



IPv6 Tutorial



RIPE 40 Meeting
Prague, Czech Republic

Florent Parent, Florent.Parent@viagenie.qc.ca

Viagénie

<http://www.viagenie.qc.ca>

2001-10-01



- Features
- Header
- Addressing
- ICMPv6, Neighbor Discovery, Autoconfiguration
- DNS
- Routing protocols
- Transition mechanisms
- Internet deployment
- Host and Router configurations

IPv6 Features

- Larger address space
- Efficient IP header and datagram
- Mandatory features

- From 32 bits to 128 bits addresses enables:
 - Global reachability:
 - No hidden networks, hosts
 - All hosts can be reachable and be "servers"
 - End-to-end security can be used
 - Flexibility
 - Multiple levels of hierarchy in the address space
 - Autoconfiguration
 - Use of 64 bits for link-layer address encapsulation with warranty of uniqueness

- "Plug and play"
 - By autoconfiguration
- Aggregation
 - Multiple prefixes for the same site enables multihoming without cutting holes in the aggregation
- Multihoming
- Renumbering
 - By using autoconfiguration and multiple prefixes, renumbering becomes doable

Efficient and Extensible IP Datagram



- Less number of fields enables:
 - Routing efficiency
 - Performance
 - Forwarding rate scalability
- Extensibility of header
 - Better handling of options
 - No checksum
- 64 bits aligned
- Flow label

- Security
- Mobility
 - More optimized in IPv6 than MobileIPv4
- Multicast use:
 - No broadcast
 - Efficient use of the network and less interrupts on NICs
 - Scoped groups
- Transition richness
 - Seamless transition
 - Software change
 - Mechanisms and tools for IPv4-IPv6 interaction

Summary



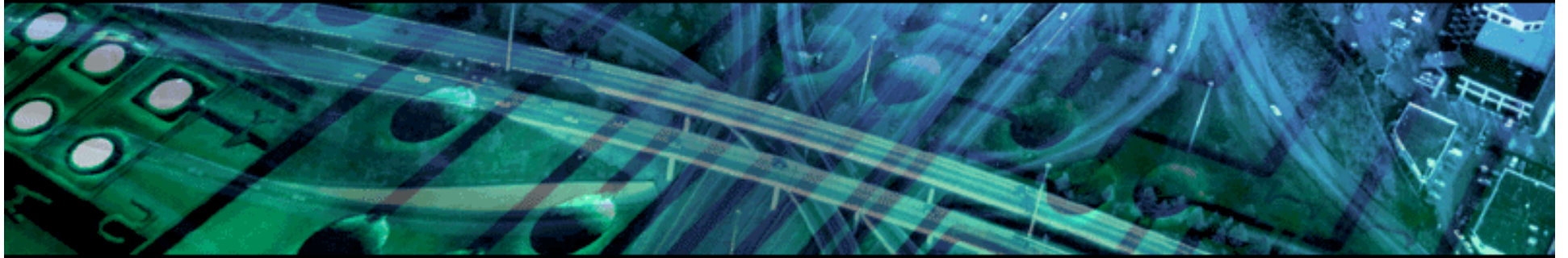
- Larger address space enables many new features
- More clean and efficient header
- Mandatory features

References



- RFC2460, *Internet Protocol, Version 6 (IPv6) Specification*, S. Deering, R. Hinden, IETF, 1998-12-01, <http://www.normos.org/ietf/rfc/rfc2460.txt>

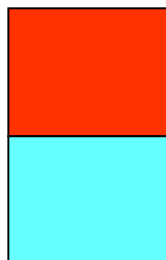
IPv6 Header



- IPv4 Header
- IPv6 Header
- Header Fields
- Extension Headers
- Routing Header

- IPv4 header = 20 bytes without options

Ver.	header	TOS	total length	
identification		flag	fragment offset	
TTL	Protocol	Checksum		
32 bits Source Address				
32 bits Destination Address				



removed

changed

- IPv6 header = 40 bytes without extentions

Ver.	TrafficClass	Flow Label	
Payload Length		Next Header	Hop Limit
128 bits Source Address			
128 bits Destination Address			

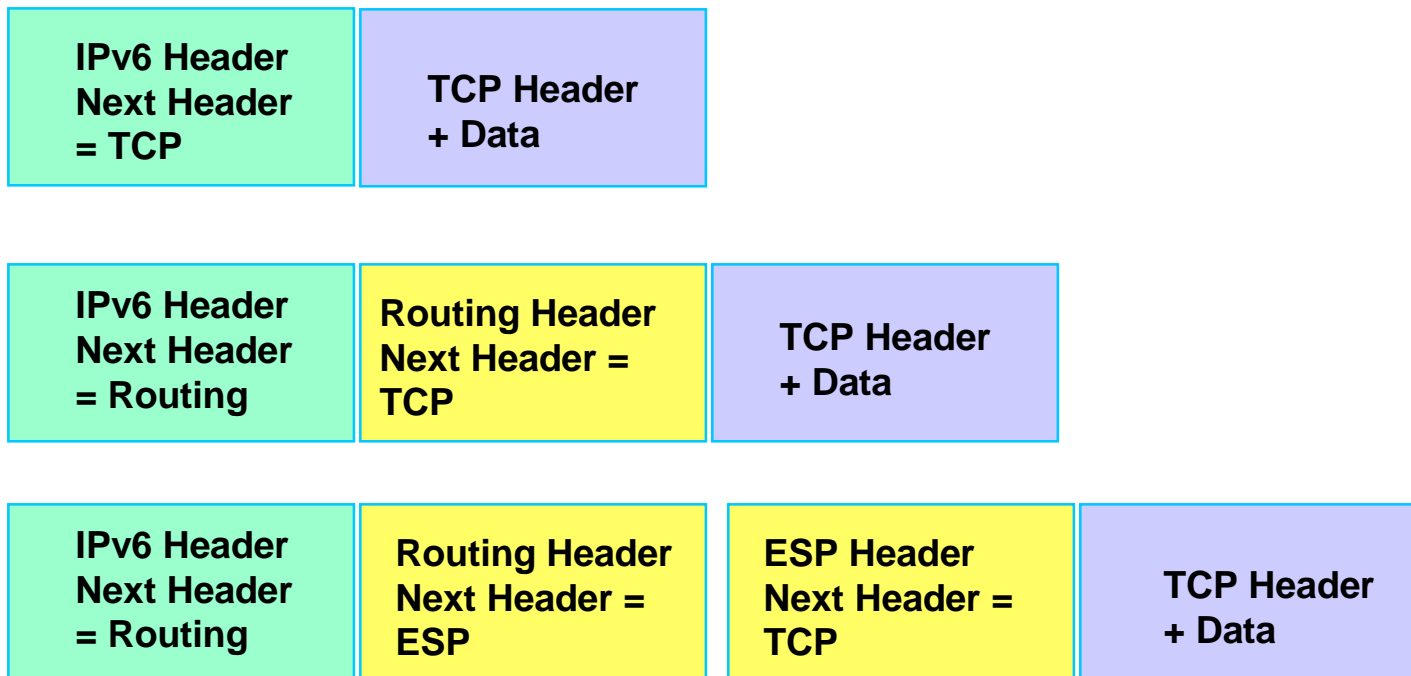
- Version (4 bits)
 - 6 for IPv6
- Traffic Class (8 bits)
 - \sim TOS in IPv4
 - Identifies different classes or priorities (diffserv)
- Flow Label (20 bits)
 - Not yet fully defined
 - Used by a source node to label sequences of packets
- Payload Length
 - \sim Total length in IPv4

- Next Header (8 bits)
 - ~ = Protocol field in IPv4
 - Used to identify the encapsulated protocol
 - TCP, UDP
 - ESP, AH (confidentiality and authentication in IPsec)
 - ICMPv6
 - Other extensions

- Hop Limit \approx TTL in IPv4
- MTU must be at least 1280 bytes
(1500+ recommended)
- Nodes should use Path MTU discovery
- UDP checksum required

Extension Headers

- New way of doing options
- Added after the basic IPv6 header
- Daisy chained



- Hop-by-hop options (0)
 - Information that must be examined by every node along the path
 - Used by Router Alert and Jumbogram
- Routing (43)
 - Similar to IPv4's Loose Source and Record Route option
 - Used by mobileIPv6
- Fragment (44)
 - Used by source node (routers don't fragment anymore!)
- Destination options (60)
 - Used to carry optional information that need to be examined only by a packet's destination node(s)
 - Used by MobileIPv6

- Order of the headers should be the following:
 - IPv6 header
 - Hop-by-Hop Options header
 - Destination Options header (when the routing header is used)
 - Routing header
 - Fragment header
 - Authentication header
 - Encapsulating Security Payload header
 - Destination Options header
 - Upper-layer header
- Source node should follow this order, but destination nodes should be prepared to receive them in any order

- Source Routing
 - Go through this list of routers: A, B, C, D
 - List is included in the routing header
 - Destination address is always the next router in the list, up to the last one where the destination address is the destination node
 - Destination address is changed on every router in the list
- Simpler use:
 - MobileIPv6: Care-of-Address is the "next router" and Home-Address is the final destination

- Comparison of IPv4 and IPv6 headers shows a longer header, but less number of fields
- Header processing is simpler
- Options are handled by extension headers
- Routing header for source routing changes the destination address in the IP header

References



- RFC2460, *Internet Protocol, Version 6 (IPv6) Specification*, S. Deering, R. Hinden, IETF, 1998-12-01, <http://www.normos.org/ietf/rfc/rfc2460.txt>

IPv6 Addressing

- IPv6 addresses
- Format
- Unicast
- Multicast
- Anycast
- Required Node Addresses
- Address Selection
- Addressing Architecture

- IPv4 = 32 bits
- IPv6 = 128 bits
 - This is not 4 times the number of addresses
 - This is 4 times the number of bits
 - $\sim 3,4 * 10^{38}$ possible addressable nodes
 - 10^{30} addresses per person on the planet
 - Well, as with any numbering scheme, we will be using only a portion of the full address space

- X:X:X:X:X:X:X
- Where x is a 16 bits hexadecimal field
 - 2001:0000:1234:0000:0000:C1C0:ABCD:0876
- Case insensitive
 - 2001:0000:1234:0000:0000:c1c0:abcd:0876
- Leading zeros in a field are optional:
 - 2001:0:1234:0:0:C1C0:ABCD:876

- Successive fields of 0 are represented as ::, but only once in an address:
 - 2001:0:1234::C1C0:ABCD:876
 - Not valid: 2001::1234::C1C0:ABCD:876
- Other examples:
 - FF02:0:0:0:0:0:0:1 => FF02::1
 - 0:0:0:0:0:0:0:1 => ::1
 - 0:0:0:0:0:0:0:0 => ::

- In a URL, it is enclosed in brackets
 - `http://[2001:1:4F3A::206:AE14]:8080/index.html`
 - URL parsers have to be modified
 - Cumbersome for users
 - Mostly for diagnostic purposes
 - Should use Fully Qualified Domain Names (FQDN)

- Unicast
 - Unspecified
 - Loopback
 - Scoped addresses:
 - Link-local
 - Site-local
 - Aggregatable Global:
- Multicast
 - Broadcast: none in IPv6
- Anycast

- Used as a placeholder when no address available
 - Initial DHCP request
 - Duplicate Address Detection (DAD)
- Like 0.0.0.0 in IPv4
- 0:0:0:0:0:0:0:0 or ::

- Identifies self
- Localhost
- Like 127.0.0.1 in IPv4
- 0:0:0:0:0:0:0:1 or ::1
- To find if your IPv6 stack works:
 - Ping6 ::1

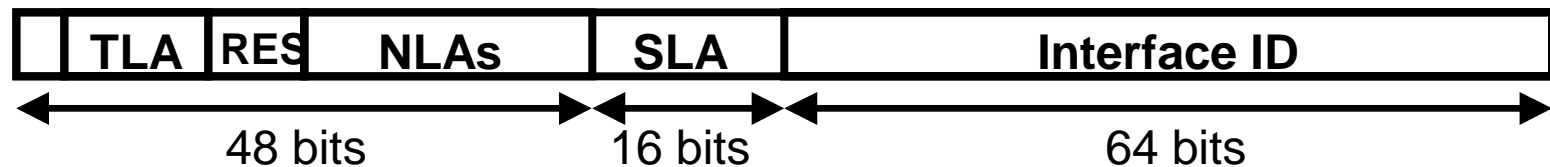
- Scoped address (new in IPv6)
- Scope = local link (i.e. VLAN, subnet)
 - Can only be used between nodes of the same link
 - Cannot be routed
- Automatically configured on each interface
 - Uses the interface identifier (based on MAC address)
- Format:
 - FE80:0:0:0:<interface identifier>
- Gives every node an IPv6 address to start communications

- Scoped address
- Scope = site (a network of links)
 - Can only be used between nodes of the same site
 - Cannot be routed outside the site (i.e. the Internet)
 - Very similar to IPv4 private addresses
- Not configured by default

- Format:
 - FEC0:0:0:<subnet id>:<interface id>
 - Subnet id = 16 bits = 64K subnets
 - Enables an addressing plan for a full site
- Usage example:
 - Number a site before connecting to the Internet:
 - Do your address plan using site locals and use the renumbering functions when connecting to the IPv6 Internet
 - Private addresses (e.g. local printers)

- Generic use. Globally reachable.
- Allocated by IANA
 - To Regional Registries
 - Then to Tier-1 Providers
 - Called Top-level Aggregator (TLA)
 - Then to Intermediate Providers
 - Called Next-level Aggregator (NLA)
 - Then to sites
 - Then to subnets

- Structure:



- 128 bits as the total
- 48 bits prefix to the site
- 16 bits for the subnets in the site
- 64 bits for host part

- Consists of the following (left to right):
 - 3 bits: 001 (10% of the total address space reserved)
 - 13 bits for the TLA
 - 2^{13} TLAs ~ 8K TLAs
 - 8 bits reserved
 - 24 bits for the NLAs
 - 2^{24} NLAs per TLA ~ 16M NLAs per TLA
 - 16 bits for the site subnets
 - 2^{16} subnets per site = 65536 subnets
 - 64 bits for the interface identifier
 - Total = 128 bits.

- Multicast = one-to-many
- No broadcast in IPv6. Multicast is used instead, mostly on local links
- Scoped addresses:
 - Node, link, site, organisation, global
 - No TTL as in IPv4
- Format:
 - FF<flags><scope>::<multicast group>

Multicast Assigned Addresses



- Some reserved multicast addresses:

Address	Scope	Use
FF01::1	Interface-local	All Nodes
FF02::1	Link-local	All Nodes
FF01::2	Interface -local	All Routers
FF02::2	Link-local	All Routers
FF05::2	Site-local	All Routers
FF02::1:FFXX:XXXX	Link-local	Solicited-Node

- One-to-nearest: great for discovery functions
- Anycast addresses are indistinguishable from unicast addresses
 - Allocated from the unicast addresses space
 - Some anycast addresses are reserved for specific uses
- Few uses:
 - Router-subnet
 - MobileIPv6 home-agent discovery
 - discussions for DNS discovery

- Any IPv6 node should recognize the following addresses as identifying itself:
 - Link-local address for each interface
 - Assigned (manually or automatically) unicast/anycast addresses
 - Loopback address
 - All-nodes multicast address
 - Solicited-node multicast address for each of its assigned unicast and anycast address
 - Multicast address of all other groups to which the host belongs

- Any IPv6 router should recognize the following addresses as identifying itself:
 - All the required node addresses
 - All-routers multicast addresses
 - Specific multicast addresses for routing protocols
 - Subnet-router anycast addresses for the interfaces configured to act as forwarding interfaces
 - Other anycast configured addresses

- A node has many IPv6 addresses
- Which one to use as source and destination address for a given communication?
- Some issues to be addressed:
 - Scoped addresses are unreachable depending on the destination
 - Preferred vs deprecated addresses
 - IPv4 or IPv6 when DNS returns both
 - IPv4 local scope (169.254/16) and IPv6 global scope. IPv6 local scope and IPv4 global scope
 - MobileIP addresses, temporary addresses, scope addresses, etc

- Algorithm:
 - Prefer same address
 - Prefer appropriate scope
 - Avoid deprecated addresses
 - Prefer home addresses
 - Prefer outgoing interface
 - Prefer matching label
 - Prefer temporary addresses
 - Use longest matching prefix

- Algorithm is basically:
 - Use the right scope based on the destination
 - Use the most similar address
 - Use home address instead of care-of-address, if in the mobility context
- Default policy can be overridden by stack or application

Addressing Architecture



Prefix	Hex	Size	Allocation
0000 0000	0000-00FF	1/256	Reserved
0000 0001	0100-01FF	1/256	Unassigned
0000 001	0200-03FF	1/128	NSAP
0000 010	0400-05FF	1/128	Unassigned
0000 011	0600-07FF	1/128	Unassigned
0000 1	0800-0FFF	1/32	Unassigned
0001	1000-1FFF	1/16	Unassigned
001	2000-3FFF	1/8	Aggregatable: IANA to registries

Addressing Architecture (cont.)

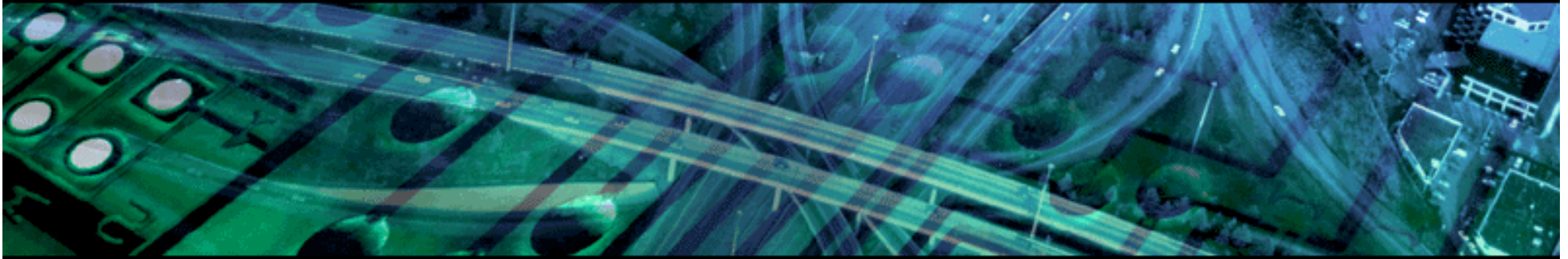


Prefix	Hex	Size	Allocation
010, 011, 100, 101, 110	4000-CFFF	$5 * 1/8 = 5/8$	Unassigned
1110	D000-EFFF	1/16	Unassigned
1111 0	F000-F7FF	1/32	Unassigned
1111 10	F800-FBFF	1/64	Unassigned
1111 110	FC00-FDFF	1/128	Unassigned
1111 1110 0	FE00-FE7F	1/512	Unassigned
1111 1110 10	FE80-FEBF	1/1024	Link-local
1111 1110 11	FEC0-FEFF	1/1024	Site-local
1111 1111	FF00-FFFF	1/256	Multicast

- IPv6
 - Has a much larger address space
 - Has specific formatting for addresses
 - Introduces new kind of addresses (scoped)
- IPv6 nodes have many addresses and need to select which one to use
- Addressing architecture has a lot of space available for future use

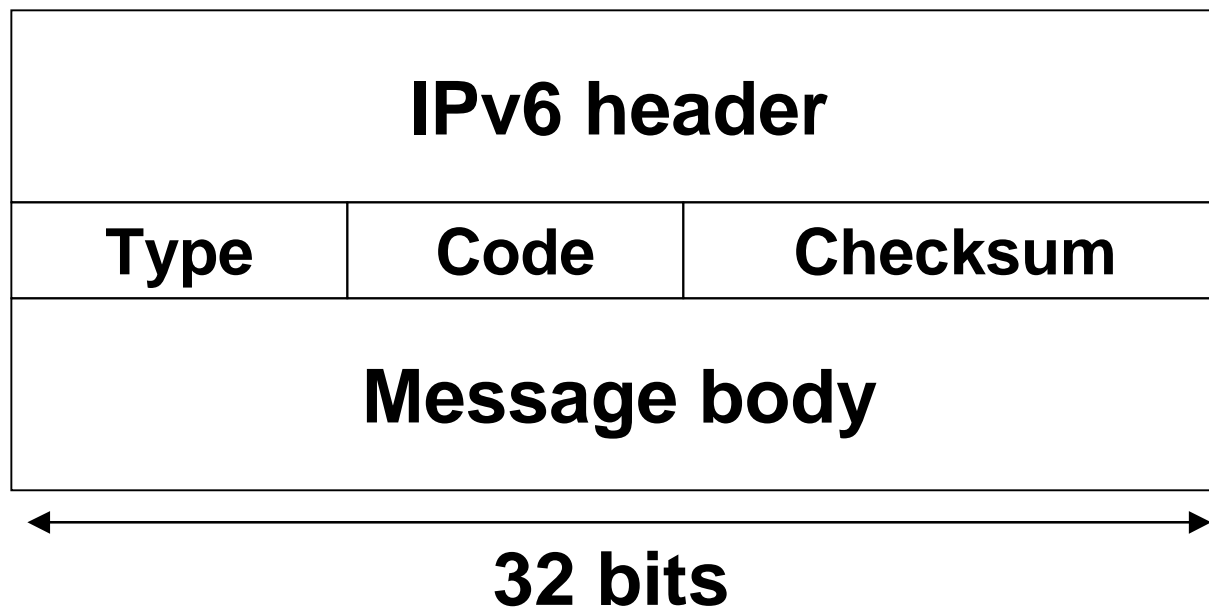
- RFC2373, *IP Version 6 Addressing Architecture*, R. Hinden, S. Deering, IETF, 1998-07-01, <http://www.normos.org/ietf/rfc/rfc2373.txt>
- RFC2374, *An IPv6 Aggregatable Global Unicast Address Format*, R. Hinden, M. O'Dell, S. Deering, IETF, 1998-07-01, <http://www.normos.org/ietf/rfc/rfc2374.txt>
- *IP Version 6 Addressing Architecture*, R. Hinden, S. Deering, IETF internet-draft, July 2001, <http://www.normos.org/ietf/draft/draft-ietf-ipngwg-addr-arch-v3-06.txt>
- *Default Address Selection for IPv6*, Richard Draves, IETF internet-draft, June 2001, <http://www.normos.org/ietf/draft/draft-ietf-ipngwg-default-addr-select-05.txt>

ICMP, Neighbor Discovery and Autoconfiguration



- ICMP
- Path MTU Discovery
- Neighbor Discovery
- Autoconfiguration
- Renumbering
- Duplicate Address Detection
- Temporary Addresses

- Internet Control Message Protocol
- Same behaviour as in IPv4, but few enhancements
- IPv6 Next Header= 58



- Many messages are the same as the IPv4 counterpart:
 - Type 1: Destination Unreachable
 - Type 2: Packet Too Big (MTU)
 - Type 3: Time Exceeded
 - Type 4: Parameter Problem
 - Type 128/129: Echo request/Echo reply

- No fragmentation done by routers in IPv6
- Fragmentation, if needed, is done by the source
- Source should do Path MTU Discovery to find the right MTU
- Minimum MTU is 1280

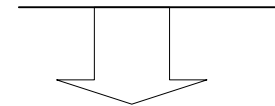
1. Send a message to the destination with MTU of your link
2. If receive a ICMP error message, then resend the message with the new MTU
3. Do 1 and 2 until response from destination
4. Last MTU is the Path MTU

- Solicited-Node multicast address

- FF02:0:0:0:0:1:FF00::/104

- address formed by appending the lower 24 bits of the IPv6 address
 - a node is required to join every unicast and anycast address it is assigned

3FFE:0B00:0C18:0001:0290:27FF:FE17:FC0F
Global unicast address



FF02:0000:0000:0000:0000:0001:FF17:FC0F
Solicited multicast address

- Replaces IPv4 ARP, plus new features
- Uses ICMPv6 messages
- Used to:
 - Find link-layer address of neighbor
 - Find neighboring routers
 - Actively keep track of neighbor reachability
 - Send network information from routers to hosts
- Protocol used for host autoconfiguration
- All ND messages must have Hop Limit=255
 - Must originate and terminate from the same link

- Sent by a node to determine link-layer address of a neighbor
- =~ ARP request
- Packet description:
 - Source Address = link-local address
 - Destination = solicited-node multicast address
 - Data contains link-layer address of source (for efficiency purposes)
 - Query is: Please send me your link-layer address
 - ICMP type 135

- Response to a Neighbor Solicitation
- =~ ARP response
- Includes my MAC address so you can send me information
- Packet description:
 - Source Address = link-local address of source
 - Destination = destination address
 - Data contains link-layer address of mine
 - ICMP type 136

- Routers advertise periodically
 - Max. time between advertisements can be in the range from 4 and 1800 seconds
 - The advertisement has a lifetime (= 0 if not a default router)
- Advertisement contains one or more prefixes
 - Prefixes have a lifetime
 - Preferred lifetime
 - Valid lifetime
- Specifies if stateful or stateless autoconfiguration is to be used
- Plays a key role in site renumbering

- Packet description:
 - Source: router address on the link
 - Destination: multicast address of all nodes on the link (FF02::1)
 - Data: prefix, lifetimes, default router, options
 - ICMP type 134

- When booting, nodes don't want to wait until the next router advertisement to configure themselves
- Host then request routers to send Router Advertisement immediately
- Packet description:
 - Source: link-local address
 - Destination: multicast address all-routers on this link (FF02::2)
 - ICMP type 133

- ~ = ICMP redirect
- Route change
- Router send better hop for a destination
- Packet description:
 - Source: router address
 - Destination: source node of the packet that needs rerouting
 - Data: better router address

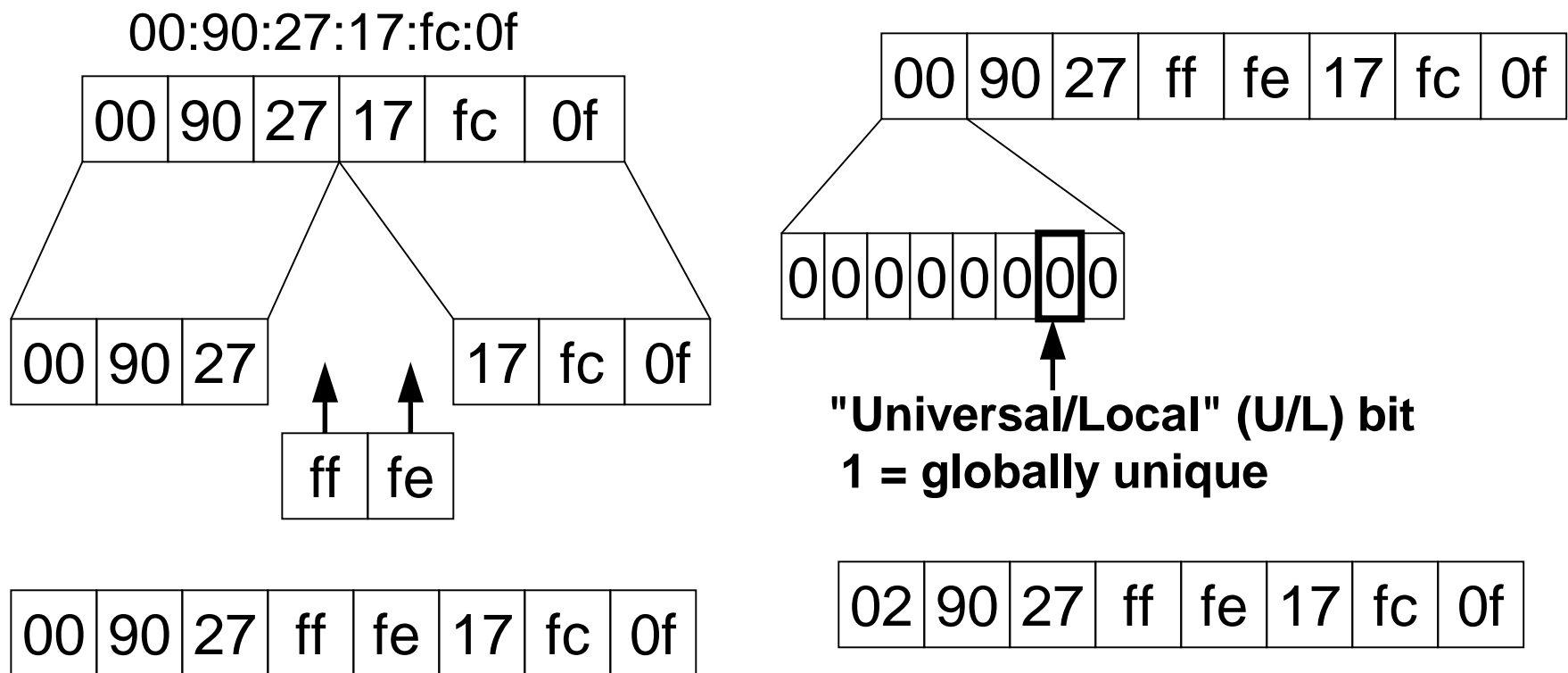
- Stateful configuration
 - Manual IP configuration
 - DHCP configuration
- Stateless Address Autoconfiguration
 - Applies to hosts only (not to routers)
 - No manual configuration required
 - Specifies the prefix, default route and lifetime
 - But does not specify the DNS servers
 - Assumes interface has unique identifier
 - Assumes multicast capable link
 - Uses Duplicate Address Detection

- Host configured for autoconfiguration
- Host boots. Sends a Router Solicitation
- Host receives the Router Advertisement, specifying subnet prefix, lifetimes, default router ...
- Host generates its IP address by appending:
 - Received subnet prefix (64 bits)
 - Interface address modified for EUI-64 format
- Host verifies usability of the address by doing the Duplicate Address Detection process

Autoconfiguration: IEEE 802 48bit MAC address to EUI-64



- Interface Identifier for stateless autoconfiguration



So lower 64 bits in address are **02:90:27:ff:fe:17:fc:0f**

- Hosts renumbering
 - On the router, decrease the lifetime of the prefix in the router advertisement
 - Preferred lifetime = 0. (this "old" address cannot be used for new connections)
 - Valid lifetime decreasing
 - Start advertising the new prefix
 - Hosts configure the new address and start using it
- Connections will continue without interruption
- Hosts must always listen to router advertisements, even after autoconfiguration

- Router Renumbering
 - Protocol to renumber routers within a site
 - Defines new ICMPv6 messages
 - Uses IPsec for authentication purposes
- At this time, not many implementations known

- Similar to IPv4 ARP self
- Join all-nodes multicast address (FF02::1)
- Join solicited-node multicast address of the tentative address
 - FF02:0:0:0:0:1:FFxx: <last 24 bits of my new address>
- Send Neighbor Solicitation on solicited-node multicast address
- If no Neighbor Advertisement is received, address is ok

- Traceability of the MAC address in IPv6 address, if using autoconfiguration all time
- Privacy concerns
- Algorithm (RFC3041) defined to:
 - Generate a random address for the rightmost 64 bits
 - Define it as temporary
 - Recycle it as needed

- ICMPv6 is similar to IPv4, but is enhanced for Neighbor Discovery functions
- Path MTU discovery is used
- Neighbor discovery enables ARP like functions and autoconfiguration
- Renumbering is achieved by modifying the advertisements of prefixes
- Duplicate address detection is used to ensure uniqueness of addresses on link
- Temporary addresses are used in case of privacy concerns

References



- *RFC1981, Path MTU Discovery for IP version 6, J. McCann, S. Deering, J. Mogul, IETF, 1996-08-01, <http://www.normos.org/ietf/rfc/rfc1981.txt>*
- *RFC2461, Neighbor Discovery for IP Version 6 (IPv6), T. Narten, E. Nordmark, W. Simpson, IETF, 1998-12-01, <http://www.normos.org/ietf/rfc/rfc2461.txt>*
- *RFC2462, IPv6 Stateless Address Autoconfiguration, S. Thomson, T. Narten, IETF, 1998-12-01, <http://www.normos.org/ietf/rfc/rfc2462.txt>*
- *RFC2463, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, A. Conta, S. Deering, IETF, 1998-12-01, <http://www.normos.org/ietf/rfc/rfc2463.txt>*
- *RFC2894, Router Renumbering for IPv6, M. Crawford, IETF, 2000-08-01, <http://www.normos.org/ietf/rfc/rfc2894.txt>*
- *RFC3041, Privacy Extensions for Stateless Address Autoconfiguration in IPv6, T. Narten, R. Draves, IETF, 2001-01-01, <http://www.normos.org/ietf/rfc/rfc3041.txt>*

IPv6 and DNS

- Name to IPv6 Address
- IPv6 Address to Name
- Transport
- Root Servers
- Transition

- New records:
 - AAAA
 - Defines the mapping from the domain name to the IPv6 address
 - Equivalent to the IPv4 A record
 - Supported in Bind since 4.9.5

- PTR record
 - Defines the mapping from an IPv6 address to a name
 - Same record as for IPv4
 - New top level for the IPv6 space is used: was ip6.int, moving to ip6.arpa)

- AAAA records

```
$ORIGIN ipv6.viagenie.qc.ca  
www in aaaa 3ffe:b00:c18:1:290:27ff:fe17:fc1d
```

- PTR records (ip6.arpa)

```
$ORIGIN 1.0.0.0.8.1.c.0.0.0.b.0.e.f.f.3.ip6.arpa  
d.1.c.f.7.1.e.f.f.f.7.2.0.9.2.0 in ptr www.ipv6.viagenie.qc.ca
```

- IPv6 data queries over IPv4 and IPv6
 - Bind4-8 answers to queries over IPv4 transport only
 - Bind 9 can answer to queries over IPv6 transport

- Not configured for IPv6 native queries now
- But AAAA records can be used on the current root servers
- Issues concerning:
 - the maximum size of data in the hints send back to clients if all root servers are all IPv4 and IPv6: too much space in the return packet
 - Cache pollution if IPv6 root servers change over time

- New record:
 - AAAA
- PTR is same but with a different root ip6.arpa
- Root servers on IPv6 have some issues

References



- *RFC1886, DNS Extensions to support IP version 6, S. Thomson, C. Huitema, IETF, 1995-12-01, <http://www.normos.org/ietf/rfc/rfc1886.txt>*
- *RFC2874, DNS Extensions to Support IPv6 Address Aggregation and Renumbering, M. Crawford, C. Huitema, IETF, 2000-07-01, <http://www.normos.org/ietf/rfc/rfc2874.txt>*
- *NGtrans IPv6 DNS operational requirements and roadmap, Alain Durand, Johan Ihren, IETF internet-draft, Sept 2001, <http://www.normos.org/ietf/draft/draft-ietf-ngtrans-dns-ops-req-02.txt>*

IPv6 Routing Protocols

- RIP
- OSPF
- ISIS
- BGP

- RIP (Routing information protocol) in IPv6
 - Based on RIP-2: same design: distance-vector, 15 hops diameter
 - ...
 - IPv6 prefix, next-hop IPv6 address
 - Uses multicast (FF02::9 = all-rip-routers as the destination address for RIP updates)
 - Uses IPv6 for transport
 - Most (if not all) IPv6 router implementations support RIP IPv6 Most IPv6-enabled Unix OS have the IPv6-RIP routed daemon available

- OSPF (Open Shortest Path First) for IPv6
 - Also known as OSPFv3
 - Important rewrite to remove IPv4 dependencies. Now it is network protocol independent
 - Link-local addresses are used
 - Uses IPv6 for transport
 - The important rewrite on the protocol has delayed the implementations. This results in an important delay for the deployment of IPv6 in large networks

- IS-IS is the OSI IGP protocol. It is network protocol independent
- Compared to OSPF, IS-IS for IPv6 is easier to implement and modify
 - 2 new type-length-values (TLV) were defined:
 - IPv6 Reachability (with 128 bits prefix)
 - IPv6 Interface Address (with 128 bits)
 - New protocol identifier for IPv6
- Router vendors supporting IS-IS for IPv4 have or are probably going to have an IPv6 version

- BGP4+
 - Includes multiprotocol extensions for BGP, for new address families (ex: IPv6, VPN, ...)
- IPv6 address family:
 - Use scoped addresses in the NEXT_HOP
 - NEXT_HOP and NLRI are expressed as IPv6 addresses and prefix
- Most IPv6 router vendors support IPv6 BGP. It has been used on the 6Bone since 1996

- All routing protocols are available for IPv6
- No real difference in the mechanics
- RIPv6 is based on RIP-2
- OSPFv6 is a major rewrite
- IS-ISv6 includes few changes
- BGPv6 is based on Multiprotocol BGP

References



- *RFC2080, RIPng for IPv6*, G. Malkin, R. Minnear, IETF, 1997-01-01, <http://www.normos.org/ietf/rfc/rfc2080.txt>
- RFC2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*, P. Marques, F. Dupont, IETF, 1999-03-01, <http://www.normos.org/ietf/rfc/rfc2545.txt>
- RFC2740, *OSPF for IPv6*, R. Coltun, D. Ferguson, J. Moy, IETF, 1999-12-01, <http://www.normos.org/ietf/rfc/rfc2740.txt>
- RFC2858, *Multiprotocol Extensions for BGP-4*, T. Bates, Y. Rekhter, R. Chandra, D. Katz, IETF, 2000-06-01, <http://www.normos.org/ietf/rfc/rfc2858.txt>
- *Routing IPv6 with IS-IS*, draft-ietf-isis-ipv6-02.txt

IPv6 Transition

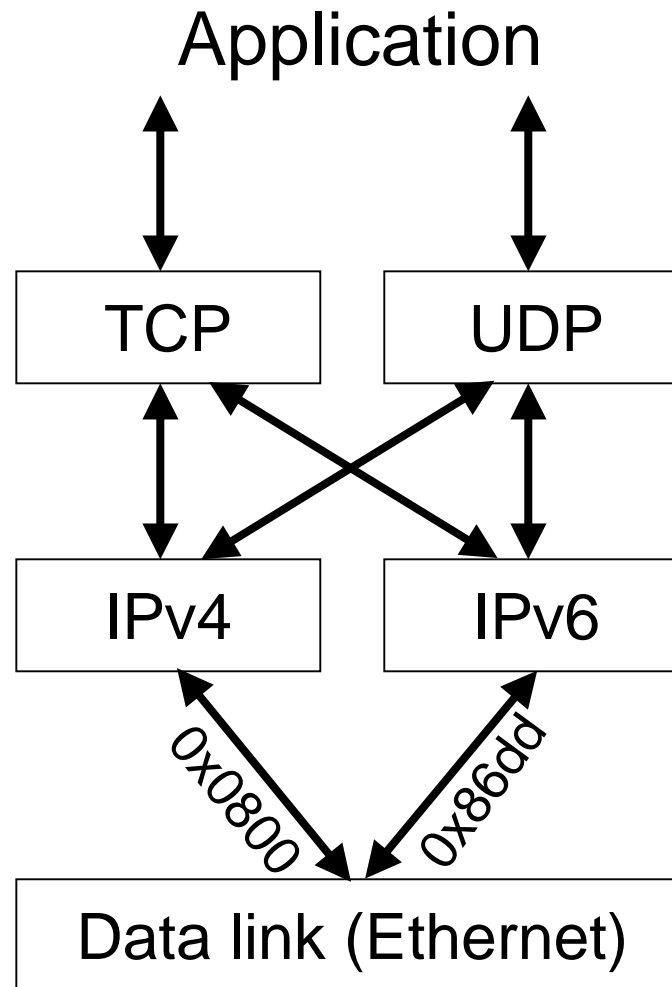


- Context
- Dual Stack
- Tunnelling over IPv4
 - Configured tunnels
 - Automatic tunnelling
 - IPv4-compatible addresses
 - 6to4
 - Tunnel Broker/server

- When moving to another technology, the transition has to be discussed and is generally very important. Often it is where most of the money is put
- Many new technologies didn't succeed because of the lack of transition scenarios/tools
- IPv6 was designed, **at the beginning**, with transition in mind: no D day
- IPv6 is transition-rich, as you will see

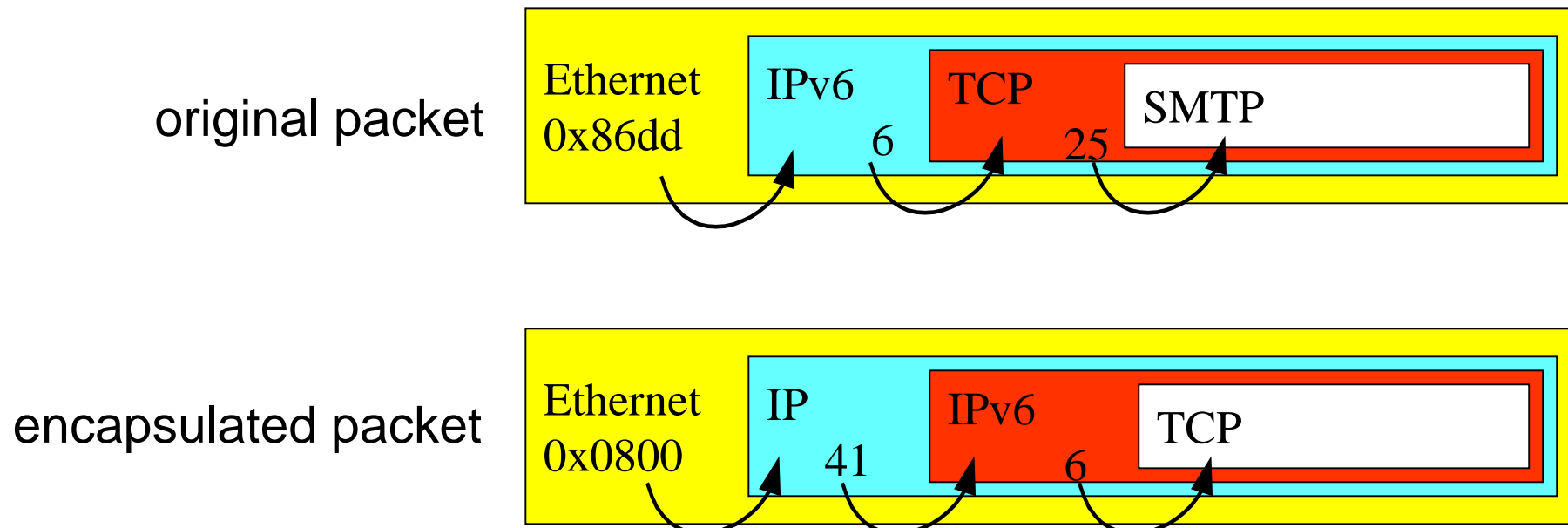
- For end-systems, there is:
 - Dual stack approach
- For network integration, there is:
 - Tunnels
 - IPv6-only to IPv4-only: some sort of translation

- Node has both IPv4 and IPv6 stacks and addresses
- IPv6-aware application asks for both IPv4 and IPv6 addresses of destination
- DNS resolver returns IPv6, IPv4 or both addresses to application
- IPv6/IPv4 applications choose the address and then can communicate
 - With IPv4 nodes using IPv4
 - Or with IPv6 nodes using IPv6



- IPv6 encapsulated in IPv4
 - IP protocol 41
- Many topologies possible
 - Router to router
 - Host to router
 - Host to host
- The tunnel endpoints take care of the encapsulation. This process is “transparent” for the intermediate nodes
- Tunnelling is used by most transition mechanisms

Tunnelling IPv6 in IPv4

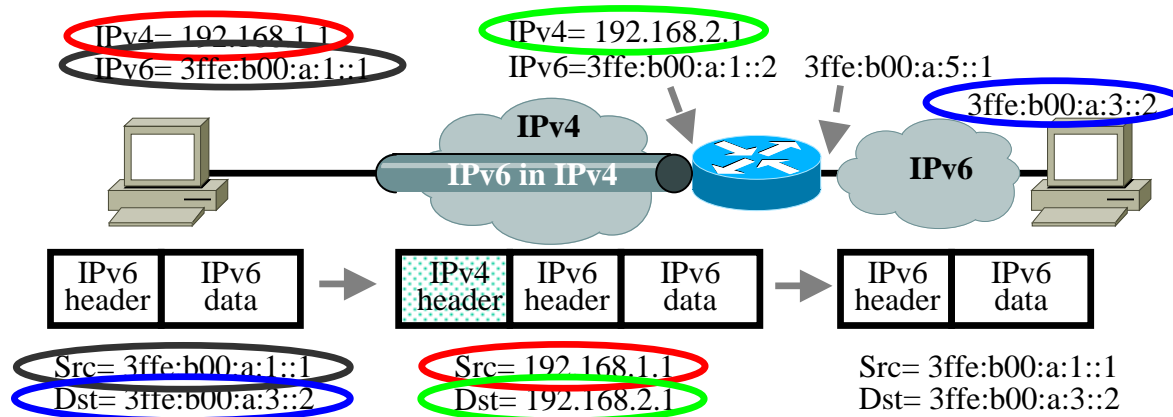


- NOTE: if present in the path, security gateways will need to let through IP packets transporting protocol 41
- Problem if NAPT is in the path
- Can also be accomplished by GRE tunnels

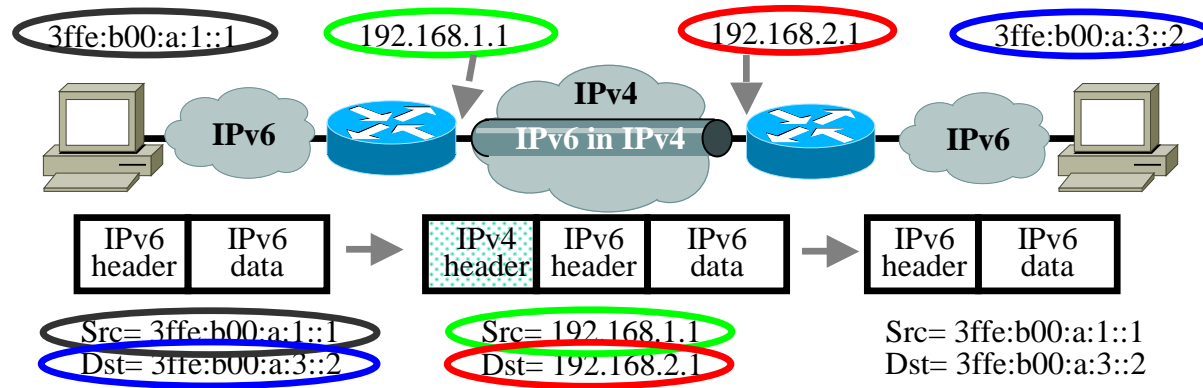
- Configured tunnels
- Automatic tunnels
- 6to4
- Tunnel broker

- Tunnel endpoints are explicitly configured
- Tunnel endpoints must be dual stack nodes
 - The IPv4 address is the endpoint for the tunnel
 - Require a reachable IPv4 address (no NAT)
- Tunnel config implies:
 - Manual configuration of:
 - Source and destination IPv4 address
 - Source and destination IPv6 address
- Between:
 - Two hosts
 - One host and one router
 - Two routers (for two networks)

Configured Tunnels



Configured Tunnels



- Tunnels cannot go through a NAPT
- If site uses a NAPT, then one scenario might be to end the tunnel at the NAPT box
- A possible transition: when IPv4 addresses are scarce, deploy private IPv4 addresses using a NAPT and deploy IPv6 end-to-end

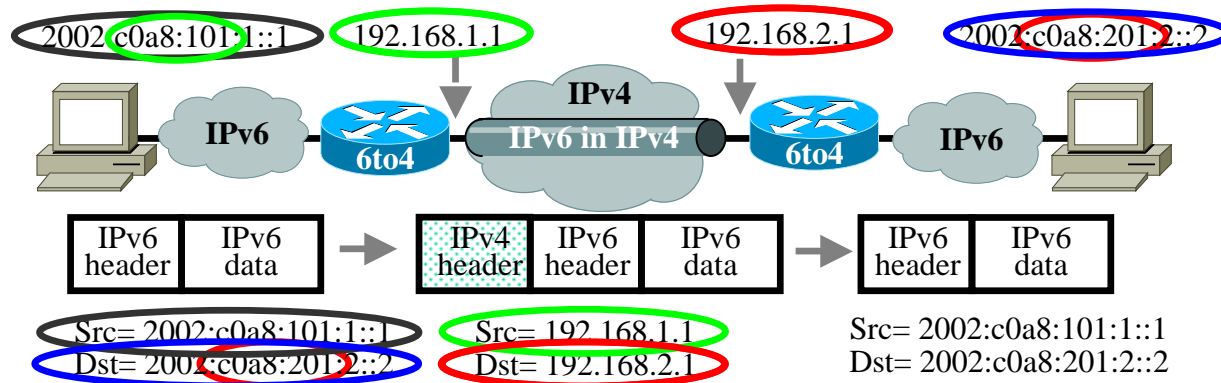
- Applicability: interconnection of isolated IPv6 domains over an IPv4 network
- Automatic establishment of the tunnel
 - No explicit tunnels
 - By embedding the IPv4 destination address in the IPv6 address
 - Under the 2002::/16 reserved prefix. (2002::/16 = 6to4)
- Gives a full /48 to a site based on its external IPv4 address
 - IPv4 external address embedded: 2002:<ipv4 ext address>::/48
 - Format: 2002:<ipv4add>:<subnet>::/64

Who Needs to Support 6to4?

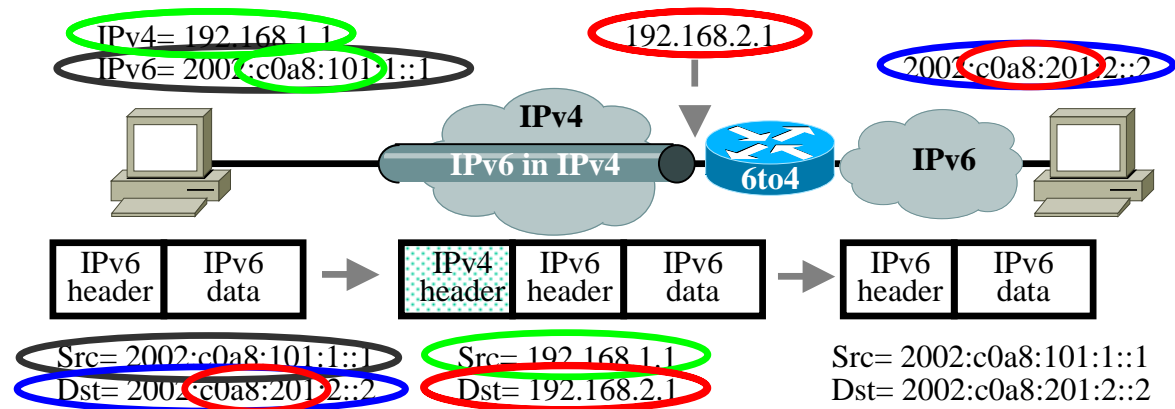


- Egress router:
 - Implements 6to4
 - Must have a reachable external IPv4 address
 - Often configured using a loopback interface address
 - Is a dual-stack node
- Individual nodes:
 - Nothing needed for 6to4 support. 2002 is an "ordinary" prefix that may be received from router advertisements
 - Doesn't need to be dual-stack

6to4 Network to Network



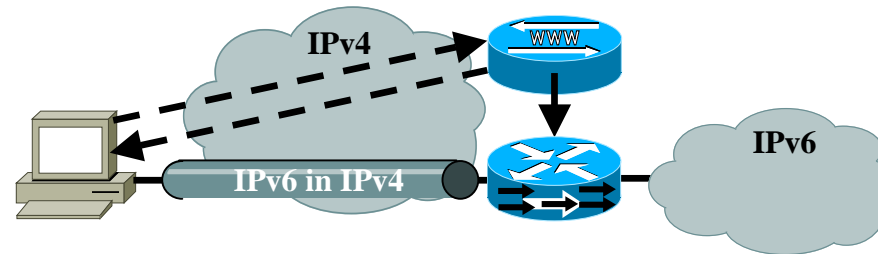
6to4 Host to Network



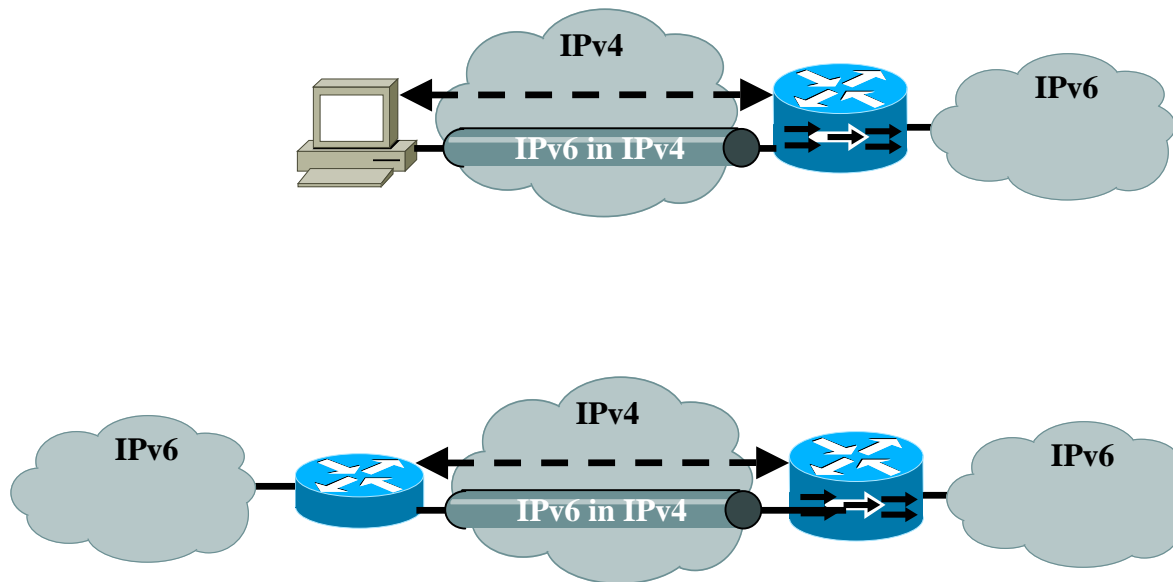
- Bound to the IPv4 external address:
 - If egress router changes its IPv4 address, then it means that you need to renumber the full IPv6 internal network
 - Only one entry point (no easy way to have multiple network entry points for redundancy)

- Discovery of the 6to4 relay (or IPv6 default route)
 - Uses anycast reserved address
- DNS inverse delegation
- Multicast
- Integration with DSTM

- Semi-automated tunnel configuration
- 1st generation Broker
 - Is a web server receiving requests from clients
 - Generates the tunnel and sends back info to client
 - Configures the server (or router)
 - In fact, this automates the manual configuration of tunnels (with explicit IPv4 source and destination addresses, and IPv6 source and destination addresses)
- Tunnel server model:
 - Degenerated (simpler) case where the broker and server are the same node
 - Popular free service: <http://www.freenet6.net>



Tunnel Server



- New functionalities:
 - Now implemented as an interactive protocol:
 - Called Tunnel Setup Protocol (TSP)
 - Client and servers can make requests/responses
 - I.e. client requests:
 - A tunnel for one host
 - A tunnel for a network (routing implied)
 - » With a prefix delegation
 - » Without a prefix delegation (I have mine, but please announce it)
 - With routing information:
 - » I use RIP, BGP, OSPF, ...
 - With domain name information
 - » The host name will be:
 - » Inverse delegation

- New functionalities:
 - Now implemented as an interactive protocol:
 - Server can respond:
 - I'm full, cannot give you a new tunnel, please go to this other tunnel server (referer)
 - I can give you a host tunnel, but not a network tunnel
 - Here is the prefix, here is my bgp info (as number) ...
 - Enables:
 - Changes of data to be negotiated between the two parties: i.e. IPv4 client address change
 - Generic Authentication: use a password, a Securid card, a public-key...
 - Can be implemented
 - In the boot sequence of a host
 - In host OS and router OS

Communication Between IPv6 and IPv4 Nodes



- How do IPv6 hosts communicate with legacy IPv4 only hosts?
 - Old printers, network equipment ...
- Many ways to do this, the simplest one is the dual stack host

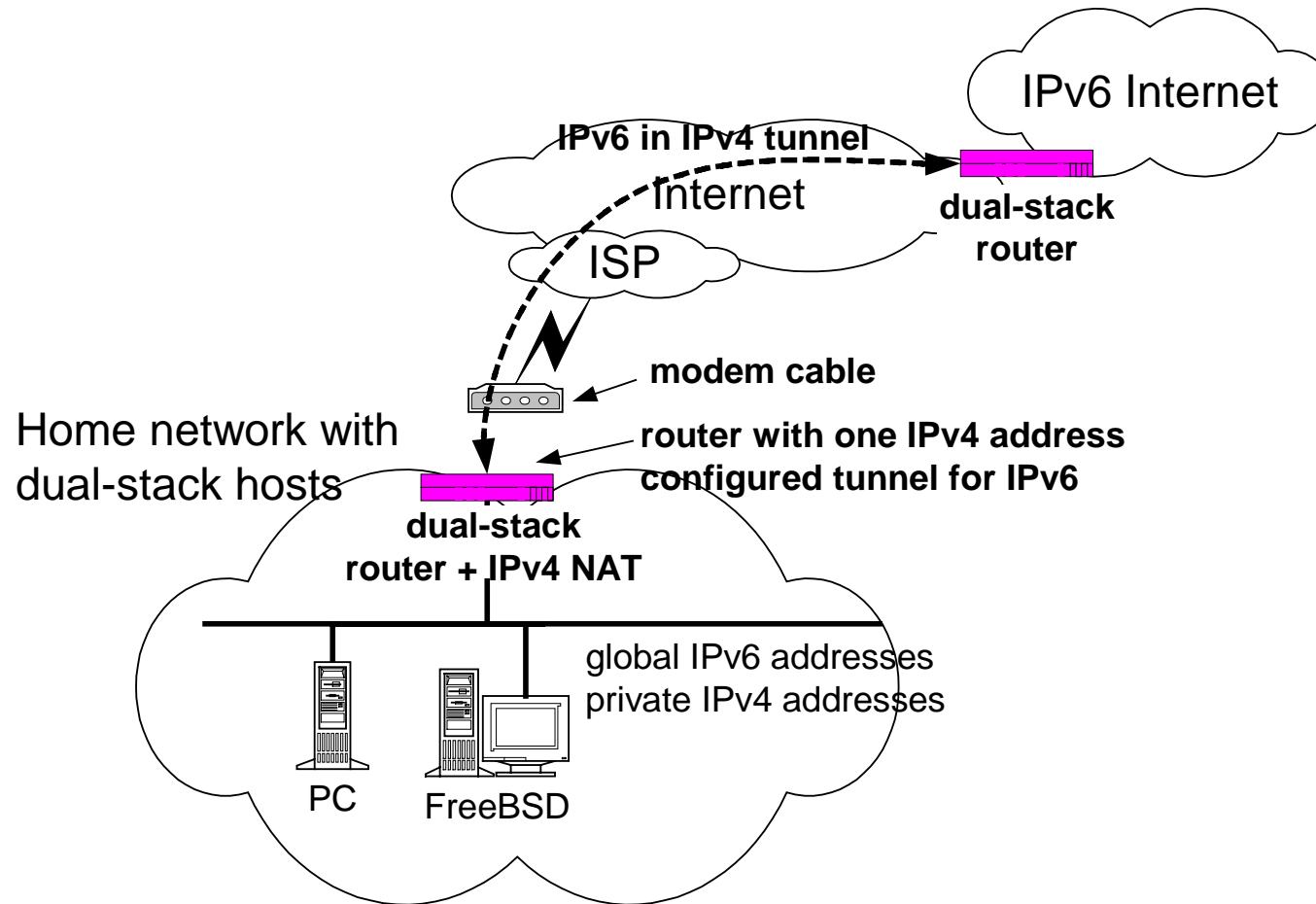
- **Dual stack host**
 - When the host initiates a communication, the DNS will provide either an IPv6 address, an IPv4 address or both
 - The host will then establish the communication using the appropriate IP stack
 - Same scenario for a server: listens on both IPv4 and IPv6 network socket
- **But every host needs an IPv4 address**

- Dual stacks connects IPv4/IPv6 nodes to IPv4-only nodes or IPv6-only nodes
- Tunnelling connects IPv6 islands together through an IPv4 network
- Translation is used for connecting IPv4-only to IPv6-only
- Where to do translation:
 - IP layer
 - Transport layer
 - Application layer

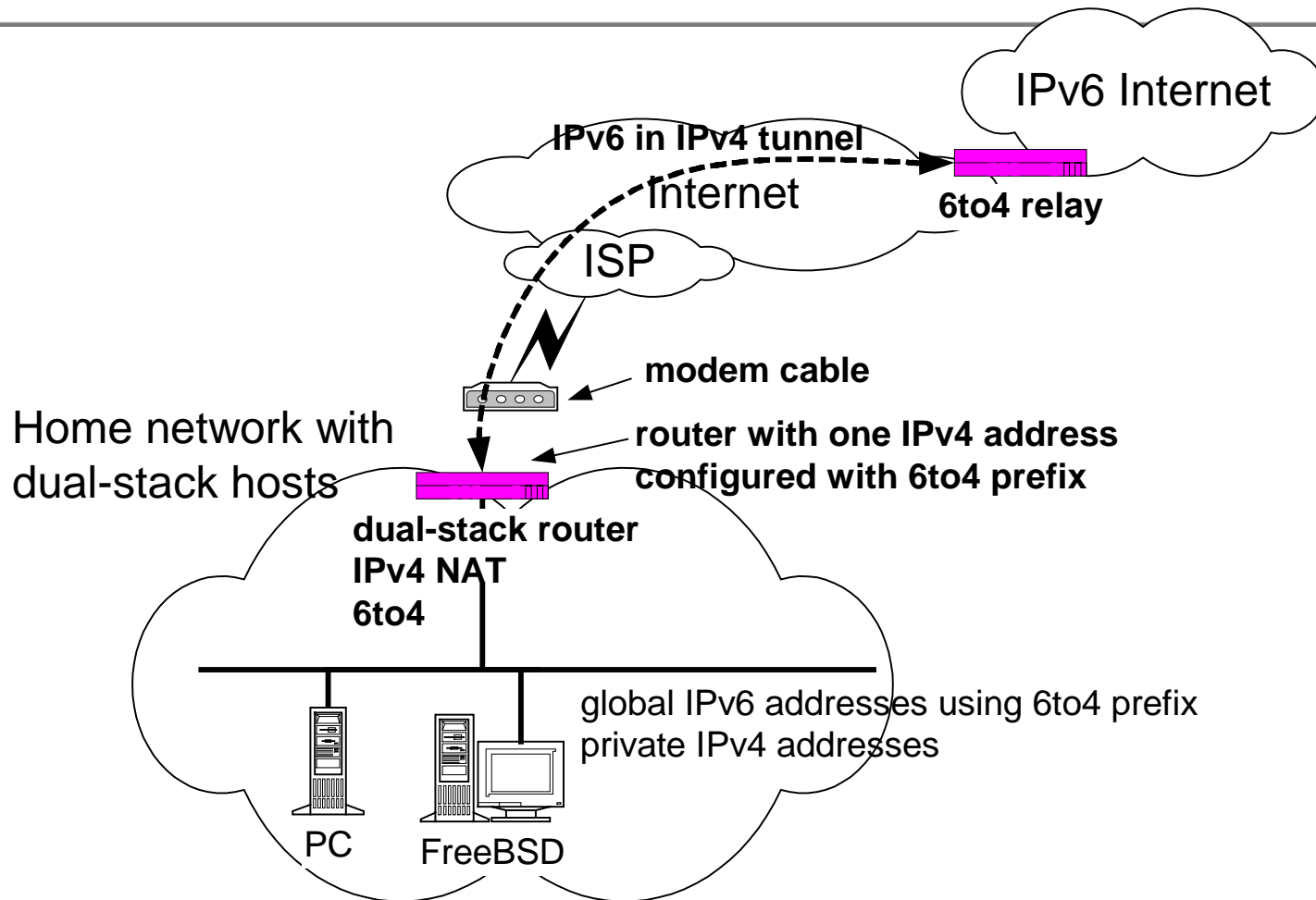
- Small IPv4 address space (/24)
- IPv6 site prefix (/48)
- IPv4 and dual-stack hosts on site
- If number of hosts is moderate, it needs IPv4 NAT device and private IPv4 addresses
 - NAT can be NAT-PT

- Install dual-stack device as egress router (NAT device if installed)
- IPv6 connectivity through
 - ISP: nothing else to configure but your network prefix assigned by ISP
 - Configured tunnel
 - 6to4: configure default route through a 6to4 relay

Typical Scenario With Configured Tunnel



Typical Scenario With 6to4



- A Guide to the Introduction of IPv6 in the IPv4 World
 - [draft-ietf-ngtrans-introduction-to-ipv6-transition-xx.txt](#)
 - Lists "all" transition mechanisms with a small description. Try to classify them
 - Other information for IPv6 deployment

- Deploy only IPv6
 - New networks
 - Private networks (control of industrial devices)
- Convert your application
 - Quake experience at Viagénie:
 - 300K lines of C code
 - 2 days for a C programmer:
 - Knows how socket calls works
 - New to IPv6
 - Few hours for a experienced converter

- Many transition tools exists:
 - To tunnel between IPv6 islands
 - To translate between IPv4 and IPv6
- Others are available, others will be define
- None is for all possible scenarios
- Not all will succeed on the market
- Choose the right one for your scenario

- IPv6 is transition-rich
- Basic transition technique is using the dual-stack approach
- Many techniques exist for tunnelling, for translation and for proxying
- Techniques should be applied in the right context

- RFC2529, *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*, B. Carpenter, C. Jung, IETF, 1999-03-01, <http://www.normos.org/ietf/rfc/rfc2529.txt>
- RFC2766, *Network Address Translation - Protocol Translation (NAT-PT)*, G. Tsirtsis, P. Srisuresh, IETF, 2000-02-01, <http://www.normos.org/ietf/rfc/rfc2766.txt>
- RFC2767, *Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)*, K. Tsuchiya, H. Higuchi, Y. Atarashi, IETF, 2000-02-01, <http://www.normos.org/ietf/rfc/rfc2767.txt>
- RFC2893, *Transition Mechanisms for IPv6 Hosts and Routers*, R. Gilligan, E. Nordmark, 2000-08-01, <http://www.normos.org/ietf/rfc/rfc2893.txt>
- RFC3053, *IPv6 Tunnel Broker*, A. Durand, P. Fasano, I. Guardini, D. Lento, IETF, 2001-01-01, <http://www.normos.org/ietf/rfc/rfc3053.txt>
- RFC3056, *Connection of IPv6 Domains via IPv4 Clouds*, B. Carpenter, K. Moore, IETF, 2001-02-01, <http://www.normos.org/ietf/rfc/rfc3056.txt>
- RFC3068, *An Anycast Prefix for 6to4 Relay Routers*, C. Huitema, IETF, 2001-06-01, <http://www.normos.org/ietf/rfc/rfc3068.txt>
- RFC3142, *An IPv6-to-IPv4 Transport Relay Translator*, J. Hagino, K. Yamamoto, IETF, 2001-06-01, <http://www.normos.org/ietf/rfc/rfc3142.txt>
- draft-ietf-ngtrans-introduction-to-ipv6-transition-04.txt, An overview of the introduction of IPv6 in the Internet

IPv6 on Cisco

- Roadmap
- Enabling IPv6
- Interface
- Routing
- Tunnels

- Implementation since 1995
- RIP and BGP since the beginning
- Bundled and supported in 12.2(2)T
 - RIP, BGP, ND, RA, Autoconfiguration, ICMP, 6to4, Standard access lists, IP over typical medias, utilities (ping, traceroute, ...)
 - All hardware except GSR and router blades in switches
 - Unsupported in current release:
 - DAD, ICMP redirect

- Next major release 12.2(3)T:
 - ISIS, CEF, NATPT, Dial, Basic Mibs
- After:
 - OSPF, IPsec, MobileIP, Multicast, QOS, Netflow, Encapsulation on other medias, MIBs+, Hardware support

- `ipv6 unicast-routing`
 - Global context
 - Enables the router to act as an IPv6 router (IPv6 forwarding table, router advertisement, ...)
 - Without it, the router can only be an IPv6 host

- Under the interface context, configure the address:
 - `ipv6 address <ipv6addr>[/<prefix-length>] [link-local]`
 - `ipv6 address <ipv6prefix>/<prefix-length> eui-64`
 - `ipv6 unnumbered <interface>`
- Example:
 - `ipv6 address 3ffe:b00:c18:1::1/64`

- As soon as an address is configured on an interface:
 - The link-local address is configured
 - Router advertisements are sent

- By default, router advertisements are automatically sent, using default parameters:
 - Prefix based on interface prefix
 - Lifetimes set to infinite
- Router advertisements can be adjusted from their default values using the `ipv6 nd` commands on the interface
- `ipv6 nd prefix-advertisement <routing-prefix>/<length> <valid-lifetime> <preferred-lifetime> [onlink | autoconfig]`
 - Useful to specify finite lifetimes
 - Don't forget the optional keywords "onlink autoconfig" for most cases

Example



```
ipv6 unicast-routing
interface Ethernet0
    ipv6 address 3ffe:b00:c18:1::/64 eui-64
    ipv6 nd prefix-advertisement 3ffe:b00:c18:1::/64
        43200 43200 onlink autoconfig
```

- Static routes:
 - `ipv6 route <prefix>/<length> <interface>`
- RIP
- BGP

- `ipv6 router rip <tag>`
 - Under global context
 - Starts a RIP process
 - `<tag>` is a table name
- `ipv6 rip <tag> enable`
 - Under interface context
 - Enables RIP on that interface
- `ipv6 rip <tag> default-information originate`
 - Under interface context
 - Router announces a default route
- `redistribute static|bgp|rip`
 - As in IPv4 for redistribution of routing entries between routing processes

RIP Example



```
ipv6 unicast-routing
ipv6 router rip T0
    redistribute static
interface Ethernet0
    ipv6 address 3ffe:b00:c18:1::/64 eui-64
    ipv6 rip T0 enable
```

- Use a new BGP address family for IPv6:
 - `address-family ipv6`
- If configuring IPv6 only router, add:
 - `no bgp default ipv4-unicast`
 - `bgp router-id <router-id>`
- Neighbor commands with IPv6 addresses
- Prefix filtering:
 - `ipv6 prefix-list`
- Route maps for attribute modification available

BGP Example



```
ipv6 unicast-routing
interface Ethernet0
  ipv6 address 3FFE:B00:C18:2:1::F/64
router bgp 65001
  no bgp default ipv4-unicast
  neighbor 3FFE:B00:C18:2:1::1 remote-as 65002
  address-family ipv6
    neighbor 3FFE:B00:C18:2:1::1 activate
    neighbor 3FFE:B00:C18:2:1::1 prefix-list bgp65002in in
    neighbor 3FFE:B00:C18:2:1::1 prefix-list bgp65002out out
  exit-address-family
```

- Similar to IPv4 tunnels, but:
 - Use tunnel mode ipv6ip
- Example:

```
interface Tunnel0
  no ip address
  ipv6 address 3ffe:b00:c18:1::3/64
  tunnel source 192.168.99.1
  tunnel destination 192.168.30.1
  tunnel mode ipv6ip
```

- Uses the tunnel interface
 - Do not specify the IPv4 address of the destination
 - Specify 6to4 in the tunnel mode
- Example:

```
interface Tunnel10
  no ip address
  ipv6 unnumbered Ethernet0
  tunnel source 192.168.99.1
  tunnel mode ipv6ip 6to4
```

- Ping, traceroute
- telnet
- debug ipv6 packet
- show ipv6 neighbors
 - Neighbors cache (arp cache)
- debug ipv6 icmp
- debug ipv6 nd
 - Neighbor discovery

- Cisco IOS supports main features of IPv6 and will continue to add
- Interface, Routing, Tunnel and Troubleshooting commands are similar to their IPv4 counterpart, but they have new features

References



- Cisco roadmap: <http://www.cisco.com/ipv6>
- Cisco documentation: <http://www.cisco.com>