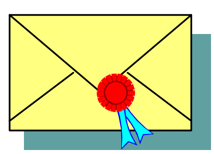
8	8	8	8
Next Header	Payload length	Reserved	
Security Parameters Index (SPI)			
Sequence Number Field			
Authentication Data (variable)			

- SPI = 0 is forbidden, 1..255 is reserved
- Seq. Number only increases (no reset to 0) for anti-replay

IPv6

IPSec – Encapsulation Security Payload

- Data integrity
- Data encryption
- Authentication (optional)
- Anti-replay (optional)



8	8	8	8
Security Parameters Index (SPI)			
Sequence Number Field			
Payload Data (variable)			
Padding (0..255 bytes)		Padding length	Next Header
Authentication Data (variable)			

- SPI = 0 is forbidden, 1..255 is reserved
- Seq. Number only increases (no reset to 0) for anti-replay

IPv6

IPSec – AH Transport Mode

Before applying AH

IP Header	Optional headers	TCP, UDP, ICMP, etc.	Data

After applying AH

Original IP Header	Optional headers(*)	AH	Optional headers(**)	TCP, UDP, ICMP, etc.	Data

Authenticated except for mutable fields

(*): Hop-by-Hop, Dest. Opt, Routing, Fragment.
(**): Dest. Opt

IPv6

IPSec – AH Tunnel Mode

Before applying AH

IP Header	Optional headers	TCP, UDP, ICMP, etc.	Data

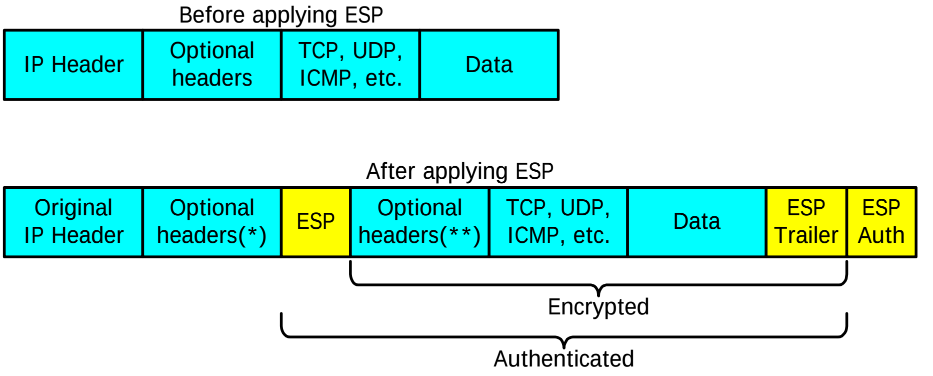
After applying AH

New IP Header	New optional headers	AH	Original IP Header	Optional headers	TCP, UDP, ICMP, etc.	Data

Authenticated except for mutable fields in new IP hdr

IPv6

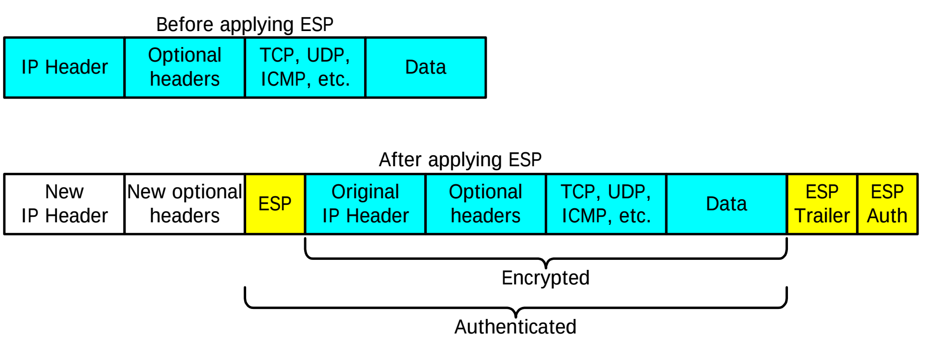
IPSec – ESP Transport Mode



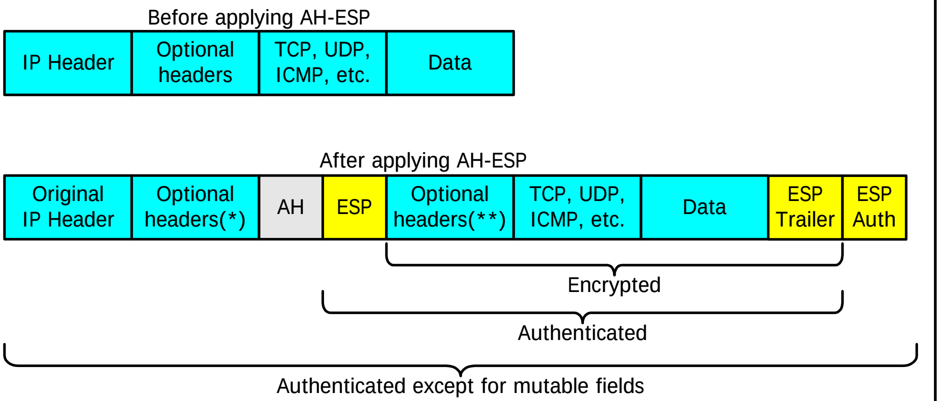
(*): Hop-by-Hop, Dest. Opt, Routing, Fragment.
(**): Dest. Opt

IPv6

IPSec – ESP Tunnel Mode

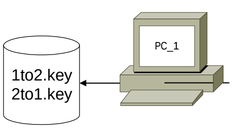


IPSec – AH-ESP Transport Mode



(*): Hop-by-Hop, Dest. Opt, Routing, Fragment.
(**): Dest. Opt

IPSec – Example



SPD

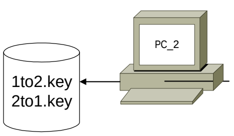
Policy	Dst Addr	Src Addr	Layer 4 Protocol	Dst Port	Src Port	IP Sec Protocol	IP Sec Mode	Direction	Action
1	PC_2	PC_1	*	*	*	AH	Transport	Bidirect.	Apply

SA

SA Entry	SPI	Dst Addr	Src Addr	Layer 4 Protocol	Src Port	Dst Port	Auth. Algorithm	Key(file)	Direction
2	**101**	PC_2	policy	policy	policy	policy	HMAC-MD5	1to2.key	**Out**
1	100	PC_1	policy	policy	policy	policy	HMAC-MD5	2to1.key	In

SPD

Policy	Dst Addr	Src Addr	Layer 4 Protocol	Dst Port	Src Port	IP Sec Protocol	IP Sec Mode	Direction	Action
1	PC_2	PC_1	*	*	*	AH	Transport	Bidirect.	Apply



SA

SA Entry	SPI	Dst Addr	Src Addr	Layer 4 Protocol	Src Port	Dst Port	Auth. Algorithm	Key(file)	Direction
2	**101**	PC_2	policy	policy	policy	policy	HMAC-MD5	1to2.key	**In**
1	100	PC_1	policy	policy	policy	policy	HMAC-MD5	2to1.key	Out

IPv6

Socket Interface Extensions for IPv6
Motivation

- While IPv4 addresses are 32 bits long, IPv6 interfaces are identified by 128-bit addresses.
- The socket interface makes the size of an IP address quite visible to an application.
- Those parts of the API that expose the addresses must be changed.
- IPv6 also introduces new features which must be made visible to applications via the API, e.g.:
  - Traffic class
  - Flow Label

IPv6

Socket Interface Extensions for IPv6
Design Considerations

- The API changes should:
  - Provide both source and binary compatibility for programs written to the original API.
  - Be as small as possible in order to simplify the task of converting existing IPv4 applications to IPv6.
  - Be able to use this API to interoperate with both IPv6 and IPv4 hosts. Applications should not need to know which type of host they are communicating with.

Socket Interface Extensions for IPv6
What Needs to be Changed

- Core socket functions
  - These functions need not change for IPv6.
- Address data structures
  - A new IPv6-specific address data structure is needed.
- Name-to-address translation functions
  - New functions are defined to support IPv4 and IPv6.
  - The POSIX 1003.g draft specifies a new nodename-to-address translation function which is protocol independent.
- Address conversion functions
  - New functions that convert both IPv4 and IPv6 addresses.
- Miscellaneous features
  - New interfaces to support the IPv6 traffic class, flow label, and hop limit header fields.
  - New socket options are needed tocontrol the sending and receiving of IPv6 multicast packets.

Socket Interface Extensions for IPv6
IPv6 Address Family and Protocol Family

- New address family name: AF_INET6
  - Defined in <sys/socket.h>
  - New sockaddr_in6 data structure.
- New protocol family name: PF_INET6
  - Defined in <sys/socket.h>
  - Used in the first argument to the socket() function.

IPv6

Socket Interface Extensions for IPv6
IPv6 Address Structure

- A new in6_addr structure holds a single IPv6 address:

IPv6
```
struct in6_addr {
    uint8_t  s6_addr[16];   /* IPv6 address */
};
```

IPv4
```
struct in_addr {
    u_long s_addr;
} ;
```


IPv6

Socket Interface Extensions for IPv6
Socket Address Structure

- New sockaddr_in6 structure holds IPv6 addresses (<netinet/in.h>)

IPv6
```
struct sockaddr_in6 {
    sa_family_t   sin6_family;      /* AF_INET6 */
    in_port_t     sin6_port;        /* transport layer port # */
    uint32_t      sin6_flowinfo;    /* IPv6 traffic class & flow info */
    struct in6_addr sin6_addr;      /* IPv6 address */
    uint32_t      sin6_scope_id;    /* set of intf. for a scope */
};
```

- *sin6_flowinfo* contains the traffic class and the flow label.
- *sin6_scope_id* identifies a set of interfaces as appropriate for the scope of the address carried in the *sin6_addr* field.
  - Link scope: interface index.
  - Site scope: site identifier.
  - Not completely specified

IPv4
```
struct sockaddr_in {
    short   sin_family;
    u_short sin_port;
    struct  in_addr sin_addr;
    char    sin_zero[8];
};
```

Socket Interface Extensions for IPv6
The Socket Functions

IPv6

- Applications call the socket() function to create a socket descriptor that represents a communication endpoint.

IPv6

*s = socket(**PF_INET6**, SOCK_STREAM, 0);* */* TCP Socket */*

*s = socket(**PF_INET6**, SOCK_DGRAM, 0);* */* UDP Socket */*

- Once the application has created a PF_INET6 socket, it must use the *sockaddr_in6* address structure when passing addresses in to the system.
  - *bind()*
  - *connect()*
  - *sendmsg()*
  - *sendto()*

IPv4

s = socket(PF_INET, SOCK_STREAM, 0);
s = socket(PF_INET, SOCK_DGRAM, 0);

Socket Interface Extensions for IPv6
The Socket Functions

IPv6

- The system will use the sockaddr_in6 address structure to return addresses to applications that are using PF_INET6 sockets.
- The functions that return an address from the system to an application are:
  - *accept()*
  - *recvfrom()*
  - *recvmsg()*
  - *getpeername()*
  - *getsockname()*
- No changes to the syntax of the socket functions are needed to support IPv6.

Socket Interface Extensions for IPv6
Compatibility with IPv4 Nodes

- IPv6 applications are able to interoperate with IPv4 applications.
  - Uses the IPv4-mapped IPv6 address format.
  - IPv4-mapped addresses are written as follows:

    ::FFFF:<IPv4-address>

- Applications may use PF_INET6 sockets to:
  - open TCP connections to IPv4 nodes
  - send UDP packets to IPv4 nodes

  Encoding the destination's IPv4 address as an IPv4-mapped IPv6 address.

- When applications use PF_INET6 sockets to:
  - accept TCP connections from IPv4 nodes
  - receive UDP packets from IPv4 nodes

  The system returns the peer's address using a sockaddr_in6 structure encoded this way.

Socket Interface Extensions for IPv6
IPv6 Wildcard Address

- While the bind() function allows applications to select the source IP address of UDP packets and TCP connections, applications often want the system to select the source address for them.
  - With IPv4, one specifies the address as the symbolic constant INADDR_ANY.
  - In IPv6 a symbolic constant can be used to initialize an IPv6 address variable, but cannot be used in an assignment.
  - Systems provide the wildcard in two forms:
    - *extern const struct in6_addr* ***in6addr_any;***
    - *struct in6_addr anyaddr =* ***IN6ADDR_ANY_INIT;***

      (can be used ONLY at declaration time)
  - Applications use in6addr_any similarly to the way they use INADDR_ANY in IPv4.

Socket Interface Extensions for IPv6
IPv6 Loopback Address

- Applications may need to send UDP packets to, or originate TCP connections to, services residing on the local node.
  - In IPv4, they can do this by using the constant IPv4 address INADDR_LOOPBACK
  - The IPv6 loopback address is provided in two forms:
    - *extern const struct in6_addr* ***in6addr_loopback;***
    - *struct in6_addr loopbackaddr =* ***IN6ADDR_LOOPBACK_INIT****;*
      (can be used ONLY at declaration time)

Socket Interface Extensions for IPv6
Unicast Hop Limit

- A new setsockopt() option controls the hop limit used in outgoing unicast IPv6 packets.
- The name of this option is **IPV6_UNICAST_HOPS**, and it is used at the **IPPROTO_IPV6** layer.
- Example:

```
int  hoplimit = 10;
   if (setsockopt(s, IPPROTO_IPV6, IPV6_UNICAST_HOPS,
               (char *) &hoplimit, sizeof(hoplimit)) == -1)
      perror("setsockopt IPV6_UNICAST_HOPS");
```

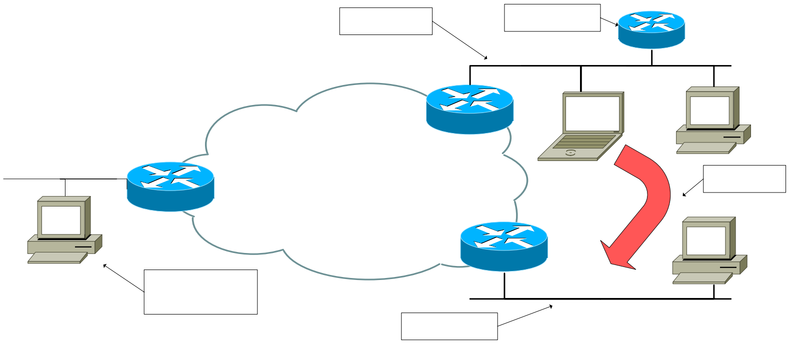
IPv6

Socket Interface Extensions for IPv6 Sending and Receiving Multicast Packets

- IPv6 applications may send UDP multicast packets by simply specifying an IPv6 multicast address in the address argument of the sendto() function.
- Three socket options at the IPPROTO_IPV6 layer control some of the`parameters for sending multicast packets.
  - **IPV6_MULTICAST_IF**: Set the interface to use for outgoing multicast packets. The argument is the index of the interface to use.
  - **IPV6_MULTICAST_HOPS**: Set the hop limit to use for outgoing multicast packets.
  - **IPV6_MULTICAST_LOOP**: If a multicast datagram is sent to a group to which the sending host itself belongs: 1: loop back a copy, 0: don't loop back a copy.
  - **IPV6_JOIN_GROUP**: Join a multicast group on a specified local interface.
  - **IPV6_LEAVE_GROUP**: Leave a multicast group on a specified interface.

IPv6

Mobility Terms



- Home address: permanent address of the mobile node.
- Home subnet prefix: prefix corresponding to the home address.
- Foreign subnet prefix: prefix of the foreign link.
- Care-of address: address assigned to the mobile node on the foreign link.
- Binding: the association of the home address of a mobile node with a care-off address.

Mobility Terms ---

- zzzz