The IPv6 Protocol & IPv6 Standards

ISP Workshops

Last updated 29th July 2019

Acknowledgements

- This material originated from the Cisco ISP/IXP Workshop Programme developed by Philip Smith & Barry Greene

- Use of these materials is encouraged as long as the source is fully acknowledged and this notice remains in place

- Bug fixes and improvements are welcomed
  - Please email *workshop (at) bgp4all.com*

Philip Smith

IPv6

- December 1995
  - First Specification published as Proposed Standard in RFC1883
- December 1998
  - Updated Specification published as Draft Standard in RFC2460
  - Virtually all implementations today adhere to RFC2460
- July 2017
  - RFC8200 declares IPv6 as Internet Standard, replacing RFC2460

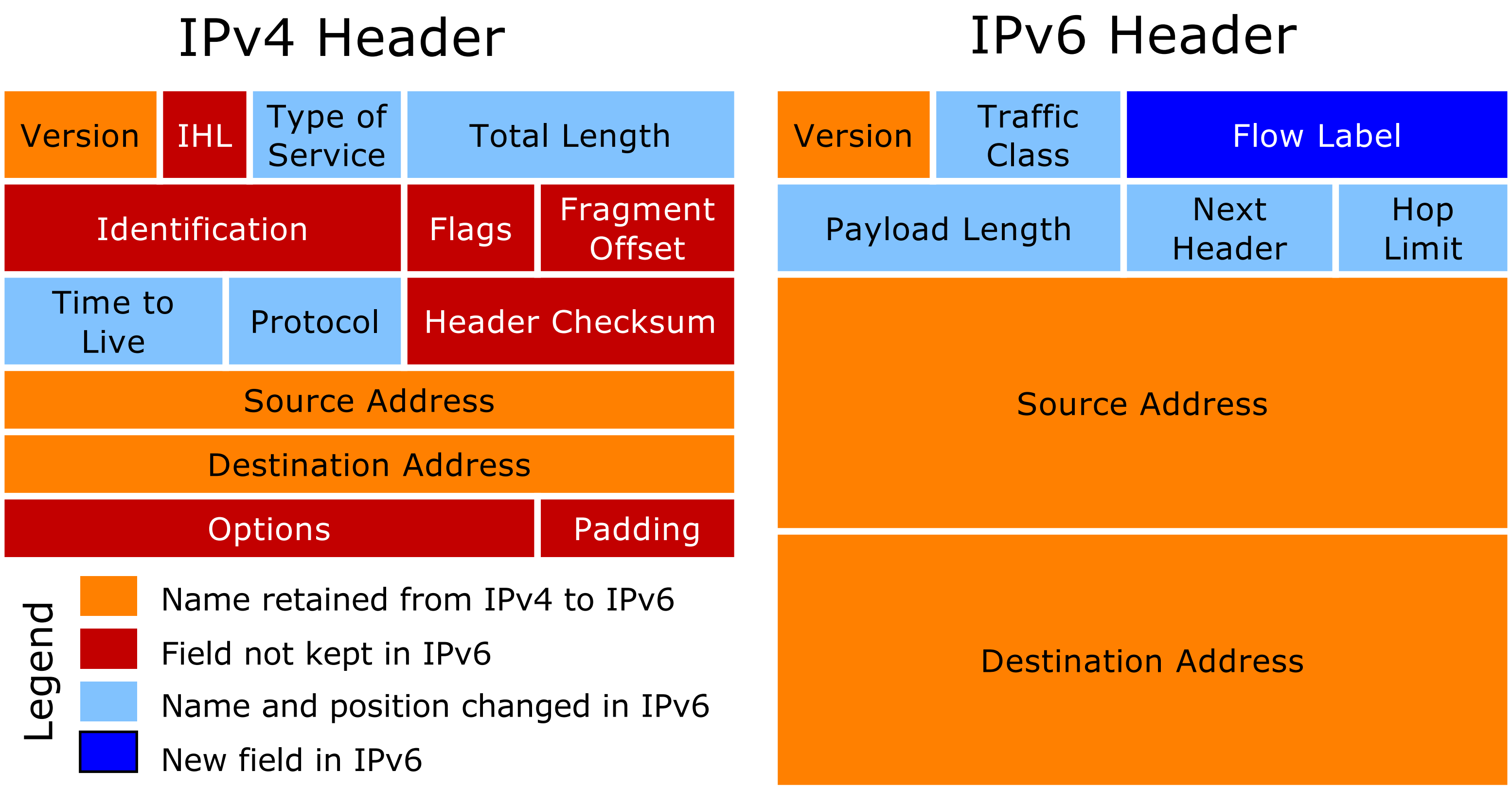
So what has really changed?

- IPv6 does not interoperate with IPv4
  - Separate protocol working independently of IPv4
  - Deliberate design intention
- Expanded address space
  - Address length quadrupled to 16 bytes
- Simplified header to remove unused or unnecessary fields
  - Fixed length headers to “make it easier for chip designers and software engineers”

What else has changed?

- Header Format Simplification
  - Fixed length, optional headers are daisy-chained
  - IPv6 header is twice as long (40 bytes) as IPv4 header without options (20 bytes)
- No checksum at the IP network layer
- No hop-by-hop fragmentation
  - Path MTU discovery
- 64 bits aligned
- Authentication and Privacy Capabilities
  - IPsec is integrated
- No more broadcast

IPv4 and IPv6 Header Comparison



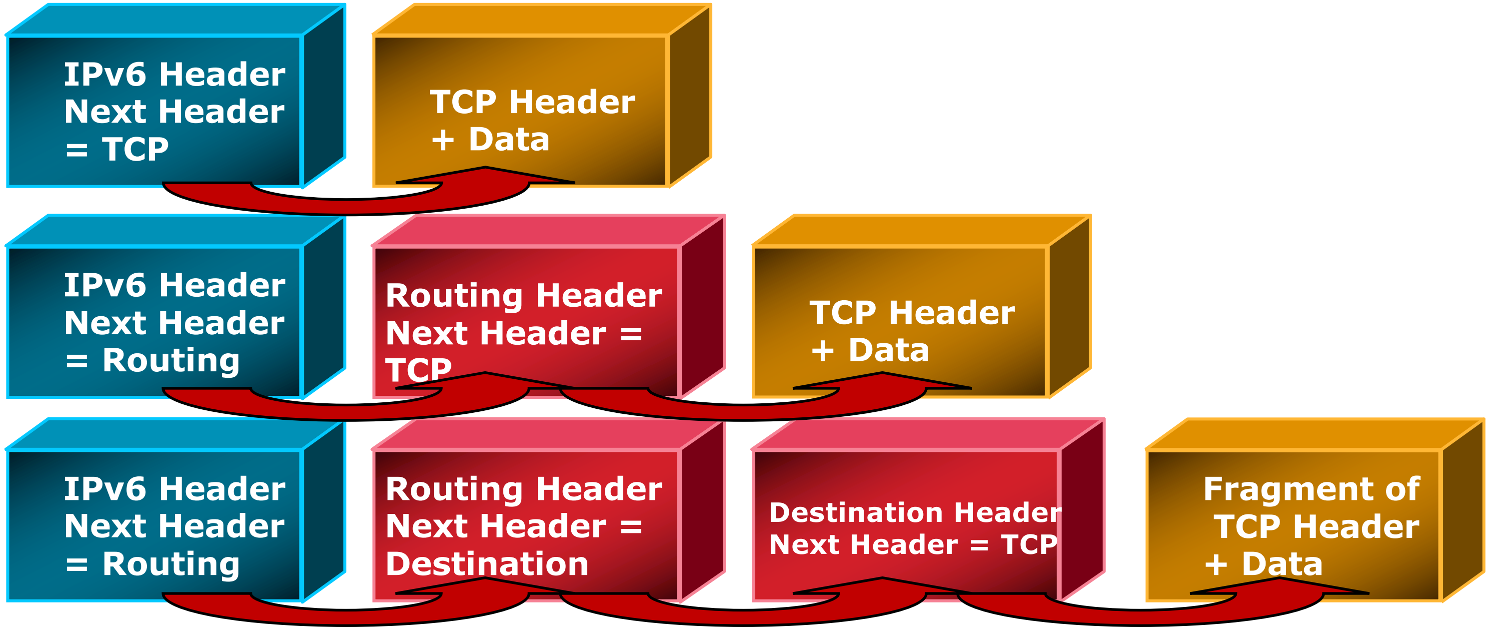
IPv6 Header

Field	Size	Commentary
Version	4-bit	Value set to “6”
Traffic Class	8-bit	Replaces IPv4 TOS field
Flow Label	20-bit	
Payload Length	16-bit	Size of rest of IPv6 packet after the Header
Next Header	8-bit	Indicates type of next header, replacing IPv4 “Protocol” field
Hop Limit	8-bit	Decreased by one at every IPv6 hop (IPv4 TTL counter)
Source Address	128-bit	
Destination Address	128-bit	

Header Format Simplification

- Fixed length
  - Optional headers are daisy-chained
- 64 bits aligned
- IPv6 header is twice as long (40 bytes) as IPv4 header without options (20 bytes)
- IPv4 contains 10 basic header fields
- IPv6 contains 6 basic header fields
  - No checksum at the IP network layer
  - No hop-by-hop fragmentation

Header Format – Extension Headers



- All optional fields go into extension headers
- These are daisy chained behind the main header
  - The last 'extension' header is usually the ICMP, TCP or UDP header
- Makes it simple to add new features in IPv6 protocol without major re-engineering of devices
- Number of extension headers is not fixed / limited

IPv6 Headers – Purpose

Header	Next Header Value	Purpose
IPv6 Header	-	Defines the IPv6 packet and payload
Hop-by-Hop Options Header	0	Actions for intermediate nodes along the path between source and destination
Destination Options Header	60	Actions for the destination node
Routing Header	43	Defines the node to be visited along the way to the destination
Fragment Header	44	Used by IPv6 to send a packet bigger than would fit in the path MTU to the destination
Authentication Header	51	Used for carrying authentication information (IPSEC AH)
ESP Header	50	Used for encrypting the payload (IPSEC ESP)
Null Header	59	No next header
Upper Layer Header	6/17/58	TCP/UDP/ICMP payload

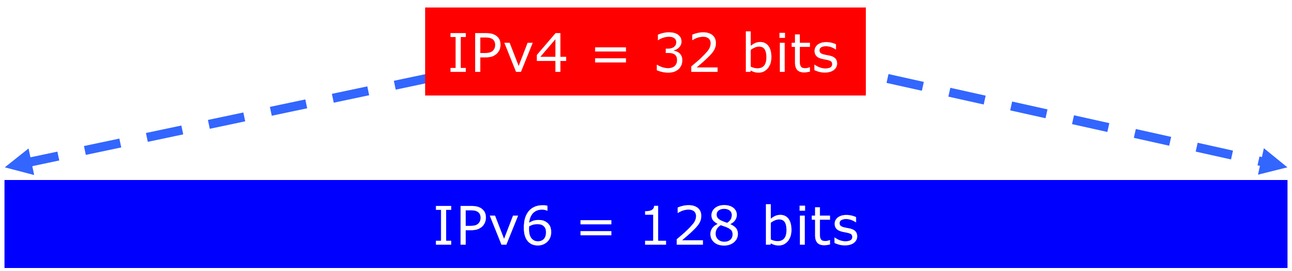
Ordering of IPv6 Headers

- RFC8200 recommends that IPv6 headers appear in this sequence:
  - IPv6 Header
  - Hop-by-Hop Options Header
  - Destination Options Header
  - Routing Header
  - Fragment Header
  - Authentication Header
  - Encapsulating Security Payload Header
  - Upper Layer Header
- Each extension header should only occur once, apart from Hop-by-Hop Options header which should appear no more than twice

Ordering of IPv6 Headers

- Order is important because:
  - Hop-by-hop header has to be processed by every intermediate node
  - Routing header needs to be processed by intermediate routers
  - At the destination, fragmentation has to be processed before other headers
- This makes header processing easier to implement in hardware

Larger Address Space



- IPv4
  32 bits
  = 4,294,967,296 possible addressable devices
- IPv6
  128 bits: 4 times the size in bits
  = 3.4 x 10^{38} possible addressable devices
  = 340,282,366,920,938,463,463,374,607,431,768,211,456
  = 4.5 x 10^{28} addresses per person on the planet

How was the IPv6 Address Size Chosen?

- Some wanted fixed-length, 64-bit addresses
  - Easily good for 10^{12} sites, 10^{15} nodes, at .0001 allocation efficiency
    - (3 orders of magnitude more than IPv6 requirement)
  - Minimizes growth of per-packet header overhead
  - Efficient for software processing
- Some wanted variable-length, up to 160 bits
  - Compatible with OSI NSAP addressing plans
  - Big enough for auto-configuration using IEEE 802 addresses
  - Could start with addresses shorter than 64 bits & grow later
- Settled on fixed-length, 128-bit addresses

IPv6 Address Representation (1)

- 16 bit fields in case insensitive colon hexadecimal representation
  - 2031:0000:130F:0000:0000:09C0:876A:130B
- Leading zeros in a field are optional:
  - 2031:0:130F:0:0:9C0:876A:130B
- Successive fields of 0 represented as ::, but only once in an address:
  - 2031:0:130F::9C0:876A:130B is ok
  - 2031::130F::9C0:876A:130B is **NOT** ok
  - 0:0:0:0:0:0:0:1 → ::1 (loopback address)
  - 0:0:0:0:0:0:0:0 → :: (unspecified address)

IPv6 Address Representation (2)

- :: positioning recommendations – RFC5952
  - The largest set of :0: be replaced with :: for consistency
    - 2001:DB8:0:2F:0:0:0:5 becomes 2001:DB8:0:2F::5 rather than 2001:DB8::2F:0:0:0:5
  - The first set of :0: be replaced with :: in the case there are two sets of :0:
    - 2001:db8:0:0:1:0:0:1 becomes 2001:db8::1:0:0:1 instead of 2001:db8:0:0:1::1
- IPv4-compatible (deprecated in RFC4291)
  - 0:0:0:0:0:0:192.168.30.1
  - = ::192.168.30.1
  - = ::C0A8:1E01
- IPv4-mapped (RFC4038: application considerations for IPv6)
  - 0:0:0:0:0:FFFF:192.168.30.1
  - = ::FFFF:192.168.30.1

IPv6 Address Representation (3)

- In a URL, it is enclosed in brackets (RFC3986)
  - http://[2001:DB8:4F3A::206:AE14]:8080/index.html
  - Cumbersome for users, mostly for diagnostic purposes
  - Use fully qualified domain names (FQDN)
  - ⇒ The DNS has to work!!
- Prefix Representation
  - Representation of prefix is just like IPv4 CIDR
  - The prefix length (subnet size) appears after the "/"
  - IPv4 address:
    - 198.10.0.0/16
  - IPv6 address:
    - 2001:DB8:1200::/40

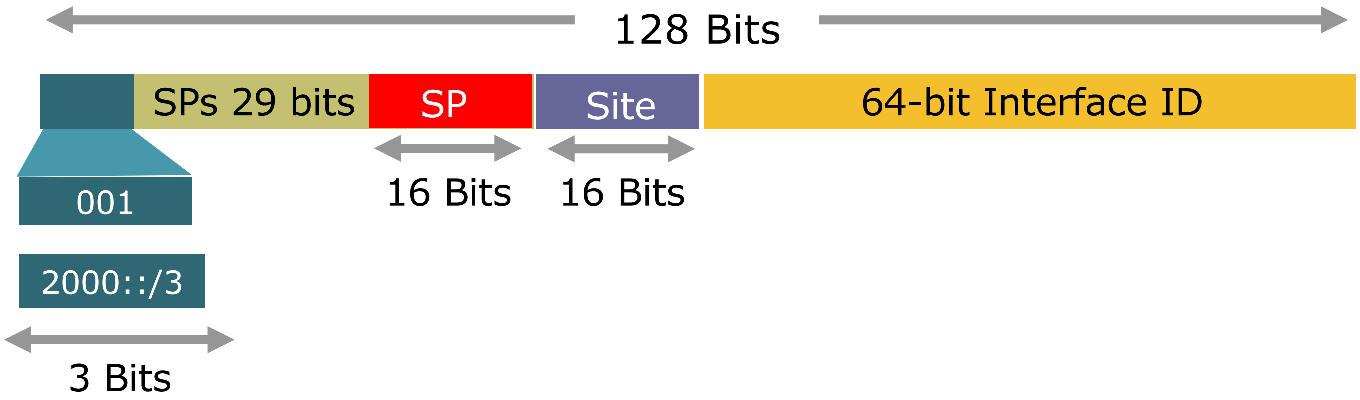
IPv6 Addressing

- IPv6 Addressing rules are covered by multiple RFCs
  - Architecture defined by RFC4291
- Address Types are :
  - Unicast : One to One (Global, Unique Local, Link local)
  - Anycast : One to Nearest (Allocated from Unicast)
  - Multicast : One to Many
- A single interface may be assigned multiple IPv6 addresses of any type (unicast, anycast, multicast)
  - No Broadcast Address → Use Multicast

IPv6 Addressing

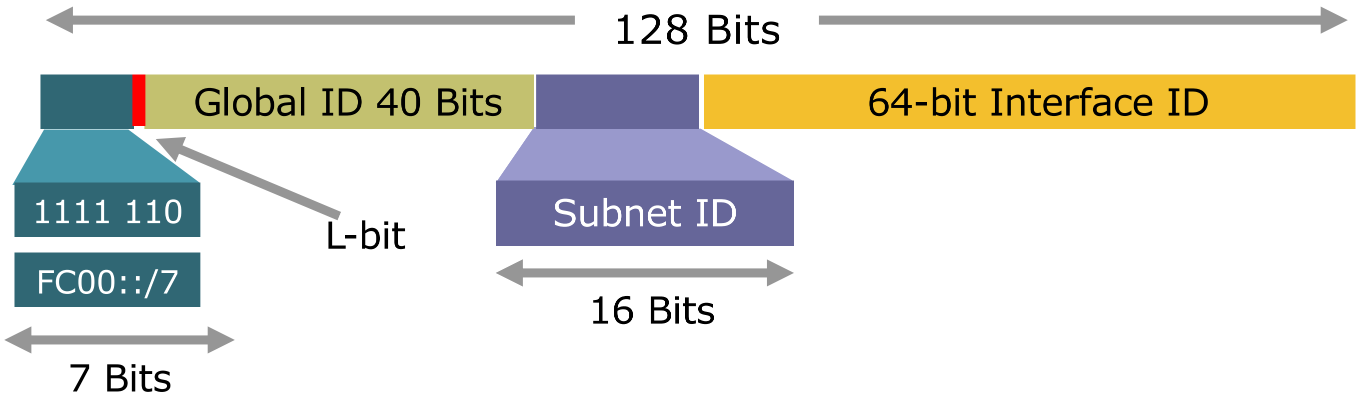
Type	Binary	Hex
Unspecified	000…0	::/128
Loopback	000…1	::1/128
Global Unicast Address	0010	2000::/3
Unique Local Unicast Address	1111 1100 1111 1101	FC00::/7
Link Local Unicast Address	1111 1110 10	FE80::/10
Multicast Address	1111 1111	FF00::/8

Global Unicast Addresses



- Address block delegated by IETF to IANA
- For distribution to the RIRs and on to the users of the public Internet
- Global Unicast Address block is 2000::/3
  - This is 1/8th of the entire available IPv6 address space

Unique-Local Addresses

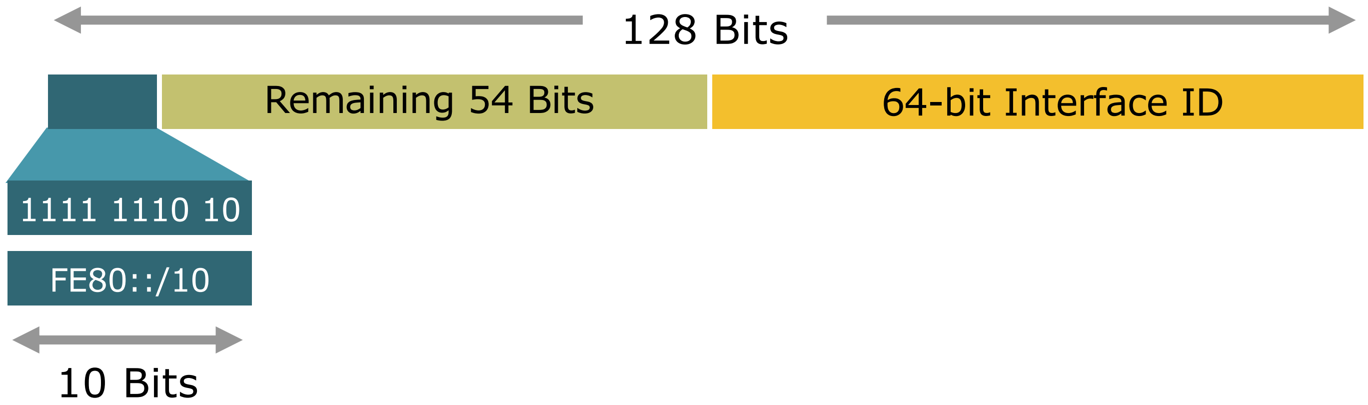


- Unique-Local Addresses (ULAs) are NOT routable on the Internet
  - L-bit set to 1 – which means the address is locally assigned
- ULAs are used for:
  - Isolated networks
  - Local communications & inter-site VPNs
  - (see https://datatracker.ietf.org/doc/draft-ietf-v6ops-ula-usage-considerations/ – now expired)

Unique-Local – Typical Scenarios

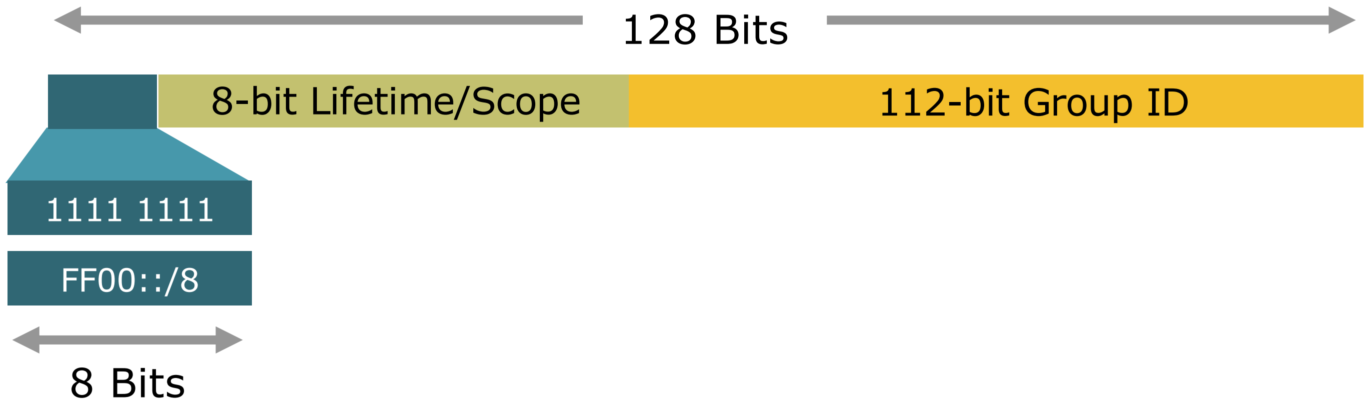
- Isolated IPv6 networks:
  - Never need public Internet connectivity
  - Don’t need assignment from RIR or ISP
- Local devices such as printers, telephones, etc
  - Connected to networks using Public Internet
  - But the devices themselves do not communicate outside the local network
- Site Network Management systems connectivity
- Infrastructure addressing
  - Using dual Global and Unique-Local addressing
- Public networks experimenting with NPTv6 (RFC6296)
  - One to one IPv6 to IPv6 address mapping

Link-Local Addresses



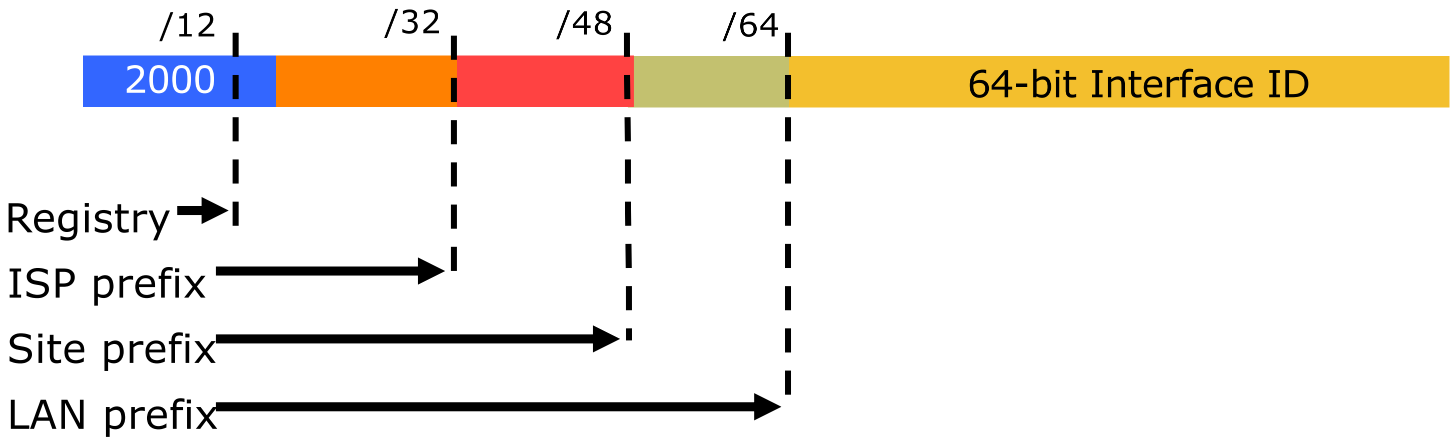
- Link-Local Addresses Used For:
  - Communication between two IPv6 device (like ARP but at Layer 3)
  - Next-Hop calculation in Routing Protocols
- Automatically assigned by device as soon as IPv6 is enabled on an interface
  - **Mandatory Address**
- Only Link Specific scope
- Remaining 54 bits could be Zero or any manual configured value

Multicast Addresses



- Multicast Addresses Used For:
  - One to many communication
- 2nd octet reserved for Lifetime and Scope
- Remainder of address represents the Group ID
- (Substantially larger range than for IPv4 which only had 224.0.0.0/4 for Multicast)

Global Unicast IPv6 Address Allocation



- The allocation process is:
  - The IANA is allocating out of 2000::/3 for initial IPv6 unicast use
  - Each registry gets a /12 prefix from the IANA
  - Registry allocates a /32 prefix (or larger) to an IPv6 Network Operator
  - Network Operators allocate a /48 prefix to each end customer

IPv6 Addressing Scope

- 64 bits reserved for the interface ID
  - Possibility of 2^{64} hosts on one network LAN
  - In theory 18,446,744,073,709,551,616 hosts
  - Arrangement to accommodate MAC addresses within the IPv6 address
- 16 bits reserved for the end site
  - Possibility of 2^{16} networks at each end-site
  - 65536 subnets equivalent to a /12 in IPv4 (assuming a /28 or 16 hosts per IPv4 subnet)

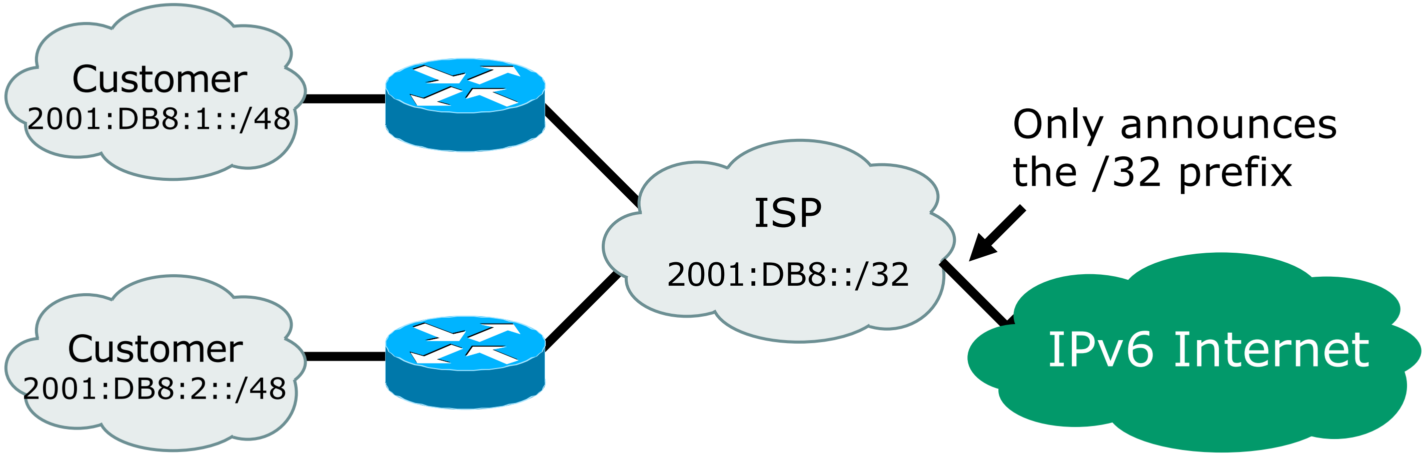
IPv6 Addressing Scope

- 16 bits reserved for each service provider
  - Possibility of 2^{16} end-sites per service provider
  - 65536 possible customers: equivalent to each service provider receiving a /8 in IPv4 (assuming a /24 address block per customer)
- 29 bits reserved for all service providers
  - Possibility of 2^{29} service providers
  - i.e. 536,870,912 discrete service provider networks
    - Although some service providers already are justifying more than a /32

How to get an IPv6 Address?

- IPv6 address space is allocated by the 5 RIRs:
  - AfriNIC, APNIC, ARIN, LACNIC, RIPE NCC
  - Network Operators get address space from the RIRs
  - End Users get IPv6 address space from their upstream provider
- In the past, there were also:
  - 6to4 tunnels using 2002::/16
    - Intended to give isolated IPv6 nodes access to the IPv6 Internet
    - Obsoleted in May 2015 (BCP196) because it was very unreliable and totally insecure
  - 6Bone using 3FFE::/16
    - The experimental IPv6 network launched in the mid 1990s
    - Was retired on 6th June 2006 (RFC3701)

Aggregation hopes

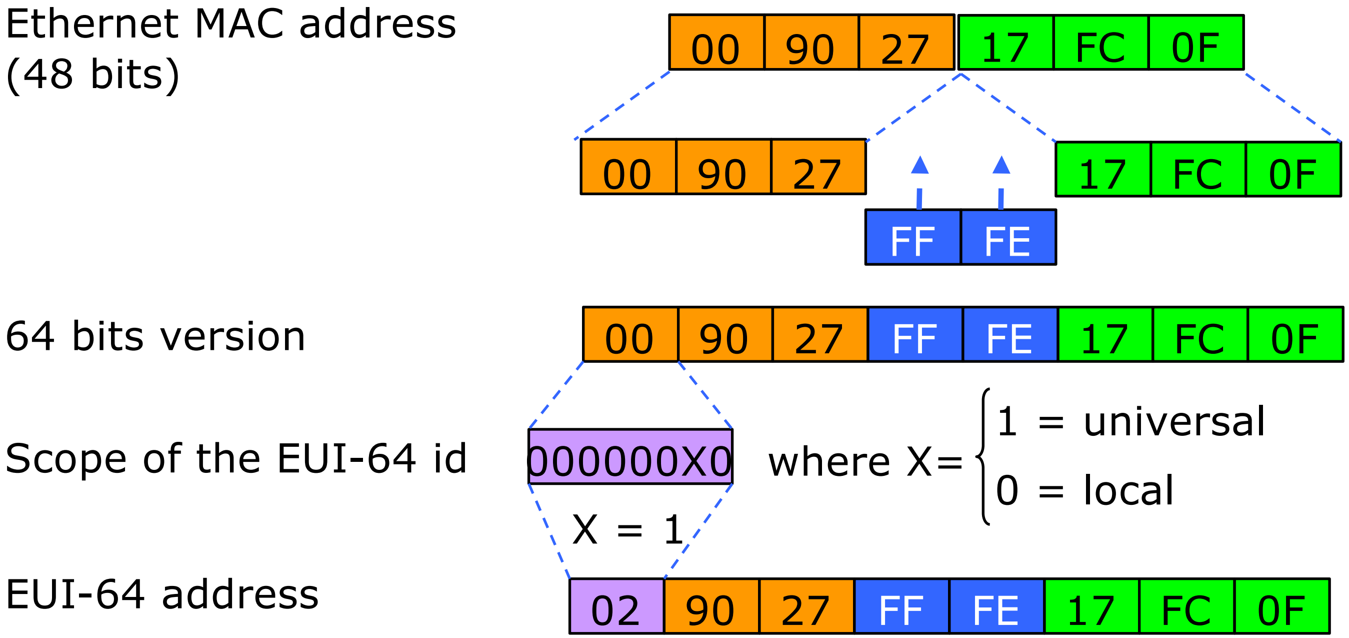


- Larger address space enables aggregation of prefixes announced in the global routing table
- Idea was to allow efficient and scalable routing
- But current Internet multihoming solution breaks this model

Interface IDs

- Lowest order 64-bit field of unicast address may be assigned in several different ways:
  - Auto-configured from a 64-bit EUI-64, or expanded from a 48-bit MAC address (e.g., Ethernet address)
  - Auto-generated pseudo-random number (to address privacy concerns)
  - Semantically Opaque Interface Identifier (RFC7217)
  - Assigned via DHCP
  - Manually configured

EUI-64

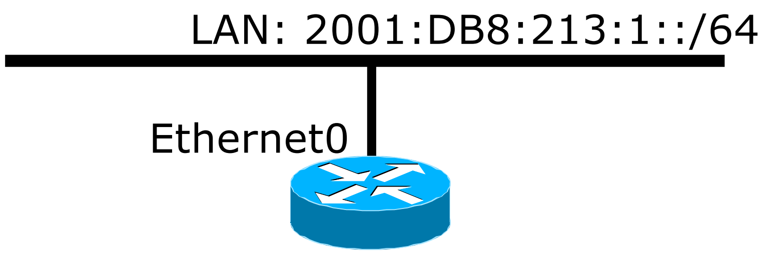


- EUI-64 address is formed by inserting FFFE between the company-id and the manufacturer extension, and setting the "u" bit to indicate scope
  - Global scope: for IEEE 48-bit MAC
  - Local scope: when no IEEE 48-bit MAC is available (eg serials, tunnels)

EUI-64

- Device MAC address is used to create:
  - Final 64 bits of global unicast address e.g.
    - 2001:DB8:0:1:290:27FF:FE17:FC0F
  - Final 64 bits of link local address e.g.
    - FE80::290:27FF:FE17:FC0F
  - Final 24 bits of solicited node multicast address e.g.
    - FF02::1:FF17:FC0F
- Note that both global unicast and link local addresses can also be configured manually

EUI-64 on Cisco IOS

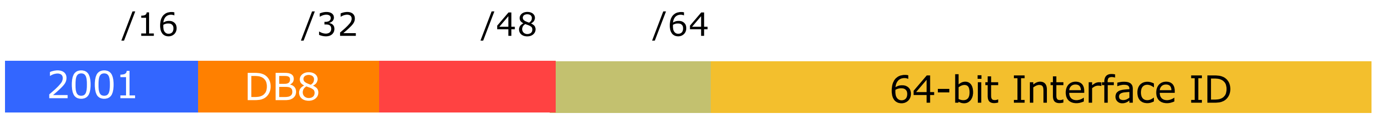



```
interface Ethernet0
 ipv6 address 2001:DB8:213:1::/64 eui-64
```

MAC address: 0060.3e47.1530

```
router# show ipv6 interface Ethernet0
Ethernet0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::260:3EFF:FE47:1530
Global unicast address(es):
    2001:DB8:213:1:260:3EFF:FE47:1530, subnet is 2001:DB8:213:1::/64
  Joined group address(es):
    FF02::1:FF47:1530
    FF02::1
    FF02::2
  MTU is 1500 bytes
```

IPv6 Address Privacy (RFC4941)



- Temporary addresses for IPv6 host client application, e.g. Web browser
- Intended to inhibit device/user tracking but is also a potential issue
  - More difficult to scan all IP addresses on a subnet
  - But port scan is identical when an address is known
- Random 64-bit interface ID, run DAD before using it
- Rate of change based on local policy
- Implemented on Microsoft Windows Vista onwards and on Apple MacOS 10.7 onwards
  - Can be activated on FreeBSD/Linux with a system call

Semantically Opaque Interface Identifier

- Described in RFC7217
  - Helps resolve privacy concerns with EUI-64
    - MAC address is unchanging meaning the same device can be tracked across multiple networks
    - Location 1: `2001:DB8:3:1A4:` `260:3EFF:FE47:1530`
    - Location 2: `2001:DB8:213:1:` `260:3EFF:FE47:1530`

EUI-64

  - Helps resolve network management concerns with RFC4941 privacy address
    - Including complexity of event logging, troubleshooting, QoS, enforcement of access controls, etc

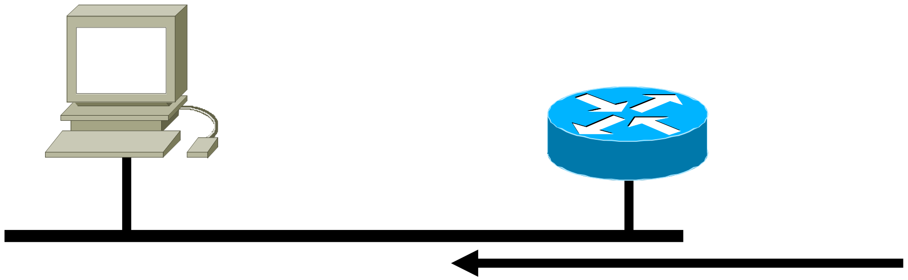
Semantically Opaque Interface Identifier

- Final 64-bits of Global Unicast IPv6 Address is generated by algorithm using a cryptographic hash of:
  - Prefix to be used (learned by router advertisements)
  - Identifier associated with the interface
  - Network specific data associated with interface (eg 802.11 SSID)
  - Secret key – a pseudo random number
- And taking the necessary number of bits to make up the rest of the IPv6 address
- Implemented on most modern operating systems like Linux, macOS, etc

Host IPv6 Addressing Options

- Stateless (RFC4862)
  - SLAAC – Stateless Address AutoConfiguration
- Stateful
  - DHCPv6 (RFC3315) – required by most enterprises
  - DHCPv6-PD – **new**: Prefix Delegation
    - Allows Network Operators to distribute subnets to End-sites
  - Manual – like IPv4 before DHCP was developed
    - Useful for servers and router infrastructure
    - Does not scale for typical end user devices

Stateless Address Auto-configuration



- Device auto-configures link-local address & checks for duplicates
- Device sends router solicitation (RS) message
- Router responds with router advertisement (RA)
  - This includes prefix and default route
  - RFC8106 adds DNS server option
- Device configures its IPv6 address by concatenating prefix received with its generated “stable” address

Stateful DHCPv6

- Stateful DHCPv6 is similar to DHCP in IPv4
  - Described in RFC3315
  - DHCP server responds to device request with:
    - IPv6 address information (and its lifetime)
    - DNS resolver
    - And any other options defined by the network operator
  - Desired by most enterprises to ensure manageability of the network and provides an association between the device and the IPv6 address it uses
  - SLAAC is disabled on the subnet where DHCP is being provided

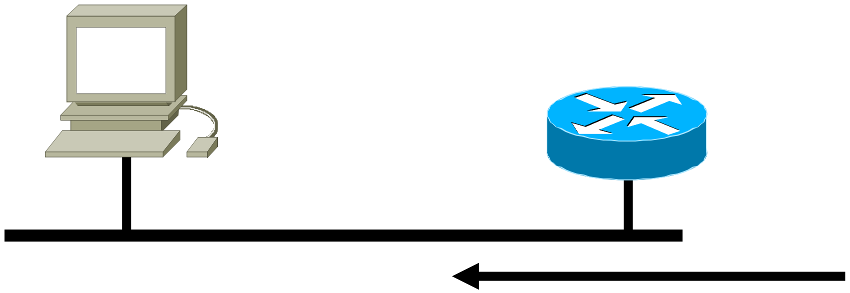
Stateless DHCPv6

- Stateless DHCPv6 works along with SLAAC
  - Described in RFC3736
  - Device uses SLAAC to receive:
    - IPv6 address information (and its lifetime)
  - DHCP server provides:
    - DNS resolver
    - And any other options defined by the network operator
  - Used in situations where SLAAC cannot provide DNS resolver information (lack of RFC8106 support)

IPv6 Renumbering

- Renumbering Hosts
  - Stateless:
    - Hosts renumbering is done by modifying the RA to announce the old prefix with a short lifetime and the new prefix
  - Stateful:
    - DHCPv6 uses same process as DHCPv4
- Renumbering Routers
  - Router renumbering protocol was developed (RFC2894) to allow domain-interior routers to learn of prefix introduction / withdrawal
  - **No known implementation!**

SLAAC: Renumbering

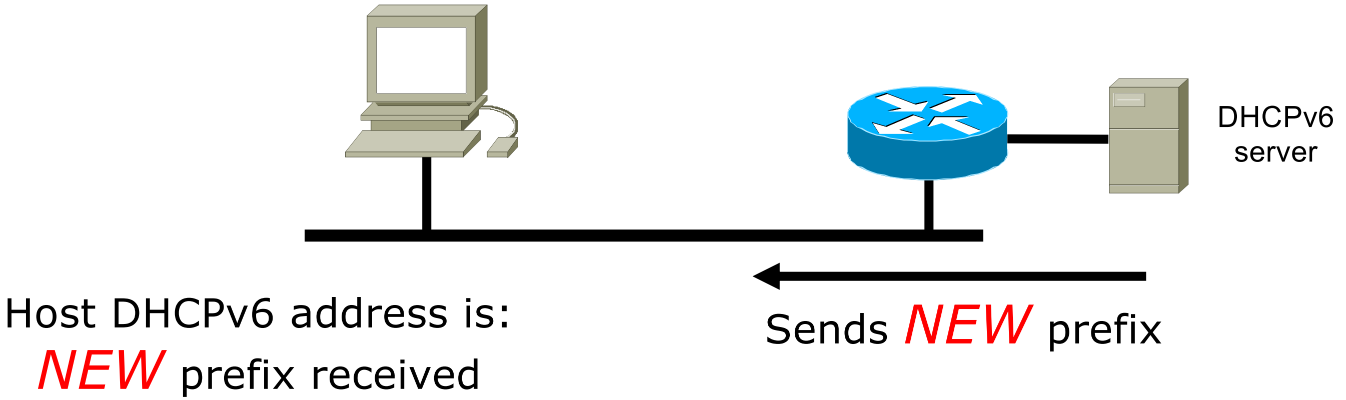


Host auto-configured address is:
NEW prefix received + SAME final 64-bits

Sends *NEW* network-type information (prefix, default route, …)

- Router sends router advertisement (RA)
  - This includes the new prefix and default route (and remaining lifetime of the old address)
- Device configures a new IPv6 address by concatenating prefix received with its generated “stable” address
  - Retains old address but attaches lifetime to it

DHCPv6: Renumbering



- Router has old and new subnet configured on it
- When the existing IPv6 address preferred lifetime expires, the host sends a request to renew the IPv6 address
- The DHCPv6 server sends new IPv6 address information
- Device configures the new IPv6 address on the interface
  - Uses new address for new connections, removes old address when valid lifetime expires

Multicast use

- Broadcasts in IPv4
  - Interrupts all devices on the LAN even if the intent of the request was for a subset
  - Can completely swamp the network ("broadcast storm")
- Broadcasts in IPv6
  - Are not used and replaced by multicast
- Multicast
  - Enables the efficient use of the network
  - Multicast address range is much larger

IPv6 Multicast Address

- IP multicast address has a prefix FF00::/8
- The second octet defines the lifetime and scope of the multicast address.

8-bit	4-bit	4-bit	112-bit
1111 1111	Lifetime	Scope	Group-ID

Lifetime	
0	If Permanent
1	If Temporary

Scope	
1	Node
2	Link
5	Site
8	Organisation
E	Global

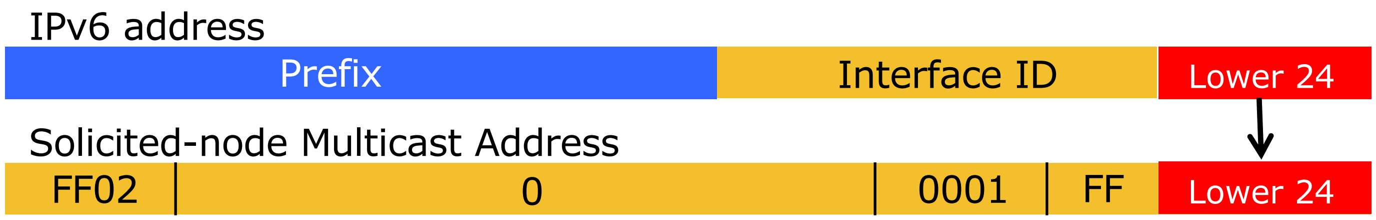
IPv6 Multicast Address Examples

- RIPng
  - The multicast address AllRIPRouters is FF02::9
    - Note that 02 means that this is a permanent address and has link scope
- OSPFv3
  - The multicast address AllSPFRouters is FF02::5
  - The multicast address AllDRouters is FF02::6
- EIGRP
  - The multicast address AllEIGRPRouters is FF02::A

Solicited-Node Multicast

- Solicited-Node Multicast is used for Duplicate Address Detection
  - Part of the Neighbour Discovery process
  - Replaces ARP
  - Duplicate IPv6 Addresses are rare, but still have to be tested for
- For each unicast and anycast address configured there is a corresponding solicited-node multicast address
  - This address is only significant for the local link

Solicited-Node Multicast



- Solicited-node multicast address consists of FF02:0:0:0:0:1:FF::/104 prefix joined with the lower 24 bits from the unicast or anycast IPv6 address

Solicited-Node Multicast

```
R1#sh ipv6 int e0
Ethernet0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::200:CFF:FE3A:8B18
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF3A:8B18
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
R1#
```

Solicited-Node Multicast Address

IPv6 Anycast

- An IPv6 anycast address is an identifier for a set of interfaces (typically belonging to different devices/nodes)
  - A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the “nearest” one, according to the routing protocol’s measure of distance).
  - Anycast addresses are allocated from the Global Unicast Address pool
  - If the device uses the anycast address on more than one interface, it must indicate that the address is anycast

```
interface FastEthernet0/0
 description Network Services
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:0:1::1/64
 ipv6 address 2001:DB8:F:1::1/64 anycast
```

The anycast address

  - RFC4291 describes IPv6 Anycast in more detail

Anycast on the Internet

- Anycast usage today:
  - Described in RFC4786 / BCP126
  - A global address is assigned to all nodes which need to respond to a service being offered
    - This address is routed as part of its parent address block
  - The responding node is the one which is closest to the requesting node according to the routing protocol
    - Each anycast node looks identical to the other
  - Applicable within an ASN, or globally across the Internet

Anycast on the Internet

- Typical examples today include:
  - Global DNS resolvers

Google	8.8.8.8	2001:4860:4860::8888
Google	8.8.4.4	2001:4860:4860::8844
Quad9	9.9.9.9	2620:FE::FE

  - Root DNS and ccTLD/gTLD nameservers

F-root	192.5.5.241	2001:500:2F::F
I-root	192.36.148.17	2001:7fe::53
.com	192.5.6.30	2001:503:a83e::2:30
.se	194.146.106.22	2001:67c:1010:5::53

  - SMTP relays and DNS resolvers within ISP autonomous systems

MTU Issues

- Minimum link MTU for IPv6 is 1280 octets (versus 68 octets for IPv4)
  - $\Rightarrow$ on links with MTU < 1280, link-specific fragmentation and reassembly must be used
- Implementations are expected to perform path MTU discovery to send packets bigger than 1280
- Minimal implementation can omit PMTU discovery as long as all packets kept ≤ 1280 octets
- A Hop-by-Hop Option supports transmission of "jumbograms" with up to 2^{32} octets of payload
  - Although RFC8504 / BCP220 (IPv6 Node Requirements) removes jumbograms from the Hop-by-Hop Option Header requirements for IPv6 Nodes

IPv6 Neighbour Discovery

- Protocol defines mechanisms for the following problems:
  - Router discovery
  - Prefix discovery
  - Parameter discovery
  - Address autoconfiguration
  - Address resolution
  - Next-hop determination
  - Neighbour unreachability detection
  - Duplicate address detection
  - Redirects

IPv6 Neighbour Discovery

- Defined in RFC4861
- Protocol built on top of ICMPv6 (RFC 4443)
  - Combination of IPv4 protocols (ARP, ICMP, IGMP,…)
- Fully dynamic, interactive between Hosts & Routers
- Defines 5 ICMPv6 packet types:
  - Router Solicitation
  - Router Advertisement
  - Neighbour Solicitation
  - Neighbour Advertisement
  - Redirect

IPv6 and DNS

- Hostname to IP address:

IPv4	www.abc.test.	A	192.168.30.1
IPv6	www.abc.test.	AAAA	2001:DB8:C18:1::2

Example Forward Zone File

```
@  IN  SOA  ns.example.  admin.example.  (2018040300 3600 1800 604800 86400)
                NS      ns1.example.
;;; Servers
www                     A       192.168.1.1
                        AAAA    2001:DB8:1::1
ns                      A       192.168.1.2
                        AAAA    2001:DB8:1::2
mail                    A       192.168.1.3
                        AAAA    2001:DB8:1::3
;;; Routers
cr.city1                A       192.168.0.1
                        AAAA    2001:DB8::1
cr.city2                A       192.168.0.2
                        AAAA    2001:DB8::2
cr.city3                A       192.168.0.3
                        AAAA    2001:DB8::3
;;; P2P Links
xe-2-0-0.cr.city1       A       192.168.0.33
                        AAAA    2001:DB8:0:1::0
xe-2-1-0.cr.city2       A       192.168.0.34
                        AAAA    2001:DB8:0:1::1
xe-1-2-0.cr.city2       A       192.168.0.37
                        AAAA    2001:DB8:0:2::0
xe-2-1-0.cr.city3       A       192.168.0.38
                        AAAA    2001:DB8:0:2::1
```

NB: Only create AAAA entry once the service has been configured to respond to IPv6

IPv6 and DNS

- IP address to Hostname:

IPv4	1.30.168.192.in-addr.arpa.	PTR	www.abc.test.
IPv6	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.C.0.8.B.D.0.1.0.0.2.ip6.arpa	PTR	www.abc.test.

Example IPv6 Reverse Zone File

```
@  IN  SOA  ns.example.  admin.example.  (2018040300 3600 1800 604800 86400)
        NS      ns1.example.
;;; Servers
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1       PTR     www.example.
2       PTR     ns.example.
3       PTR     mail.example.
;;; Routers
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1       PTR     cr.city1.example.
2       PTR     cr.city2.example.
3       PTR     cr.city3.example.
;;; P2P Links
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
0       PTR     xe-2-0-0.cr.city1.example.
1       PTR     xe-2-1-0.cr.city2.example.
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
0       PTR     xe-2-0-0.cr.city2.example.
1       PTR     xe-2-1-0.cr.city3.example.
```

Example BIND9 configuration

- And to join the previous examples together, this might be what the configuration for BIND looks like:

```
zone ″example" IN {
        type master;
        file "zones/db.example.master";
      };

zone ″8.b.d.0.1.0.0.2.ip6.arpa" IN {
        type master;
        file "zones/db.2001.db8.master";
      };

zone ″0.168.192.in-addr.arpa" IN {
        type master;
        file "zones/db.0.168.192.master";
      };

zone ″1.168.192.in-addr.arpa" IN {
        type master;
        file "zones/db.1.168.192.master";
      };
```

- Critical point:
  - Never create any DNS entry unless the device is able to provide the claimed service

IPv6 Technology Scope

IP Service	IPv4 Solution	IPv6 Solution
Addressing Range	32-bit, Network Address Translation	128-bit, Multiple Scopes
Autoconfiguration	DHCP	DHCP, Serverless, Reconfiguration
Security	IPsec	IPsec works End-to-End
Quality of Service	Differentiated Service, Integrated Service	Differentiated Service, Integrated Service
Multicast	IGMP, PIM, Multicast BGP	MLD, PIM, Multicast BGP, Scope Identifier

What does IPv6 do for:

- Security
  - Everything that IPv4 already supports
  - IPSec runs on both
- QoS
  - Everything that IPv4 already supports
  - Differentiated and Integrated Services run on both
  - So far, the Flow label has not been used

IPv6 Security

- IPSec standards apply to both IPv4 and IPv6
- All implementations required to support authentication and encryption headers (“IPSec”)
- Authentication is separate from encryption for use in situations where encryption is prohibited or prohibitively expensive
  - AH = Authentication Header
  - ESP = Encrypted Security Payload
- Key distribution protocols are not yet defined (independent of IP v4/v6)
- Support for manual key configuration required

IP Quality of Service Reminder

- Two basic approaches developed by IETF:
  - “Integrated Service” (int-serv)
    - Fine-grain (per-flow), quantitative promises (e.g., x bits per second), uses RSVP signalling
  - “Differentiated Service” (diff-serv)
    - Coarse-grain (per-class), qualitative promises (e.g., higher priority), no explicit signalling
  - Signalled diff-serv (RFC2998)
    - Uses RSVP for signalling with course-grained qualitative aggregate markings
    - Allows for policy control without requiring per-router state overhead

IPv6 Support for Int-Serv

- 20-bit Flow Label field to identify specific flows needing special QoS
  - Each source chooses its own Flow Label values; routers use Source Addr + Flow Label to identify distinct flows
  - Flow Label value of 0 used when no special QoS requested (the common case today)
- Originally standardised as RFC3697

IPv6 Flow Label

- Flow label has not been used since IPv6 standardised
  - Suggestions for use in recent years were incompatible with original specification (discussed in RFC6436)
- Specification updated in RFC6437
  - RFC6438 describes the use of the Flow Label for equal cost multi-path and link aggregation in Tunnels

IPv6 Support for Diff-Serv

- 8-bit Traffic Class field to identify specific classes of packets needing special QoS
  - Same as new definition of IPv4 Type-of-Service byte
  - May be initialized by source or by router enroute; may be rewritten by routers enroute
  - Traffic Class value of 0 used when no special QoS requested (the common case today)

IPv6 Status – Standardisation

- Core IPv6 Specifications are IETF Standards
  - Well tested & stable
  - Years of deployment experience
- 3GPP UMTS Rel 5 cellular wireless standards (2002) mandated IPv6
- Several key components on standards track…

Specification (STD86)	Neighbour Discovery (RFC4861)
ICMPv6 (STD89)	IPv6 Addresses (RFC4291 & 3587)
RIP (RFC2080)	BGP (RFC2545)
IGMPv6 (RFC2710)	OSPF (RFC5340)
Router Alert (RFC2711)	Jumbograms (RFC2675)
Autoconfiguration (RFC4862)	Radius (RFC3162)
DHCPv6 (RFC3315 & 4361)	Flow Label (RFC6436/7/8)
IPv6 Mobility (RFC6275)	Mobile IPv6 MIB (RFC4295)
GRE Tunnelling (RFC2473)	Unique Local IPv6 Addresses (RFC4193)
DAD for IPv6 (RFC4429)	Teredo (RFC4380)
ISIS for IPv6 (RFC5308)	VRRP (RFC5798)

IPv6 Status – Standardisation

- IPv6 available over:
  - PPP (RFC5072)
  - FDDI (RFC2467)
  - NBMA (RFC2491)
  - Frame Relay (RFC2590)
  - IEEE1394 (RFC3146)
  - Facebook (RFC5514)
  - LoWPAN (RFC8138)
  - Cellular Networks (RFC6459)
  - Ethernet (RFC2464)
  - Token Ring (RFC2470)
  - ATM (RFC2492)
  - ARCnet (RFC2497)
  - FibreChannel (RFC4338)
  - MS/TP (RFC8163)
  - DECT ULE (RFC8105)

Recent IPv6 Hot Topics

- IPv6 on Mobile Networks
  - "The end of NAT" and "NAT offload"
- IPv4 depletion debate
  - IANA IPv4 pool ran out on 3rd February 2011
    - http://www.potaroo.net/tools/ipv4/
- IPv6 Transition "assistance"
  - CGNAT, 6rd, NAT64, DS-Lite, 464XLAT…
- IPv6 Security
  - Security industry & experts taking much closer look
- Multihoming
  - Multihoming in IPv6 is the same as in IPv4

Conclusion

- Protocol is “ready to go”
- The core components have already seen more than 20 years global operational experience

The IPv6 Protocol & IPv6 Standards

ISP Workshops