The Internet Protocol – Next Generation
The Internet Protocol – version 6 (IPv6)

Diarmuid O'Briain

Table of Contents

Illustration Index

Abbreviations

464LAT	IPv4 to IPv6 to IPv4 Translation
6rd	IPv6 Rapid Deployment
AFI	Address Family Identifier
AfriNIC	African Network Information Centre
AFTR	Address Family Transition Router
API	Application Programmable Interface
ARP	Address Resolution Protocol
ASBR	Autonomous System Boundary Router
B4	Basic Bridging BroadBand
BGP	Border Gateway Protocol
CGN	Carrier Grade NAT
CIDR	Classless Inter-Domain Routing
CLAT	Customer-side transLATor
CPE	Customer Premises Equipment
DBD	Database Descriptor
DHCP	Dynamic Host Configuration Protocol
DHCPv6	DHCP version 6
DNS64	DNS from IPv6 to IPv4
DNS	Domain Name Server
DNSSL	DNS Search List Option
DUID	DHCP Unique Identifier
eBGP	external BGP
EIGRP	Enhanced Interior Gateway Routing Protocol
FFR	Free Range Routing
FIB	Forwarding Information Base
FQDN	Fully Qualified Domain Name
GRE	Generic Routing Encapsulation
GUA	Global Unicast Addresses
iBGP	internal BGP
ICMP	Internet Control Message Protocol
ICMPv6	ICMP version 6
IGMP	Internet Group Membership Protocol
IGMPv3	IGMP version 3
IHL	Internet Header Length
Interface ID	Interface IDentifier
IPSec	Internet Protocol Security
IPv6, IPng	Internet Protocol version 6
IS-IS	Intermediate System to Intermediate System
LIR	Local Internet Registry
LLA	Link-Local Address
LSA	Link State Advertisement
LSN	Large Scale NAT
lw4o6	Lightweight 4over6
lwAFTR	Lightweight AFTR
lwB4	Lightweight B4
MAC	Medium Access Control
MLD	Multicast Listener Discovery
MP-BGP	Multiprotocol BGP
MPLS	Multiprotocol Label Switching
MTU	Maximum Transmission Unit
NA	Neighbour Advertisement
NAPT	Network and Port Translation
NAT64	NAT from IPv6 to IPv4
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbour Discovery
ND	Neighbour Discovery
NDP	Neighbour Discovery Protocol
NIS	Network Information Service
NLRI	Network Layer Reachability Information
NMS	Network Management System

NS	Neighbour Solicitation
NSSA	Not-So-Stubby Area
NTP	Network Time Protocol
NUD	Neighbour Unreachability Detection
OSI	Open Systems Interconnection model
OSPF	Open Shortest Path First
OSPFv2	OSPF version 2
OSPFv3	OSPF version 3
P2MP	Point to Multipoint
P2P	Point to Point
PA	Provider Aggregatable
PCP	Port Control Protocol
PIM	Protocol Independent Multicast
PI	Provider Independent
PLAT	Provider-side transLATor
PMTUD	Path MTU Discovery
POP	Point of Presence
PSID	Port Set ID
PTP	Precision Time Protocol
RA	Router Advertisement
RDNSS	Recursive DNS Server
RIB	Routing Information Base
RIID	RP Interface Identifier
RIP	Routing Internet Protocol
RIR	Regional Internet Registries
RP	Rendezvous Point
RS	Router Solicitation
SAFI	Subsequent Address Family Identifier
SIIT	Stateless IP/ICMP Translation
SLAAC	StateLess Address AutoConfiguration
SNMA	Solicited-Node Multicast Address
SNMP	Simple Network Management Protocol
SPF	Shortest Path First
SPF	Shortest Path First
SSH	Secure Shell
TB	Tunnel Broker
TOS	Type of Service
TR-69	Technical Report 069
TTL	Time to Live
ULA	Unique Local Addresses
VPN	Virtual Private Network
YAML	Yet Another Setup Language

1. Introduction to Internet Protocol version 6 (IPv6, IPng)

IPv6 also called IPng is the replacement for IPv4. It has 3.4 x 10^{38} addresses (2^{128}) more than 7.9 x 10^{28} times as many as IPv4. This updated version of IP was developed by Steve Deering and Craig Mudge at Xerox PARC, it was then adopted by the Internet Engineering Task Force in 1994 as IPng.

The adoption of IPv6 has been slowed by the introduction of Network Address Translation (NAT), which partially alleviates IPv4 address exhaustion. Japan and Korea had started with their implementation of IPv6 in the late 1990's. The European Union (EU) formed an IPv6 Task Force as a steering committee in 2001 and member states all had their own IPv6 Task Forces by 2004. The United States of America (US) has specified that the network backbones of all federal agencies must have deployed IPv6 by 2008.

In October 2007 Vint Cerf the founder of the Internet issued a warning to operators of the urgent need to roll out IPv6 because the IPv4 pool is finite and has all but run out in 2012 (The RIPE NCC body started allocating its last */8* in September 2012). Each Local Internet Registry (LIR) received one final /22 allocation (1,024 IPv4 addresses) upon application for IPv4 resources. No new IPv4 Provider Independent (PI) space will be assigned.

By 2017 all the Regional Internet Registries (RIR) have exhausted their allocations with African Network Information Centre (AfriNIC), the African RIR being the last to declare on the 3 April 2017 [1].

It is expected that IPv4 will be supported alongside IPv6 for the foreseeable future with hosts running dual-stack software.

1 http://www.afrinic.net/en/library/news/2053-afrinic-enters-ipv4-exhaustion-phase-1

1.1 Features of IPv6

IPv6 supports many new features over IPv4, these features were developed considering the problems that were showing in IPv4.

- **Much larger address space**
  - 28 bit addressing gives 3.4 x 10^{38} address versus IPv4 giving 4.3 x 10^{9}, that is 7.9 x 10^{28} more addresses.
- **Multicast**
  - Multicast (both on the local link and across routers) is part of the base protocol suite in IPv6. This is different to IPv4, where multicast is optional.
  - IPv6 does not have a link-local broadcast facility; the same effect can be achieved by multicasting to the all-hosts group with a hop count of one.
- **Jumbograms**
  - In IPv4, the payload length field of 2 bytes limits the maximum theoretical payload to 65 kB. IPv6 has a payload length field of 4 bytes so when used between capable communication partners, IPv6 can support packets up to 4.3 GB in size, referred to as jumbograms. The use of jumbograms improves performance over high throughput networks.
- **Faster routing**
  - By using a simpler and more systematic header structure, IPv6 improves the performance of routing. Recent advances in router technology, however, may have made this improvement less relevant.
- **Network-layer security**
  - Internet Protocol Security (IPSec), the protocol for IP network-layer encryption and authentication, is an integral part of the base protocol suite in IPv6. It is, however, not yet deployed widely except for securing BGP traffic between IPv6 routers.
- **Mobility**
  - IPv6 was designed to support mobility. IPv6 Neighbour Discovery (ND) and SLAAC allow hosts to operate in any locations without any special support. This makes it more scalable and the performance is better because less traffic passes through the home link and less redirection and less rerouting. It also means no single point of failure.

1.2 IPv6 Address Scope

IPv6 address have a *scope* to specify where the address is valid. Within unicast addressing, Link-local Addresses (LLA) and the loopback address have ***link-local*** scope, which means they are to be used in the directly attached network (link) only. All other addresses, including Unique Local Addresses (ULA) and Global Unicast Addresses (GUA), have global (or universal) scope, which means they are globally routable, and can be used to connect to addresses with global scope anywhere, or addresses with *link-local* scope on the directly attached network. The scope of an *anycast* address is defined identically to that of a *unicast* address.

For multicasting, the four least-significant bits of the second address octet of a multicast address (ff0X::) define the address scope, the span over which the multicast address is propagated.

2. IPv6 Address Architecture

IPv6 addresses are normally written as 32 nibbles within 8 groups of 4 hexadecimal digits. For example, *2a02:2158:435a:0000:83:314:ea21:b33f* is a valid IPv6 address.

128 bits

⇣

32 nibbles

⇣

8 x 4 nibble groups

2.1 The Zero Suppression rule

Leading zeros in a group can be omitted. Thus

2a02:0201:0000:0000:0000:0000:00a1:b33f is shortened to *2a02:201:0:0:0:0:a1:b33f*.

2.2 The Zero Compression rule

Contiguous groups of '0' can be replaced with '::' as long as there is only one double colon used in an address.

2a02:201:0:0:0:0:0a1:b33f maybe shortened to 2a02:201::a1:b33f .

Having more than one double-colon abbreviation in an address is invalid as it would make the notation ambiguous. Leading zeros in a group can be omitted. Thus

2a02:0000:0000:0000:0022:0000:0000:b33f may be shortened as follows:

2a02:0:0:0:22:0:0:b33f (Zero suppression rule)

2a02::22:0:0:b33f (Zero compression rule)

Following the rules above, confirm if the addresses below are all valid and equivalent:

2a02:2158:0000:0000:0000:0000:00a1:b33f
2a02:2158:0000:0000:0000::00a1:b33f
2a02:2158:0:0:0:0:0a1:b33f
2a02:2158:0::0:0a1:b33f
2a02:2158::0a1:b33f
2a02:2158::a1:b33f

A sequence of 4 bytes at the end of an IPv6 address can also be written in decimal, using dots as separators. This notation is often used with compatibility addresses. Thus, *::ffff:1.2.3.4* is the same address as *::ffff:102:304*.

2.3 IPv6 Address Notation

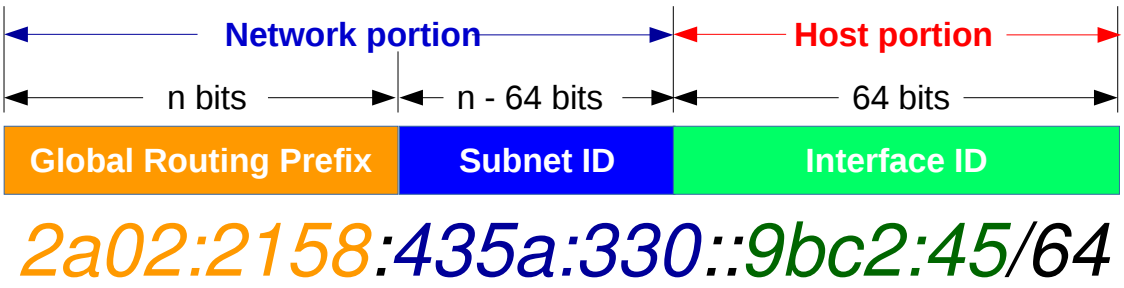


Illustration 1: IPv6 network notation

IPv6 networks are written using Classless Inter-Domain Routing (CIDR) notation. An IPv6 network is a contiguous group of IPv6 addresses the size of which must be a power of two; the initial bits of addresses which are identical for all hosts in the network are called the network prefix.

A network is denoted by the first address in the network and the size in bits of the prefix, separated with a slash. For example, *2a02:2158:435a:330::/64* stands for the network with

- First address: *2a02:2158:435a:330::*
- Last address: *2a02:2158:435a:330:ffff:ffff:ffff:ffff*

Because a single host can be seen as a network with a 128-bit prefix, a host address may be shown with */128* mask.

In IPv4 the first and last address in a prefix cannot be used to cater for broadcast functionality, as IPv6 does not have a broadcast address in the prefix all the addresses in the prefix are free to be used.

Like IPv4 the IPv6 Address is constructed of two parts the Prefix + host Identifier (ID) (Sometimes the Interface ID). The idea is to separate *who u are* from *where u are connected to*. The Prefix is dependant on the routing topology and the Interface ID identifies a node. IPv6 removes the Broadcast address and instead uses special Multicast addresses *all hosts ff0X::1* or *all routers ff0X::2* where X is replaced by the scope number. IPv6 also introduces a new *anycast* address. An *anycast* address is an IPv6 address that is assigned to one or more network interfaces, with the property that a packet sent to an *anycast* address is routed to the *nearest* interface having that address, according to the routing protocols measure of distance.

- **Unicast**: from one host to another.
- **Multicast**: from one to all belonging to a group.
- **Anycast**: from one to the nearest belonging to a group.

2.4 Special IPv6 addresses

2.4.1 Unspecified Address

IPv6 has a special address reserved for situations where a host does not have an IPv6 address but needs to send a packet. This is called the *unspecified* address.

- ::
- ::/128

2.4.2 Default route Address

This special address format is for the default route. Similar to 0.0.0.0/0 in IPv4.

- ::/0

2.4.3 Loopback Address

Similar to IPv4, IPv6 has a special address reserved for loopback.

- 0:0:0:0:0:0:0:1
- ::1
- ::1/128

2.5 IPv6 Packet Structure

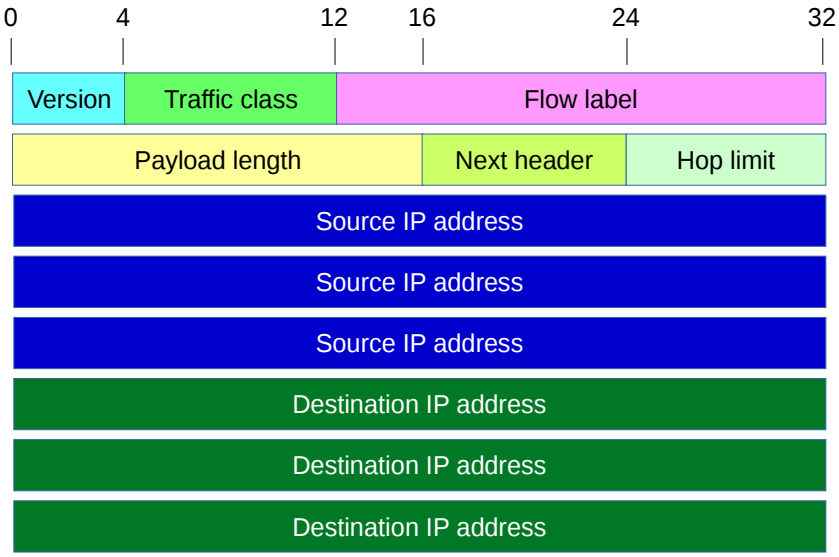


Illustration 2: IPv6 packet structure

The IPv6 packet header has many changes compared to the IPv4 header while maintaining necessary elements. Unlike the IPv4 packet header the IPv6 header is a fixed 40 bytes in size and adds extension headers to add additional information. Fragmentation is IPv6 only occurs at the source, intermediary routers will not fragment like IPv4 and if a router cannot pass the packet it drops it. This reduces processing by intermediary routers. The source node determines the available Maximum Transmission Unit (MTU) size via the Path MTU Discovery (PMTUD) process. In this process the source node tests various MTU sizes starting with its link-local MTU and judges based on received Internet Control Message Protocol version 6 (ICMPv6) *type 1 – Destination Unreachable* and *type 2 – Packet too big* error messages from upstream routers.

The IPv6 header contains:

Header	Bytes	Description
Version	4	Describes the version as 6
Traffic Class	8	One byte field
Flow Label	20	20 bit flow label for label tagging
Payload Length	16	Two byte integer giving the length of the packet less the base header but including the extension headers
Next Header	8	Specifies IPv6 extension headers or a upper layer protocol
Hop Limit	8	Single byte decremented at each router, packet discarded if zero
Source Address	128	Address of originator
Destination Address	128	Address of the destination

Here is an example IPv6 packet which has an IPv6 Hop by Hop extension header followed by ICMPv6 as the upper layer protocol. This will become clearer in the next section.

```
Frame: 90 bytes on wire (720 bits)
    Encapsulation type: Ethernet (1)
Ethernet II
    Destination: IPv6mcast_16 (33:33:00:00:00:16)
        .... ..1. .... .... .... .... = LG bit: Locally administered address
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: 00:00:00_aa:00:02
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: ::, Dst: ff02::16
    0110 .... = Version: 6
        .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
        .... 0000 00.. .... .... .... .... .... = Differentiated Services Codepoint: Default (0)
        .... .... ..00 .... .... .... .... .... = Explicit Congestion Notification:
                                                   Not ECN-Capable Transport (0)
        .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 36
    Next header: IPv6 Hop-by-Hop Option (0)
    Hop limit: 1
    Source: ::
    Destination: ff02::16
    Hop-by-Hop Options
        Next Header: ICMPv6 (58)
        Length: 0 (8 bytes)
        IPv6 Option (Router Alert)
            Type: Router Alert (5)
            Length: 2
            Router Alert: MLD (0)
        IPv6 Option (PadN)
            Type: PadN (1)
            Length: 0
            PadN: <MISSING>
Internet Control Message Protocol v6
    Type: Multicast Listener Report Message v2 (143)
    Code: 0
    Checksum: 0x6ede [correct]
    Reserved: 0000
    Number of Multicast Address Records: 1
    Multicast Address Record Changed to exclude: ff02::1:ffaa:2
        Record Type: Changed to exclude (4)
        Aux Data Len: 0
        Number of Sources: 0
        Multicast Address: ff02::1:ffaa:2
```

2.6 IPv6 Option headers

Unlike IPv4 the IPv6 options are handled outside the IPv6 header. This is achieved by the addition of extensions headers which are only processed as necessary. For example only routers process the *Hop by Hop options header*. With this method it is easier to define new extensions and options as the protocol evolves. Here is a list of some optional headers that are used with IPv6 today. They always appear in this order within packets if they are being added.

Header	Code	Description
Hop by Hop options	0	
Destination options	60	Examined only by destination node
Routing	43	Specify the route for a datagram
Fragment	44	Fragmentation parameters
Authentication header (AH)	51	Verify packet authenticity
Encapsulation security payload (ESP)	50	Encrypted data
Destination options	60	Examined only by destination node
Mobility (Mobile IPv6)	135	Parameters for use with mobile IPv6

2.7 Main field differences between IPv4 and IPv6 headers

The following are the main differences and comparison between IPv4 header and IPv6 header.

Maintained fields:

- Version
- Total length
- Source IP address
- Destination IP address

Similar fields:

- Type of Service (TOS) → Traffic Class
- Time to Live (TTL) → Hop Limit (HL)
- Protocol → Next Header

Header fields moved to Ipv6 extension headers:

- Identification
- Flags
- Fragment Offset
- Options
- Padding

Eliminated fields from IPv6:

- Internet Header Length (IHL)
- Header checksum

3. Basic IPv6 configuration

3.1 GNU/Linux and UNIX

To configure IPv6 Address on UNIX or GNU/Linux systems there are a number of possibilities depending on the system.

GNU/Linux uses the route2 module for IP management, add an IPv6 Address by

```
$ sudo ip -6 addr add 2001::ffff:20/112 dev eth0
$ sudo ip -6 route add ::/0 via 2001::ffff:1
```

Review the new configuration.

```
$ ip -6 addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 2001::ffff:20/112 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::d0d2:9abd:2c7e:876d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
$ ip -6 route show
::1 dev lo proto kernel metric 256 pref medium
2001::ffff:0/112 dev enp0s3 proto kernel metric 100 pref medium
fe80::/64 dev eth0 proto kernel metric 100 pref medium
default via fe80::223:ebff:fe6c:dd1b dev eth0 proto ra metric 20100 pref high
default via 2001::ffff:1 dev eth0 metric 1024 pref medium
```

```
$ ping6 2001::ffff:10
PING 2001::ffff:10(2001::ffff:10) 56 data bytes
64 bytes from 2001::ffff:10: icmp_seq=1 ttl=128 time=1.44 ms
64 bytes from 2001::ffff:10: icmp_seq=2 ttl=128 time=0.458 ms
64 bytes from 2001::ffff:10: icmp_seq=3 ttl=128 time=0.477 ms
64 bytes from 2001::ffff:10: icmp_seq=4 ttl=128 time=0.456 ms
64 bytes from 2001::ffff:10: icmp_seq=5 ttl=128 time=0.463 ms

--- 2001::ffff:10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.456/0.660/1.446/0.393 ms
```

3.1.1 Persistent configuration

Using the /**etc**/**network**/**interfaces** file.

```
$ sudo vi /etc/network/interfaces
  iface eth0 inet6 static
     address 2001::ffff:0020
     netmask 112
     gateway 2001::ffff:0001
  ~
  :wq!

$ sudo ip link set dev eth0 down
$ sudo ip link set dev eth0 up
```

Using the **netplan Yet Another Setup Language (YAML)** file.

```
$ sudo vi /etc/netplan/01-network-manager-all.yaml
  network:
    version: 2
    renderer: networkd
    ethernets:
      eth0:
        accept-ra: no
        addresses:
          - 2001::ffff:0020/112
        gateway6: 2001::ffff:0001
  ~
  :wq!

$ sudo netplan apply
```

Add the DNS Server as system-wide DNS server. This is done in /**etc**/**systemd**/**resolved.conf** file:

```
$ sudo vi /etc/systemd/resolved.conf
[Resolve]
DNS=2001:4860:4860::8888 2001:4860:4860::8844
```

Reload configuration and to restart services:

```
$ sudo systemctl daemon-reload
$ sudo systemctl restart systemd-networkd
$ sudo systemctl restart systemd-resolved
```

3.2 Microsoft Windows

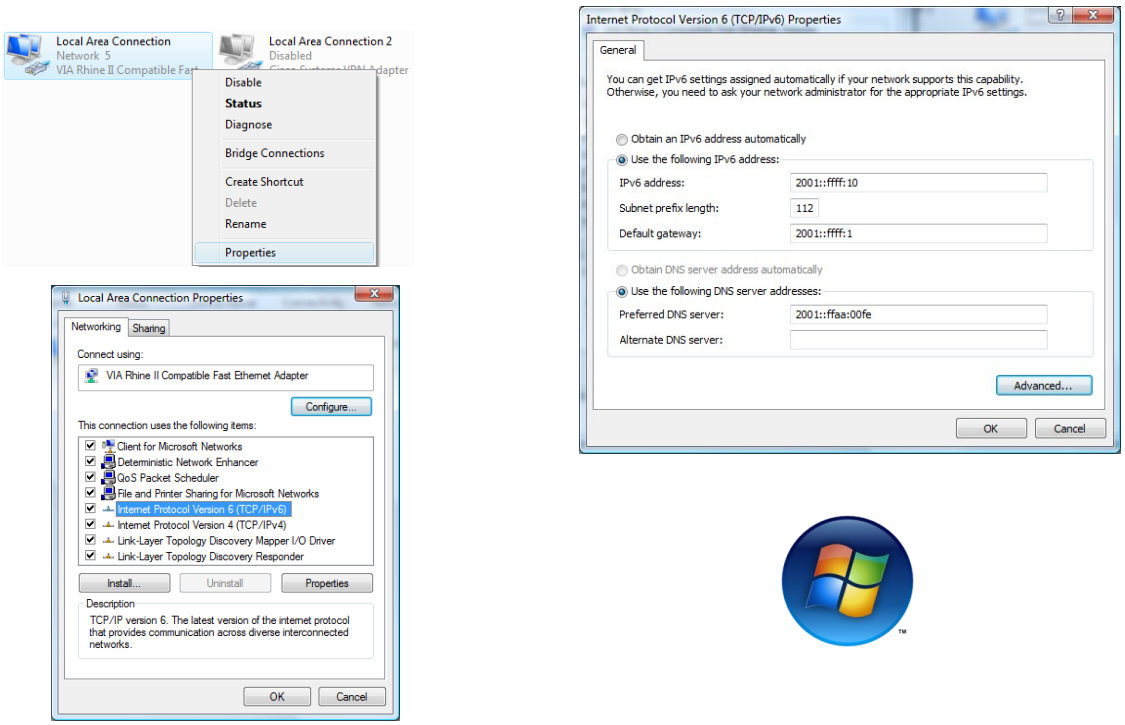


Illustration 3: Configure IPv6 on Microsoft Windows

On Windows access the network configuration via:

Start → Settings → Network Connections

Test the configuration.

```
C:\> ping 2001::ffff:20

Pinging 2001::ffff:20 from 2001::ffff:10 with 32 bytes of data:

Reply from 2001::ffff:20: time<1ms
Reply from 2001::ffff:20: time<1ms
Reply from 2001::ffff:20: time<1ms
Reply from 2001::ffff:20: time<1ms

Ping statistics for 2001::ffff:20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

4. IPv6 Prefix Terminology

IPv6 does not have a *classful* concept like IPv4 but within GUA assignments there are a number of prefixes, with different prefix lengths. The table below outlines these key prefix terms.

Prefix Term	Assigned by	Example prefix
Registry Prefix	Assigned to Regional Registry (RR)	2a02::/12
ISP Prefix	Assigned to Internet Service Provider (ISP)	2a02:2158::/32
Site Prefix	Assigned to Large Organisation	2a02:2158:1111::/48
Site Prefix	Assigned to Smaller Organisation	2a02:2158:1111:100::/56
Subnet Prefix	Internal subnet within Organisation	2a02:2158:1111:110::/64
A host address	Organisation/Residential home user	2a02:2158:1111:110::10/128

The following table give an indication of IPv6 Relative Network Sizes.

Mask	Size	Description
128	1 IPv6 Address	A network interface
64	1 IPv6 subnet	18,446,744,073,709,551,616 IPv6 addresses
56	256 LAN segments	Popular prefix size for smaller subscriber site
48	65,536 LAN segments	Popular prefix size for larger subscriber site
32	65,536 /48 subscriber sites	Minimum IPv6 allocation by RR
24	16,777,216 subscriber sites	256 times larger than the min IPv6 allocation

4.1 IPv6 assignments

4.1.1 Provider Aggregatable (PA) Assignments

Provider Aggregatable (PA) addresses are assigned from a LIR allocation and are registered in the RIR by the LIR. The advantage of PA address space is that the routing information for many customers can be aggregated once it leaves the provider's routing domain. The minimum assignment to a LIR from the RIR is /32.

4.1.2 Provider Independent (PI) Assignments

The RIR will assign Provider Independent (PI) prefix directly to the End User organisations. The minimum size of these assignments is /48. Organisations requesting a larger assignment (shorter prefix) must provide documentation justifying the need for additional subnets.

Additional assignments may also be made when the need is demonstrated and documented based on address usage, or because different routing requirements exist for additional assignments. When possible, these further assignments will be made from an adjacent address block. The PI assignment cannot be further sub-assigned to other organisations.

4.2 Link-Local Address (LLA)

The address block fe80::/10 has been reserved for link-local unicast addressing. To conform to standard /64 addressing on subnets, the actual LLA are assigned with the prefix fe80::/64. The 54 bits after the most significant ten bits must be zero.

IPv6 requires an LLA on every network interface on which the IPv6 protocol is enabled, even when routable addresses are also assigned. Therefore IPv6 hosts usually have more than one IPv6 address assigned to each of their IPv6-enabled network interfaces. The LLA is required for the Neighbour Discovery Protocol (NDP) and DHCPv6.

LLA addresses can be assigned automatically by a process called SLAAC using NDP or manually.

4.2.1 Forming an LLA

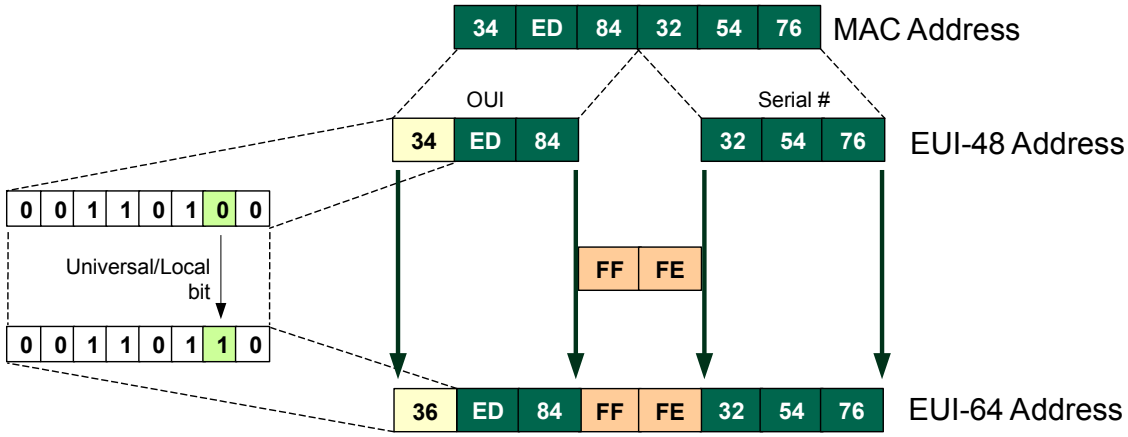


Illustration 4: Forming an EUI-64 MAC from an EUI-48 MAC

Nodes, both hosts and routers generate an LLA for each interface on boot. Such an LLA is formed by appending the Interface IDentifier (Interface ID) to the well-known Link-local prefix *fe80::*.

EUI-64 : 36ed:84ff:fe32:5476 → IPv6 Host ID : fe80::36ed:84ff:fe32:5476

The first step is to generate an EUI-64 Medium Access Control (MAC) address. IPv6 uses 64 bits for the network and subnets while it reserves the last 64 bits to identify the host. Traditionally MAC addresses are of the EUI-48 type with 48 bits. However the range of unique EUI-48 MAC addresses are running out and it was decided to migrate to EUI-64 format in the future. IPv6 was built for EUI-64 addresses.

To convert process of an EUI-48 to EUI-64 refer to Illustration 4 where the original address is split and FF:FE inserted. The 6th bit, called the Universal/local bit in the first octet is changed to a '1' to indicate the new MAC is not unique.

Universal/local bit	0 = Unique MAC
	1 = non-Unique MAC

4.2.2 Resolving LLA ambiguity with zone IDs

As LLA fe80::/10 address exist on all interfaces and therefore each interface is part of the same network prefix it is necessary to let the device know which interface to use when communicating with a neighbouring node. Basically the node has no way to determine the interface to send packet out on.

Here is an example where a ping to an IPv6 LLA address fails on a router.

```
n1# ping ipv6 fe80::200:ff:feaa:0
connect: Invalid argument
```

Appending the **%<interfaceID>** to the LLA resolves this ambiguity on the router.

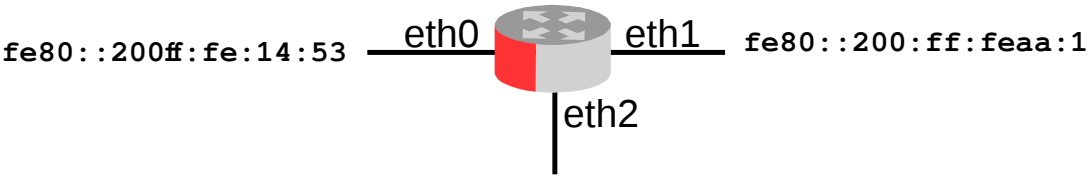


Illustration 5: Resolving LLA ambiguity

```
n1# ping ipv6 fe80::200:ff:feaa:0%eth1
PING fe80::200:ff:feaa:0%eth1(fe80::200:ff:feaa:0) 56 data bytes
64 bytes from fe80::200:ff:feaa:0: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from fe80::200:ff:feaa:0: icmp_seq=2 ttl=64 time=0.036 ms
```

Another example, this time from a host.

```
$ ssh fe80::287b:8236:8feb:df65%wlp4s0
alovelace@fe80::287b:8236:8feb:df65%wlp4s0's password: babbage
Welcome to Linux Mint 18.1 Serena (GNU/Linux 4.4.0-53-generic x86_64)

 * Documentation:  https://www.linuxmint.com
Last login: Tue Jan  3 23:04:33 2017
alovelace@remote ~ $
```

4.2.3 Reserved Interface IDs (RFC 5453)

Interface Identifier Range	Description
0000:0000:0000:0000	Subnet router anycast
FDFF:FFFF:FFFF:FF80 - FDFF:FFFF:FFFF:FFFF	Reserved Subnet anycast

5. IPv6 Address planning

An ISP performing address planning must consider how many addresses to request from the RIR and therefore needs to plan their address space carefully. Take for example an ISP in a country with 10 regions, each with 50 Points of Presence (POP) and each of these supporting 3,500 clients. First, calculate the number of bits in the mask for each tier in multiples of 4 (i.e. nibbles). Take the value that gives a result higher that the requirement. For example 12 bits = 2^{12} = 4,096 where 2^8 = 256 so 12 bits are required for the blocks of 3,500 clients. A similar process is carried out for each item as outlined below. Assuming each client is assigned a /48 the mask for POPs can be determined by subtracting 12 from 48 giving a /36 for POPs and subtract 8 from 36 to give a /28 for regions and finally subtracting 4 from 28 gives a /24 for the ISP. Therefore in this example the IP address space requires a /24 from the RIR.

Item	#	Bits (multiple of 4)	Possible #	Mask
ISP	1	1	2	/24
Regions/Cities	10	4	16	/28
POPs	50	8	256	/36
Clients	3,500	12	4,096	/**48**

24

Assignment per client (/48) – Clients per POP (12) – POPs (8) – Regions (4) = /**24**

6. IPv6 Multicast address

An IPv6 multicast address is an identifier for a group of interfaces that are typically on different nodes. An interface may belong to any number of multicast groups. Multicast addresses have the following format:

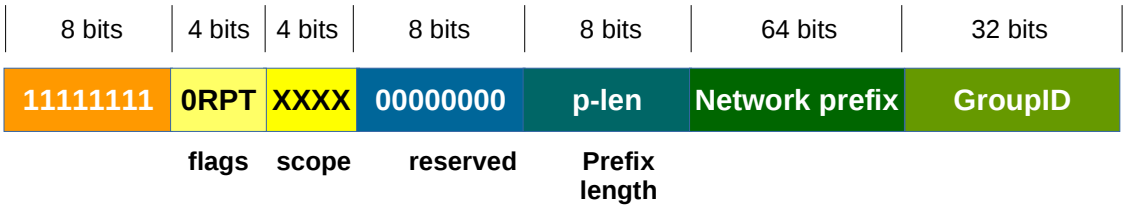


Illustration 6: IPv6 Multicast identifier

6.1 Flags

The multicast identifier contain a 4 bit flag field.

Nibble value	flag	Meaning when 0	Meaning when 1
8	reserved	reserved	reserved
9	R (Rendezvous)	RP not embedded	RP embedded
10	P (Prefix)	Without prefix info	Address based on network prefix
11	T (Transient)	Well-known multicast address	Dynamically assigned multicast address

R = 0 indicates a multicast address that does not embed the address of the Rendezvous Point (RP) and the value of RP Interface Identifier (RIID) MUST be sent as zero and MUST be ignored on receipt. A RP is the downstream point of intersection for multicast streams.

R = 1 indicates a multicast address that embeds the address on the RP. Then P MUST be set to 1, and consequently T MUST be set to 1 also.

6.2 Scope

Multicast scope is a 4-bit value used to limit the scope of the multicast group.

ff0X::/8	Meaning
0x1	Interface local
0x2	Link local
0x4	Admin local
0x5	Site local
0x8	Organisation local
0xE	Global
0x0	Reserved
0xF	Reserved

6.3 GroupID

The group ID identifies the multicast group, either permanent or transient, within the given scope.

ID	Meaning
1	All nodes on the local network segment
2	All routers on the local network segment
5	Open Shortest Path First (OSPF) v3 (OSPFv3) All SPF routers
6	OSPFv3 All DR routers
8	Intermediate System to Intermediate System (IS-IS) for IPv6 routers
9	Routing Internet Protocol (RIP) routers
a	Enhanced Interior Gateway Routing Protocol (EIGRP) routers
d	Protocol Independent Multicast (PIM) routers
16	MLDv2 reports (defined in RFC 3810)
1:2	All Dynamic Host Configuration Protocol (DHCP) servers and relay agents on the local network segment (defined in RFC 3315)
1:ff	Solicited-Node Multicast Address (SNMA)
fb	Multicast DNS
101	Network Time Protocol (NTP)
108	Network Information Service (NIS)
181	Precision Time Protocol (PTP)
114	Used for experiments

Examples:

- ff05::1 All nodes on the local site
- ff02::2 All routers on the link local
- ff02::5 All OSPF routers on the link local
- ff02::9 All RIPng routers on the link local
- ff02::a All EIGRP routers on the link local
- ff05::101 All NTP Servers on the local site
- ff02::1:3 All DHCPv6 servers on the link local

6.3.1 Multicast MAC

A corresponding multicast MAC is associated with each unicast address. For example:

- ff02::1 → 33:33:00:00:00:01
- ff02::2 → 33:33:00:00:00:02
- ff02::5 → 33:33:00:00:00:05
- ff02::6 → 33:33:00:00:00:06

6.4 Solicited-Node Multicast Group Address (SNMA)

Every device that uses an IPv6 address will also compute and join a SNMA. This address is required for the IPv6 NDP.

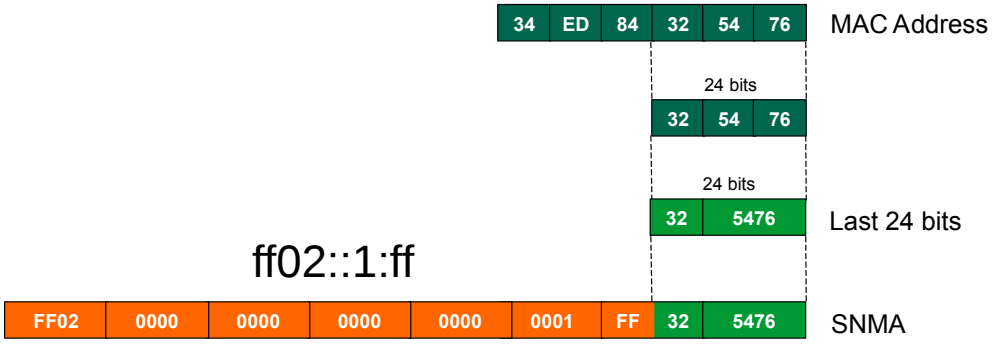


Illustration 7: Solicited-Node Multicast Group Address formation

The process of generating such an address is simple. The last 24-bits of the interface MAC address is acquired. These 24 bits are prepended with the special SNMA network ff02::1:ff/104. The example demonstrates the EUI-48: *34ed:8432:5476* MAC address being used to generate an SNMA: *ff02::1:ff:32:5476*. A corresponding MAC address for the multicast IPv6 address is also created. This takes the last 32 bits of the SNMA address and prepends 33:33. In this example that would be *33:33:ff:32:54:76.*

6.5 Special Prefix's

There are a number of specific addresses within IPv6 with special meaning:

Prefix	Meaning
::/0	The default unicast route address (similar to 0.0.0.0/0 in IPv4)
::/128	The address with all zeroes is an unspecified address, and is only to be used in software
::1/128	The loopback address is a localhost address. (like 127.0.0.1 in IPv4)
::ffff:0:0/96	This prefix is used for IPv4 mapped addresses. Transparent use of Transport Layer protocols over IPv4 through IPv6 API
64:ff9b::/96	Well known prefix for 6to4 address translation.
0400::/7	Internetwork Packet Exchange (IPX) from the IPX/SPX protocol stack routed via IPv6
2000::/3	**Global Unicast Address (GUA):** 2000:: - 3fff::
fc00::/7	**Unique Local Address (ULA)**: are only routable within a set of cooperating sites.
fe80::/10	**Link-local Address (LLA)**: Prefix specifies that the address only is valid in the local physical link. (like the Auto-configuration address 169.254.x.x in IPv4)
ff00::/8	The multicast prefix for multicast addresses
ff01::0/12	Pre-defined Multicast addresses
ff01::1/12	All host addresses (interface-local)
ff01::2/12	All routers (interface-local)
ff02::1/12	All host addresses (link-local)
ff02::2/12	All routers (link-local)
ff02::1:2/12	All DHCP servers and relay agents
ff05::2/12	All routers (site-local)

6.5.1 Depreciated Prefix's

The following prefixes were originally defined as part of IPv6 but have since been depreciated or obsoleted. I have added them here for information in case you come across such addresses.

Prefix	Meaning
::/96	The zero prefix was used for IPv4-compatible addresses. Depreciated in February 2006.
fec0::/10	Site-local prefix specifies that the address is only valid inside the local organisation. Its use has been deprecated in September 2004 by IPv6 Deprecating Site Local Addresses RFC and future systems must not implement any support for this special type of address any more.
0200::/7	Network Service Access Point (NSAP) addresses from ISO/IEC 8348 routed via IPv6. Deprecated in December 2004.

7. Applications for IPv6

7.1 DHCP for IPv6 (DHCPv6)

Although IPv6's SLAAC removes the primary motivation for DHCP in IPv4, DHCP for IPv6 (DHCPv6) can still be used to statefully assign addresses if the network administrator desires more control over address management. It can also be used to distribute information which is not otherwise discoverable; the most important case of this is the Domain Name Server (DNS) server and domain name search list.

A major difference with DHCPv4 Servers is that hosts send broadcasts to find DHCP Servers whereas with DHCPv6 Servers IPv6 hosts send IPv6 multicast to the well known DHCPv6 multicast group *ff02::1:2*.

7.2 DNS Extensions to Support IP Version 6 (DNSv6)

DNS is similar for IPv4 and IPv6 (DNSv6). The main difference is that the *A* record is replaced by the *AAAA* record which maps a hostname to a 128-bit IPv6 address for forward lookups. Reverse lookups take place under *ip6.arpa*, where address space is delegated on nibble boundaries. This scheme is a straightforward adaptation of the familiar A record and the *in-addr.arpa* schemes for IPv4.

7.3 ICMPv6 for IPv6

ICMP version 6 (ICMPv6) is a new version of ICMP and is an integral part of the IPv6 architecture that must be completely supported by all IPv6 nodes. ICMPv6 combines functions previously subdivided among different protocols, such as ICMP, Internet Group Membership Protocol (IGMP), and Address Resolution Protocol (ARP). It introduces some simplifications by eliminating obsolete types of messages no longer in use.

ICMPv6 is a multi-purpose protocol and it is used for reporting errors encountered in the processing of packets, performing diagnostics, performing Neighbour Discovery (ND) and reporting IPv6 multicast memberships. For this reason, ICMPv6 messages are subdivided into two classes:

7.3.1 Error messages

The first type of ICMPv6 message is the error message. ICMPv6 is used by IPv6 nodes to report errors encountered.

Type	Message
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem

7.3.2 Information messages

The second type of ICMPv6 message is the informational message type which is subdivided into three groups: diagnostic, management of multicast groups, and ND messages.

Type	Message
128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbour Solicitation
136	Neighbour Advertisement
137	Redirect
138	Router Renumbering

8. IPv6 ND and SLAAC

Neighbour Discovery (ND) has a number of key functions essential to the running of IPv6. ND uses the ICMPv6 messages Neighbour Solicitation (NS), Neighbour Advertisement (NA), Router Solicitation (RS), Router Advertisement (RA) and Redirect to achieve the following functions:

For all **Nodes**:

- Address configuration (SLAAC)
- Link-layer address resolution
- Link-layer address change notification
- Neighbour Unreachability Detection (NUD).

For **Hosts**:

- Router discovery
- Parameter discovery (MTU, prefixes, hop limits).

For **Routers**:

- Advertise their presence & parameters
- Advertise on-link prefixes
- Determine next hops
- Redirect hosts to better next hops.

8.1 IPv6 Stateless Address Auto-configuration (SLAAC)

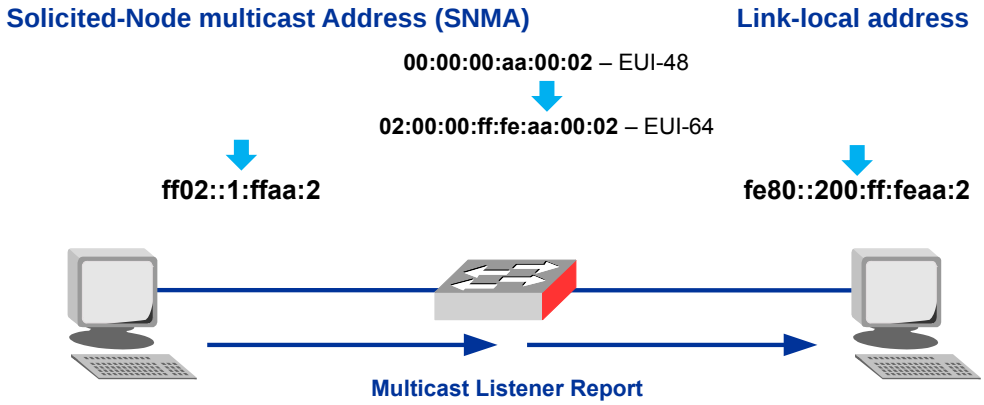
SLAAC is an IPv6 process that removes the requirement for the manual configuration of hosts, minimal configuration of routers with no additional servers. This stateless mechanism, based on ICMPv6, enables a host to generate its own global address. The stateless mechanism uses local information as well as non-local information advertised by routers to generate the addresses.

Routers advertise prefixes that identify the subnet or subnets that are associated with a link. Hosts generate an interface identifier that uniquely identifies an interface on a subnet. An address is formed by combining the prefix and the interface identifier. In the absence of routers, a host can generate only link-local addresses. However, link-local addresses are sufficient for communication among nodes that are attached to the same link.

8.2 Link Local Address (LLA)

The host sends a Multicast Listener Report to the Multicast Listener Discovery (MLD) well known address *ff02::16.* Illustration 8 demonstrates that for MAC address *00:00:00:aa:00:02* the LLA address *fe80::200:ff:feaa:2* is generated. The purpose of MLD is to enable multicast routers to learn which multicast addresses and which sources have interested listeners on that link. The information gathered by MLD is provided to whichever multicast routing protocol is used by the router, in order to ensure that multicast packets are delivered to all links where there are listeners interested in such packets. The multicast *changed to exclude* means that the specified multicast address should be excluded from the filter mode on that interface.

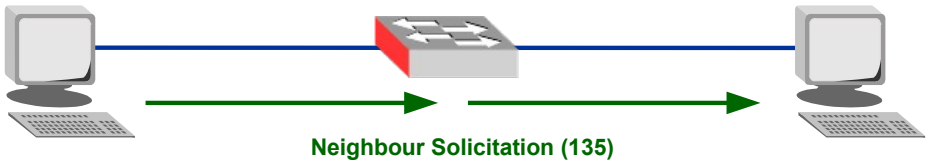
8.3 MLD joins multicast group




```
Ethernet II, Dst: 33:33:ff:aa:00:02, Src: 00:00:00:aa:00:02; Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: ::, Dst: ff02::16
   Payload length: 36
   Next header: IPv6 Hop-by-Hop Option (0)
   Hop limit: 255
   Hop-by-Hop Options
      Next Header: ICMPv6 (58)
Internet Control Message Protocol v6
   Type: Multicast Listener Report Message v2 (143)
   Multicast Address Record Changed to exclude: fe80::200:ff:feaa:2
```

Illustration 8: MLD joins multicast group

8.4 Neighbour Solicitation (135)




```
Ethernet II, Dst: 33:33:ff:aa:00:02, Src: 00:00:00:aa:00:02; Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: ::, Dst: ff02::1:ffaa:2
   Payload length: 24
   Next header: ICMPv6 (58)
   Hop limit: 255
Internet Control Message Protocol v6
   Type: Neighbor Solicitation (135)
   Target Address: fe80::200:ff:feaa:2
```

Illustration 9: Neighbour Solicitation (135)

The host then sends a NS (135) ICMPv6 message to the SNMA address from the unassigned address (::) with the LLA as the target address. If a NA (136) is received then a duplicate address has been detected and the process stops. If no NA (136) is detected then the process continues.

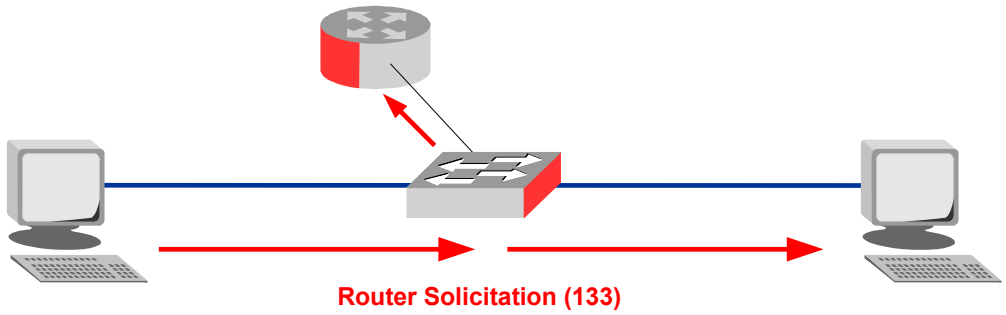
8.5 MLD joins multicast group for the second time


```
Ethernet II, Dst: 33:33:00:00:00:16, Src: 00:00:00:aa:00:02; Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: fe80::200:ff:feaa:2, Dst: ff02::16
   Payload length: 36
   Next header: IPv6 Hop-by-Hop Option (0)
   Hop limit: 255
   Hop-by-Hop Options
      Next Header: ICMPv6 (58)
Internet Control Message Protocol v6
   Type: Multicast Listener Report Message v2 (143)
   Multicast Address Record Changed to exclude: ff02::1:ffaa:2
```

Illustration 10: MLD joins multicast group (2)

The host sends a second Multicast Listener Report to the MLD multicast address *ff02::16* but this time with the LLA as the source and the SNMA as the *Multicast Address Record Changed to exclude*.

8.6 Router Solicitation (133)

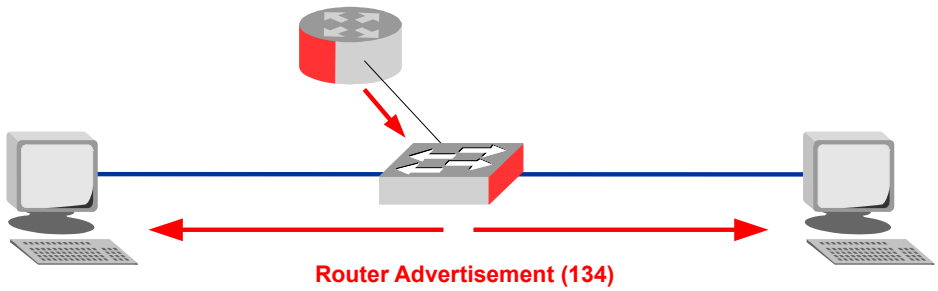



```
Ethernet II, Dst: 33:33:00:00:00:02, Src: 00:00:00:aa:00:02;
Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: fe80::200:ff:feaa:2, Dst: ff02::2
    Payload length: 16
    Next header: ICMPv6 (58)
    Hop limit: 255
Internet Control Message Protocol v6
    Type: Router Solicitation (133)
    Link-layer address: 00:00:00:aa:00:02
```

Illustration 11: Router Solicitation (133)

The host now sends an RS (133) ICMPv6 message to the multicast group *ff02::2*, the well known multicast group of all routers on the link-local using its LLA as the source IPv6 address.

8.7 Router Advertisement (134)

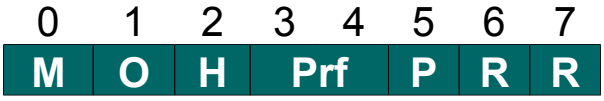



```
Ethernet II, Dst: 33:33:00:00:00:01, Src: 00:00:00:aa:00:03; Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: fe80::200:ff:feaa:3, Dst: ff02::1
   Payload length: 16
   Next header: ICMPv6 (58)
   Hop limit: 255
Internet Control Message Protocol v6
   Type: Router Advertisement (134)
   Flags: 0xc0          .1.. .... = Other configuration: Set
   ICMPv6 Option, Prefix information
      Prefix Length: 64
      Flag: 0xc0        1... .... = On-link flag (L): Set
                        .1.. .... = Autonomous address-configuration flag (A): Set
      Valid Lifetime: 86400, Preferred Lifetime: 86400
      Prefix: 2001:a::
   ICMPv6 Option
      Link-layer address: 00:00:00:aa:00:03
```

Illustration 12: Router Advertisement (134)

The router responds with an RA (134) to the multicast group *ff02::1*, the well known group of all hosts on the link-local. As part of this message it sends the GUA prefix *2001:1::/64* with the prefix length to the hosts plus some flags.

8.7.1 Router Advertisement flags




```
ICMPv6 Option (Prefix information)
      Type: Prefix information (3)
      Length: 32
      Prefix length: 64
      Flags: 0xc0
         0... .... = IP Address not DHCPv6
         .1.. .... = Other config on DHCPv6
         ..0. .... = Not router address
         ...0 .... = Not site prefix
      Valid lifetime: 86400
      Preferred lifetime: 86400
      Prefix: 2001:a::
```

Illustration 13: RA flags

RA (134) informational messages contain two flags that indicate what type of Auto-configuration should be performed. A Managed address configuration flag (M-Flag) indicates whether hosts should use stateful DHCPv6 (1) or SLAAC (0) to obtain global scope IPv6 addresses. The other stateful configuration flag (O-Flag) if set (1) indicates that hosts should use a stateless DHCPv6 Server to obtain additional information, excluding addresses. The O-flag is meaningless if the M-flag is set.

Additionally the RA (134) prefix flags have the options On-link flag (L-flag) and Address configuration flag (A-flag). The L-flag announces that other devices with the same prefix are on the same subnet and the host should communicate between them using only switch (L2) and not to send every message to the router and then from the router to another host on the same subnet. If the A-flag is set (1) the RA is telling the node to use SLAAC to configure addressing information.

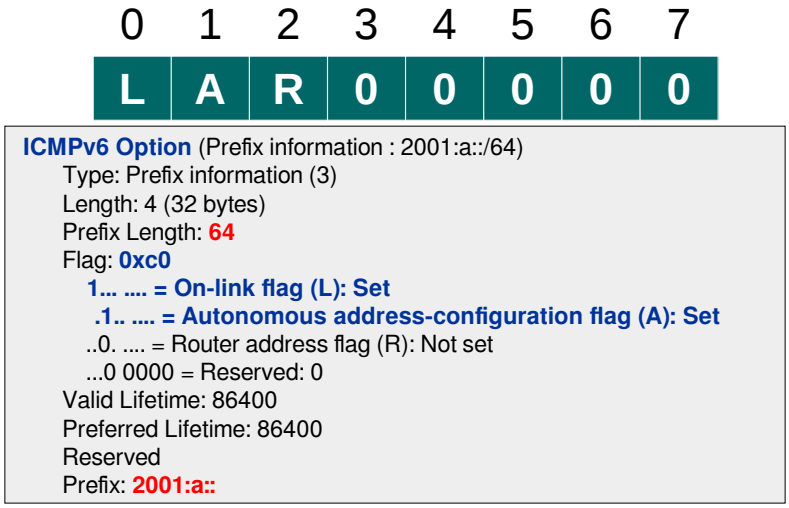
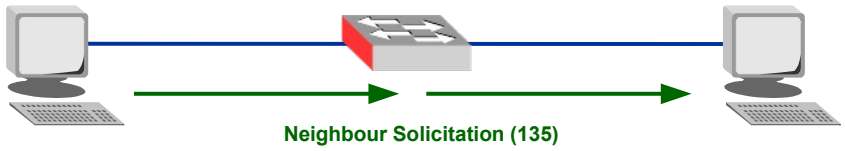


Illustration 14: RA - Prefix flags

The following table outlines the influence of the '**M**' & '**A**' flags on auto-configuration. Remember that hosts must be set to obtain IP address *automatically* and all hosts continue to generate and use a LLAs.

M	A	Resulting non-Link Local addresses on client
0	0	No addresses will be auto-configured
0	1	Address generated from prefix in RAs
1	1	Address generated from prefix in RAs or full address from DHCPv6 server
1	0	Full address from DHCPv6 server

8.8 Neighbour Solicitation (135) for the second time




```
Ethernet II, Dst: 33:33:ff:aa:00:02, Src: 00:00:00:aa:00:02; Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: ::, Dst: ff02::1:ffaa:2
    Payload length: 24
    Next header: ICMPv6 (58)
    Hop limit: 255
Internet Control Message Protocol v6
    Type: Neighbor Solicitation (135)
    Target Address: 2001:a::200:ff:feaa:2
```

Illustration 15: Neighbour Solicitation (135) (2)

Having received a prefix from the router the host generates a GUA by adding its EUI-64 MAC that it used to generate the LLA to the prefix. It then sends a NS (135) to the SNMA address from the unassigned address (::) with the new GUA as the target address. It doesn't expect a responding NA (136) and should it receive one the SLAAC process will stop.

8.9 MLD joins multicast group for the third time


```
Ethernet II, Dst: 33:33:00:00:00:16, Src: 00:00:00:aa:00:02; Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: fe80::200:ff:feaa:2, Dst: ff02::16
    Payload length: 36
    Next header: IPv6 Hop-by-Hop Option (0)
    Hop limit: 255
    Hop-by-Hop Options
        Next Header: ICMPv6 (58)
Internet Control Message Protocol v6
    Type: Multicast Listener Report Message v2 (143)
    Multicast Address Record Changed to exclude: ff02::1:ffaa:2
```

Illustration 16: MLD joins multicast group (3)

The host then sends a third Multicast Listener Report to the MLD multicast address *ff02::16* from the LLA as the source and the SNMA as the Multicast Address Record Changed to exclude value.

8.10 NDP Summary

Type	Node	Src addr	Dst addr	Notes
NS/135	Host	IP or ::	IP or SNMA	
NA/136	Host	IP	IP or FF02::1	R - flag
RS/133	Host	LLA or ::	FF02::2	
RA/134	Router	LLA	FF02::1	Flags: (M)anage & (O)ther
RD/137	Router	LLA	IP of node	Next hop

8.11 SLAAC Summary

The steps to SLAAC are:

- Host creates a SNMA
  - Host registers a Multicast Listener Report for SNMA to join group
  - from (::) to ff02::16 MLD.
- Host creates a LLA.
- Sends NS (135) from (::) to SNMA with LLA as target
  - If NA (136) received auto-configuration stops.
- Host registers a Multicast Listener Report for SNMA address to join group
  - from LLA to ff02::16 MLD.
- Host sends RS (133) to ff02::2 ‘All routers’ from LLA.
- Router sends RA (134) to ff02::1 ‘All nodes’
  - from its LLA with prefix.
- Host creates GUA from prefix and EUI-64 MAC
  - Sends NS (135) from (::) to SNMA with GUA target
  - If NA (136) received auto-configuration stops.
- Finish SLAAC.

8.12 DHCP for IPv6 (DHCPv6)

An alternative to using the SLAAC approach is to use DHCPv6 to assign IP addressing and network information. A major difference with DHCPv4 Servers is that hosts send broadcasts to find DHCP Servers whereas with DHCPv6 Servers, IPv6 hosts send messages to the well known multicast address *FF02::1:2*.

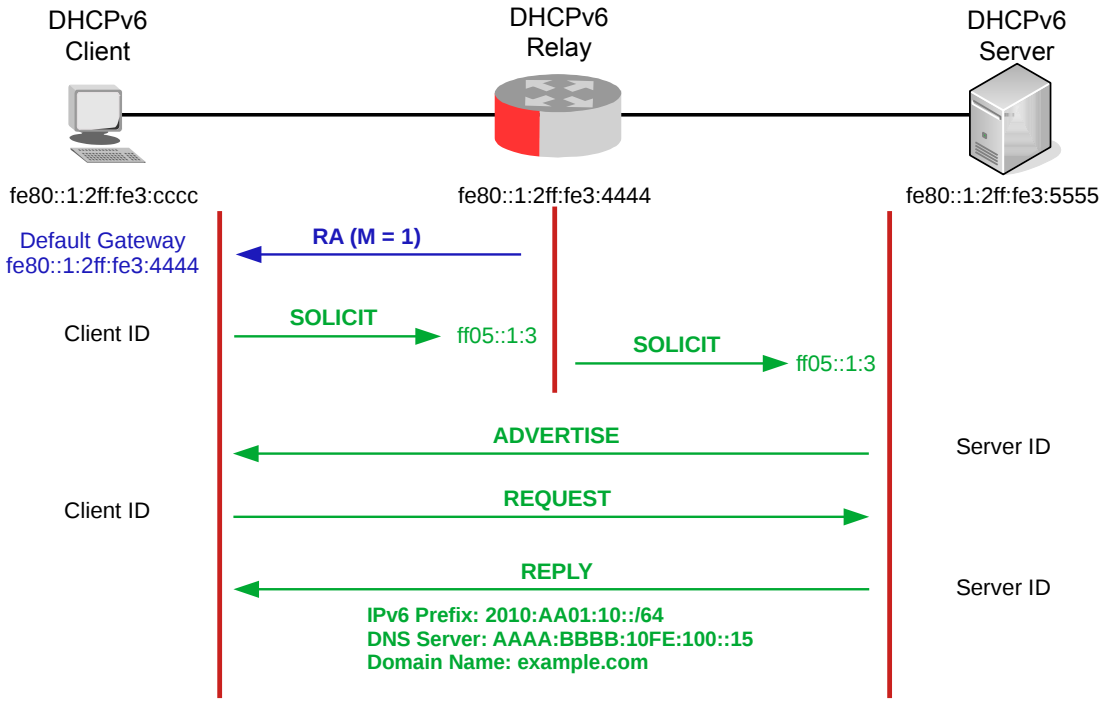


Illustration 17: Stateful DHCPv6

8.12.1 Stateful DHCPv6

Consider Illustration 17 where the client has acquired a link-local default gateway from the a Router Advertisement (RA) that also has the M-flag set. The client generates a unique client IDentifier and sends it in a DHCP Unique Identifier (DUID) within a SOLICIT message to the router. In many cases the router is also the DHCPv6 Server, otherwise it acts as a DHCPv6 Relay and forwards the SOLICIT message to the DHCPv6 Server. The DHCPv6 Server sends an ADVERTISE message to the client with its Server ID in a DUID. The client then sends a REQUEST message to the DHCPv6 Server and received a REPLY message with the requested options as demonstrated.

There is an additional DHCPv6 process called *Rapid-Commit* that reduces this process to just SOLICIT and REPLY messages. The client can request the *Rapid-Commit* option in the SOLICIT and if this option is configured in the DHCPv6 Server it will respond directly with a REPLY message.

8.12.2 Stateless DHCPv6

Should the client receive the RA from the router with the M-flag = 0 and the O-flag = 1 then it will not use DHCPv6 to get IPv6 addressing information. It will get its addressing information from SLAAC or manual configuration. It only requests stateless information like DNS Server and domain name as it has already received the IPv6 addressing information.

The following example demonstrates this. The router has a DHCPv6 pool that only includes DNS and domain information. The O-flag is set so the client will SOLICIT DHCPv6 information from the router and will be supplied with the DNS Server IPv6 addresses and the domain name information.

```
Router(config)# ipv6 dhcp pool dhcp-pool
Router(config-dhcp)# dns-server 2001:db8::d75
Router(config-dhcp)# dns-server 2001:db8::d76
Router(config-dhcp)# domain-name example.com

Router(config)# interface ethernet 0/0
Router(config-if)# ipv6 address 2001:dB8:1234::1/64
Router(config-if)# ipv6 nd other-config-flag
Router(config-if)# ipv6 dhcp server dhcp-pool
```

8.13 Recursive DNS Server (RDNSS)

An alternative case is the client receiving an RA from the router with the M-flag = 0 and the O-flag = 0. In this case the addressing information is also received from SLAAC or manual configuration. Additional RA options, Recursive DNS Server (RDNSS) contains the address of recursive DNS servers that help in DNS name resolution in IPv6 hosts and DNS Search List Option (DNSSL) is a list of DNS suffix domain names used by IPv6 hosts when they perform DNS query searches.

The following example demonstrates this. The router is configured with DNS Servers and a search list of domain names that can be supplied via RA options.

```
Router(config)# interface ethernet 1/0
Router(config-if)# ipv6 nd ra dns server 2001:db8::d75 sequence 0
Router(config-if)# ipv6 nd ra dns server 2001:db8::d76 sequence 1
Router(config-if)# ipv6 nd ra dns search-list example.com sequence 0
```

9. IPv6 Address Resolution and redirection

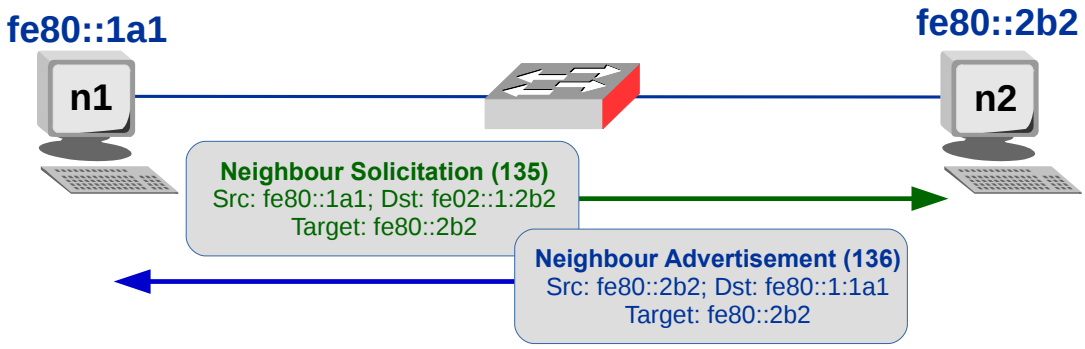


Illustration 18: Address Resolution

The IPv6 Address Resolution process uses ND to verify connectivity and establish two-way connectivity. A NS (135) packed is sent from the hosts LLA to the SNMA address of the target node with the target LLA within the ICMPv6 header. If the targetted node is available it responds with a NA (136) to the LLA of the requestor from its own LLA and with its own LLA as the target address within the ICMPv6 header. Node *n1* can now record an established two-way connectivity with node *n2*.

9.1 Neighbour Unreachability Detection (NUD)

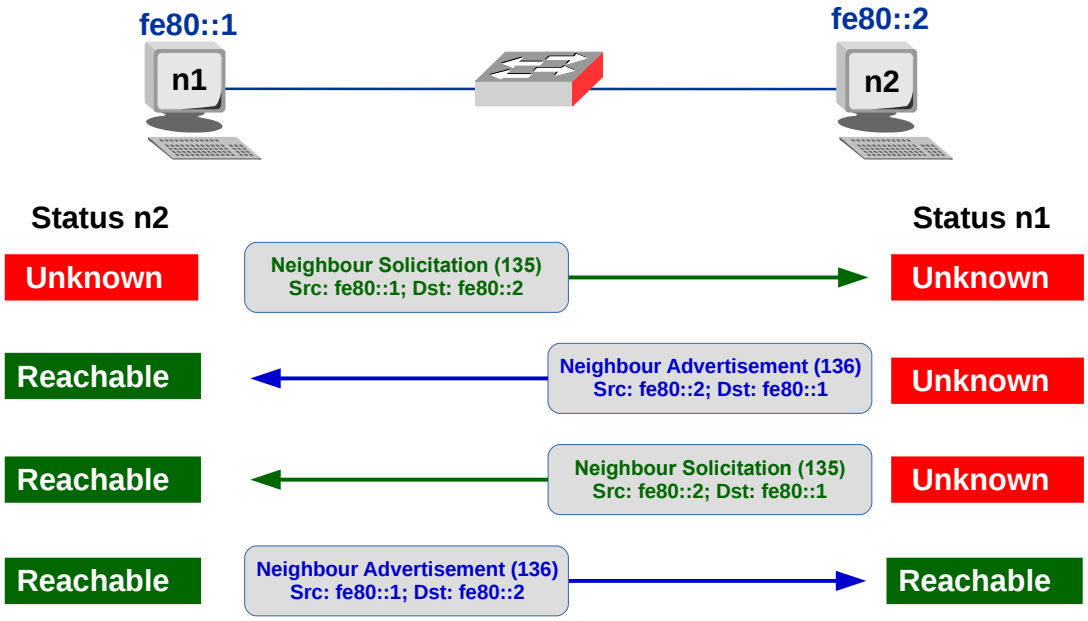


Illustration 19: Neighbour Unreachability Detection (NUD)

IPv6 NUD detection improves packet delivery in the presence of failing routers. This capability improves packet delivery over partially failing or partitioned links. This capability also improves packet delivery over nodes that change their link-local addresses. For example, mobile nodes can move off the local network without losing any connectivity because of stale ARP caches.

IPv4 has no corresponding method for NUD. Unlike ARP, ND detects half-link failures by using NUD and it therefore avoids sending traffic to neighbours when two-way connectivity is absent. NUD improves the robustness of packet delivery in the presence of failing routers or links, or mobile nodes.

Illustration 19 demonstrates the process, node *n1* sends a unicast NS (135) message to *n2* who responds with a unicast NA (136). Node *n2* then sends a unicast NS (135) message to node *n1* who responds with a unicast NA (136) of its own. In this was the two nodes confirm their established two-way connectivity.

9.2 ICMPv6 Redirect

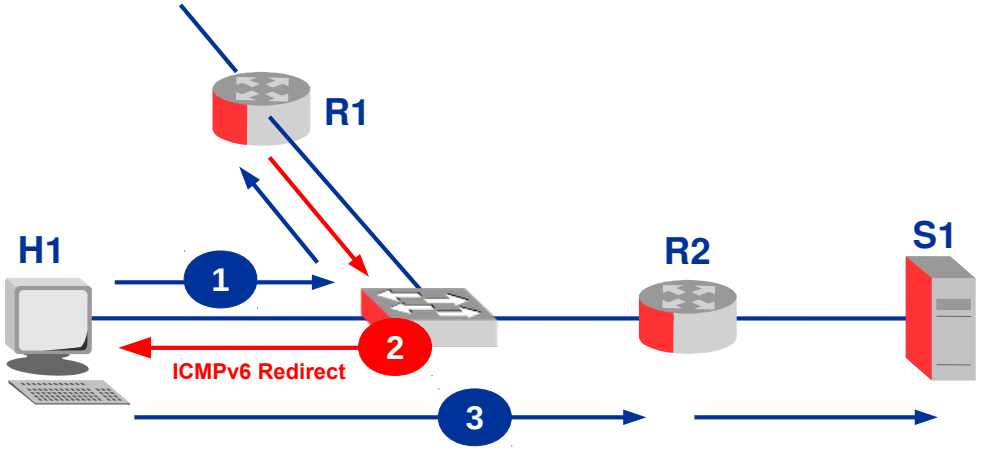


Illustration 20: ICMPv6 Redirect

On an IPv6 network, hosts don't know much about routes. They send their packets to a default gateway which handles the routing for them.

In the case of the local network having a single router, hosts will send all non-local traffic to the router. However if there is more than one local router, the host then must decide which router to use for which traffic. Generally the host does not know the most efficient choice of gateway router for packet it needs to send.

As demonstrated in Illustration 20 when a router receives datagrams destined for certain networks, it may realise that it would be more efficient if such traffic was sent by the host to a different router on the local network. In this case it can invoke the Redirect function by sending an ICMPv6 Redirect message to the device that sent the original packet. Redirect messages are always sent using the unicast address as the destination of the device that originally sent the packet that led to the Redirect being created.

10. IPv6 Configuration best practice – Inter-router links

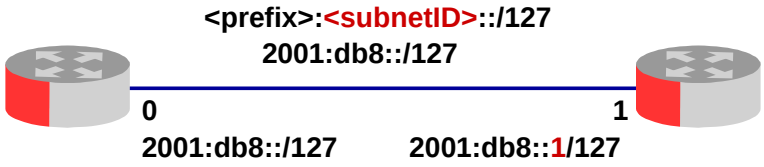


Illustration 21: Inter-router link

Best practice: use /127 for inter-router links. Addresses with the following 64 bits must NOT be used. By default routers will not send RS messages over links configured with /127 masks as they are unnecessary.

- 0000:0000:0000:0000
- ffff:ffff:ffff:ff7f ➠ :ffff

10.1 Using LLA on Inter-router links

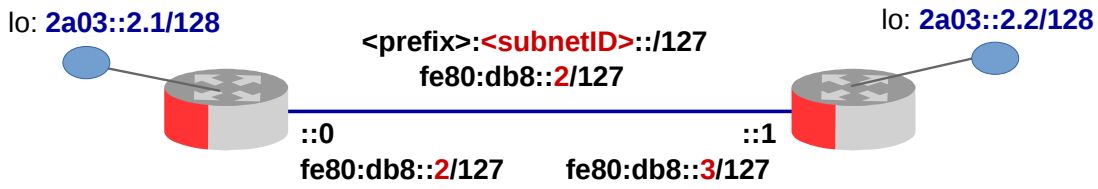


Illustration 22: Inter-router link with LLA

It is possible to use LLA on infrastructure links and it has advantages. However it is important to configure a GUA on a loopback address on each device for management plane traffic (Secure Shell (SSH), telnet, Simple Network Management Protocol (SNMP), etc). The source ICMPv6 error messages destined off-subnet.

The advantages of this approach are:

- Smaller routing tables which leads to:
  - less memory consumption
  - faster routing convergence
  - accelerated forwarding due to smaller Routing Information Bases (RIB) and Forwarding Information Bases (FIB).
- Simpler address management
- Lower configuration complexity
- Simpler DNS as LLAs are not put into zone files
- Reduced attack surface.

Caveats

- Router-interfaces not ping-able from off-link (fix: ping the loopback)
- Traceroutes to these interfaces break
- Hardware dependency – LLAs change if line cards change
- Network Management System (NMS) functions that are interface-address specific will break.

11. IPv6 Routing

11.1 Interior Gateway Routing

11.1.1 RIPng

Like its IPv4 variant *RIPng* is a Distance vector algorithm. RIPng messages use the UDP port 521 and the multicast address of FF02::9.

It has a number of implementations: GateD, MRTd, Kame, Quagga, Free Range Routing (FFR) as well as vendor equipment solutions from companies like Cisco, Juniper, HP, Huawei, MikroTik etc....

11.2 IS-IS Enhancements for IPv6

IS-IS in IPv6 functions the same and offers many of the same benefits as IS-IS in IPv4. IPv6 enhancements allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and Open Systems Interconnection model (OSI) routes. Extensions to IS-IS allow the configuration of IPv6-specific parameters. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to OSI and IPv4. IS-IS in IPv6 supports either single-topology mode or multiple topology mode.

11.2.1 OSPFv3

OSPFv3 is a Link State algorithm like OSPFv2, the IPv4 version. It is the recommended IGP of the IETF. The main differences from OSPFv2 are the removal of security as IPv6 has its own IPSec implementation embedded and the format of addresses are for IPv6.

It differs from OSPFv2 in a number of respects, including the following:

- Peering is carried out through link-local addresses
- The protocol is link based rather than network based
- Addressing semantics have been moved to leaf Link State Advertisements (LSA), which eventually allow its use for both IPv4 and IPv6.
- Point-to-point links are also supported in order to enable operation over tunnels.
- Authentication has been removed from OSPFv3 as IPv6 has underlying IPSec packet security.
- OSPFv3 is unable to set its own router ID like OSPFv2 does. Instead, it must be manually configured as a 32-bit value, same as in OSPFv2.
- OSPFv3 adds the concept of link-local flooding scope while retaining the domain (AS) and area flooding scopes of OSPFv2.

It is possible to enable OSPFv2 and OSPFv3 at the same time with OSPFv2 working with IPv4, and OSPFv3 working with IPv6.

LSA Type	LSA Code	OSPFv3 LSA	OSPFv2 LSA	Flooding Scope
1	0x2001	Router	Router	Area-local
2	0x2002	Network	Network	Area-local
3	0x2003	Inter-Area Prefix	Network Summary	Area-local
4	0x2004	Inter-Area Router	ASBR* Router	Area-local
5	0x4005	AS-external	AS-external	AS
6	0x2006	Group Membership	Group Membership	Not implemented
7	0x2007	Type-7 (NSSA**)	NSSA external	Area-local
8	0x0008	Link		Link-local
9	0x2009	Intra-Area Prefix		Area-local

* Autonomous System Boundary Router (ASBR)

** Not-So-Stubby Area (NSSA)

OSPFv3 has created a separation between prefixes and the Shortest Path First (SPF) tree. There is no prefix information in LSA type 1 and 2, topology adjacencies are only found in these LSAs and they do not contain any IPv6 prefixes. Prefixes are now advertised in type 9 LSAs and the LLAs that are used for next hops are advertised in type 8 LSAs. Type 8 LSAs are only flooded on the local link, type 9 LSAs are flooded within the area.

By separating the SPF tree and prefixes, OSPFv3 is more efficient. When the LLA on an interface changes, the router only has to flood an updated link LSA and intra-area-prefix LSA. Since there are no changes to the topology, there is no need to flood type 1 and 2 LSAs and other routers will not be required to run SPF in this case.

Link LSA

A router originates a separate Link LSA for each link it is attached to. These LSAs have link-local flooding scope and are never flooded beyond a link that they are associated with. These LSAs have three purposes:

- to notify routers attached to the link of the routers LLA
- to notify routers attached to the link of the list of IPv6 prefixes to associate with the link
- to allow the router to assert the collection of *Option* bits to associate with the Network LSA that will be originated for the link.

The Link-State ID is set to the Interface ID of link of the originating router.

Intra-Area Prefix LSA

A router uses Intra-Area Prefix LSA to advertise IPv6 prefixes that are associated with:

- the router itself
- an attached stub network segment
- an attached transit network segment.

Note: with OSPFv2 these are carried in the Router LSA.

A router can originate multiple Intra-Area Prefix LSAs for each router or transit network; each LSA is distinguished by its Link State ID.

Options field

A 24 bit *Options* field is included in Hello and Database Descriptor (DBD) packets, as well as in Router, Network and Inter-area Router LSAs. It enables OSPF routers to support optional capabilities, and to communicate their capabilities to other OSPF routers in the network.

Implementations: GateD, MRTd, Kame, Quagga, FFR and vendor hardware solutions from companies like Cisco, Juniper, HP, Huawei, MikroTik etc....

11.3 Exterior Gateway Routing

11.3.1 BGP4, MP-BGP

BGP4 is the standard inter domain routing protocol and it was extended to support IPv6 wiith RFC 2283 in 1998 as Multiprotocol BGP (MP-BGP). The protocol was extended with the concept of *address families*. An *address family* identifies a network protocol as well as an Address Family Identifier (AFI). Here are a subset of some typical values for AFI.

AFI	Description
0	Reserved
1	IP (IP version 4)
2	IP6 (IP version 6)
3	NSAP
4	HDLC (8-bit multidrop)
6	802 (Ethernet etc..)
16	DNS
18	AS Number

In addition to the AFI, BGP4 has a Subsequent Address Family Identifier (SAFI) to indicate the use of regular unicast routing, multicast routing in the Network Layer Reachability Information (NLRI), distributing Virtual Private Network (VPN) information, etc.

SAFI	Description
1	NLRI for unicast forwarding
2	NLRI for multicast forwarding
3	NLRI for unicast & multicast
4	NLRI with MPLS* labels.

* Multiprotocol Label Switching (MPLS)

For example instance, AFI 1 with SAFI 2 means BGPv4 will carry IPv4 multicast routing information, and AFI 2 with SAFI 1 specifies IPv6 unicast routing information is carried.

BGP routers that support IPv6 allow BGP sessions to be established using IPv6 addresses. MP-BGP speakers tell their neighbours which AFI+SAFI combinations they want to use in the OPEN message at the beginning of a BGP session. This can lead to the situation where IPv6 routing information is exchanged over IPv4, or IPv4 routing information is exchanged over IPv6. This can also lead to the complication of how a router knows which IPv6 next hop address to include in its updates towards an IPv4 neighbour. This can be avoided by exchanging IPv4 prefixes over an IPv4 external BGP (eBGP) sessions and only exchange IPv6 prefixes over an IPv6 eBGP sessions.

When it comes to internal BGP (iBGP) this problem doesn't exist as routers do not update the next hop address. As a result there are no problems exchanging both IPv4 and IPv6 prefixes over the same iBGP session. The only downside of this approach is that if there is an IPv4 failure, the IPv4 IBGP sessions go down and IPv6 is also affected. If IPv6 had its own iBGP sessions, it will possibly continue to operate independently of IPv4.

Implementations: GateD, MTRd, Kame, BGPd, Quagga, FRR and vendor hardware solutions from companies like Cisco, Juniper, HP, Huawei, MikroTik etc....

12. IPv6 transition mechanisms

Until IPv6 completely replaces IPv4, a number of transition mechanisms are needed to enable IPv6-only hosts to reach IPv4 services and to allow isolated IPv6 hosts and networks to reach the IPv6 Internet over the IPv4 infrastructure. As the IPv6 Internet grows larger, the need also arises for carrying IPv4 traffic over the IPv6 infrastructure.

There are a number of scenarios:

- Provider doesn't support IPv6
- Upstream doesn't support IPv6
- IPv6-only network
- MPLS-based network core
- IPv6-only services
- IPv6-only access network.

Additional scenarios:

- IPv4 Internet access with <1 public IP per subscriber
- Network needs more addresses than RFC1918 provides
- IPv4 Internet access from an IPv6-only network
- Access to private IPv4-only servers from IPv6 networks.

12.1 Categories of transition techniques

12.1.1 Dual Stack

IPv6 is a form of extension to IPv4 and therefore it is relatively easy to write a network stack that supports both IPv4 and IPv6 while sharing most of the code. Dual Stack is implemented by the various OS today. Some early experimental implementations used independent IPv4 and IPv6 stacks. There are no known implementations that implement IPv6 only. Actually when used in IPv4 communications, hybrid stacks tend to use an IPv6 API and represent IPv4 addresses in a special address format, the IPv4-mapped IPv6 address.

12.1.2 Tunnelling

This offers a mechanism for the encapsulation of one protocol within another.

12.1.3 Translation

When an IPv6 only host needs to access an IPv4 only host, or vice versa, translation is necessary. The one form of translation that actually works is the use of a dual stack application-layer proxy. Techniques for application agnostic translation at the lower layers have also been proposed, but they have been found to be too unreliable in practice due to the wide range of functionality required by common application-layer protocols, and are commonly considered to be obsolete.

12.2 Tunnelling

In order to reach the IPv6 Internet, an isolated host or network must be able to use the existing IPv4 infrastructure to carry IPv6 packets. This is achieved using a technique known as tunnelling which consists of the encapsulation of IPv6 packets within IPv4, in effect using IPv4 as a link layer for IPv6.

IPv6 packets can be encapsulated directly within IPv4 packets using protocol number 41. They can also be encapsulated within UDP packets e.g. in order to cross a router or NAT device that blocks protocol 41 traffic. Another options is to use generic encapsulation schemes like Generic Routing Encapsulation (GRE).

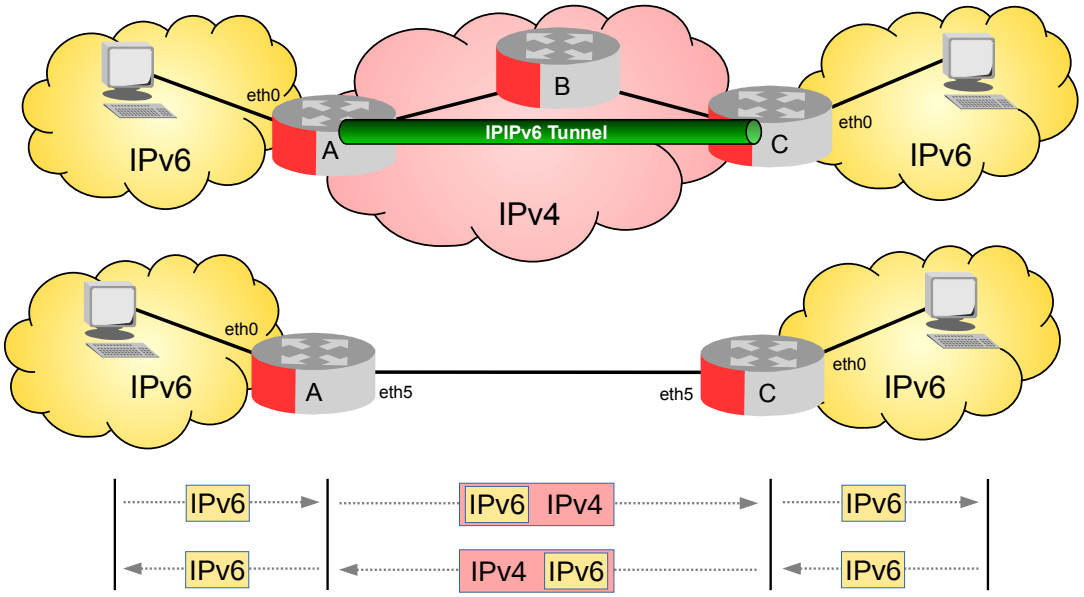


Illustration 23: Overlay tunnels for IPv6

12.2.1 Manual tunnelling

Manual tunnelling is achieved by configuring the end points of the tunnel. This tunnelling method can be used for sites with few nodes or for a limited number of remote connections. As is the case with static routing, scalability and management overhead are major issues limiting the use of manual tunnelling. There are a number of tunnel types:

- ***Router-router tunnel***: connect two IPv6 networks across an IPv4-only network or vice-versa.
- ***Host-router tunnel***: Connect IPv6 to an IPv6 host on an IPv4 network
- ***Host-host tunnel***: Connect IPv6 hosts over IPv4 networks.

Some potential problems for tunnelling.

- Fragmentation due to increased packet size
  - Not a problem for IPv4
  - IPv6 doesn't permit non-source fragmentation.
- Manual tunnels is not scalable
- Possibly suboptimal routing of IPv6 packets.

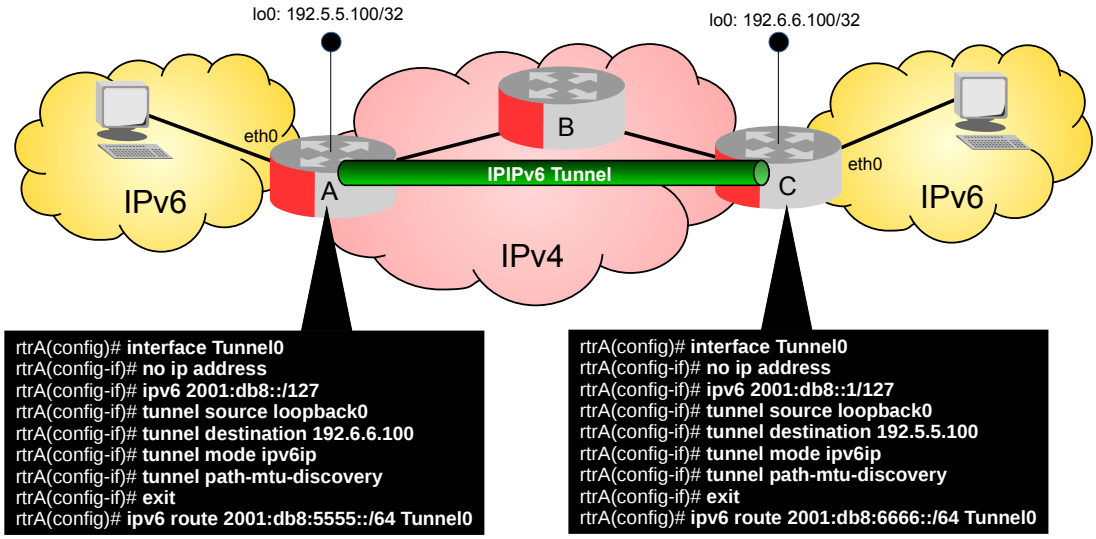


Illustration 24: Configure a manual tunnel

12.2.2 Automatic tunnelling

Automatic tunnelling refers to a technique where the tunnel endpoints are automatically determined by the routing infrastructure.

12.2.3 Tunnel Broker (TB)

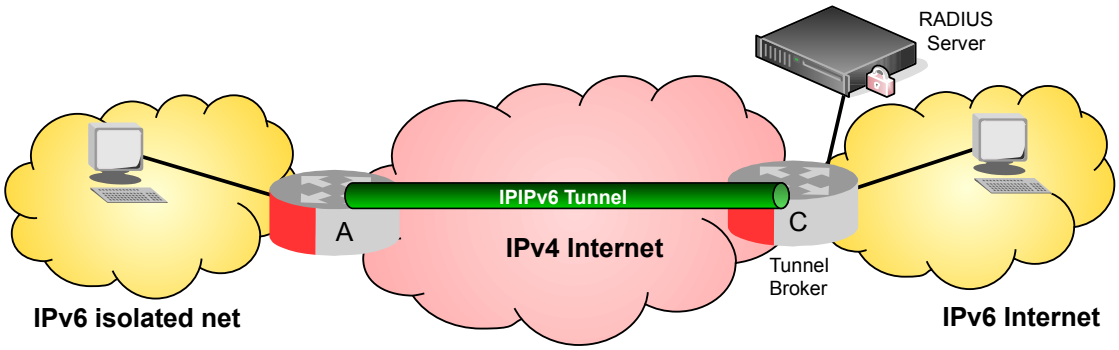


Illustration 25: Tunnel Broker (TB)

A Tunnel Broker (TB) allows isolated users/routers to connect to the IPv6 network and is defined in RFC 3053. The router or host establishes an IPv6 encapsulation over IPv4 to the TB who then authenticates the connection using RADIUS. If authorised, the router or host is assigned an IPv6 address and can now route IPv6. An example of a tunnel broker is the free service operated by Hurricane Electric[2].

12.2.4 Connection of IPv6 Domains via IPv4 Clouds (6to4)

6to4 is another technique for automatic tunnelling that uses protocol 41 encapsulation and is defined in RFC 3056. Tunnel endpoints are determined by using a well-known IPv4 *anycast* address on the remote side, and embedding IPv4 address information within IPv6 addresses on the local side.

2 https://tunnelbroker.net

6to4 performs three functions:

- Assigns a block of IPv6 address space to any host or network that has a global IPv4 address.
- Encapsulates IPv6 packets inside IPv4 packets for transmission over an IPv4 network using 6to4.
- Routes traffic between 6to4 and *native* IPv6 networks.

The key difference between automatic 6to4 tunnels and manually configured tunnels is the tunnel is not Point to Point (P2P) in 6to4; it uses Point to Multipoint (P2MP). Routers treat the IPv4 infrastructure as a virtual Non-Broadcast Multi-Access (NBMA) link. The IPv4 address embedded in the IPv6 address is used to find the other end of the automatic tunnel.

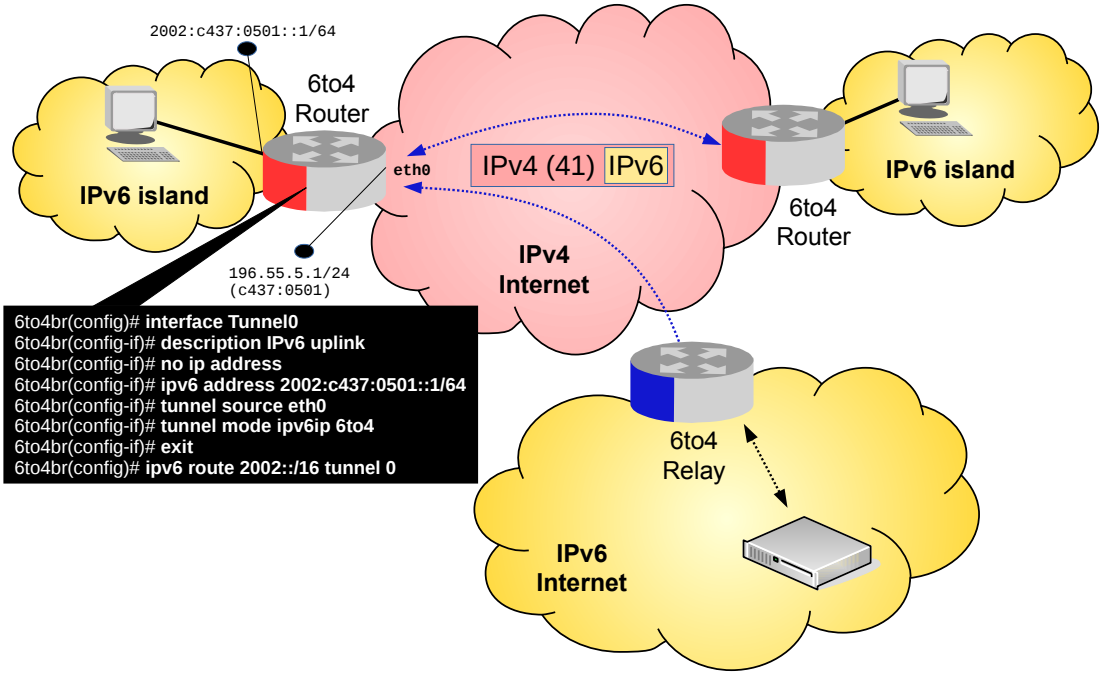


Illustration 26: 6to4

An automatic 6to4 tunnel may be configured on a border router in an isolated IPv6 network, which creates a tunnel on a per-packet basis to a border router in another IPv6 network over an IPv4 infrastructure. The tunnel destination is determined by the IPv4 address of the border router extracted from the IPv6 address that starts with the prefix:

2002::/16 → 2002:**<BR-IPv4-addr>**::/48 → 2002:**<BR-IPv4-addr>**<net#>::/64

196.55.5.1/24 → **c437:0501** → 2002:**c437:0501**::/64

Following the embedded IPv4 address are 16 bits (between /48 and /64) that can be used to number networks within the site. 6to4 tunnels can be configured between border routers or between a border router and a host. Appropriate entries in a DNS server that map between hostnames and IP addresses for both IPv4 and IPv6 allow the applications to choose the required address.

12.2.5 IPv6 Rapid Deployment (6rd)

IPv6 Rapid Deployment (6rd) is an extension of the 6to4 feature specified in RFC 5969. With the 6rd feature a provider can deliver a unicast IPv6 service to customers over its IPv4 network by using encapsulation of IPv6 in IPv4.

The main differences between 6rd and 6to4 tunnelling are as follows:

- 6rd does not require addresses to have a 2002::/16 prefix. Prefixes come from the providers own IPv6 block. From a customer perspective the IPv6 service provided is equivalent to native IPv6.
- All 32 bits of the IPv4 destination need not be carried in the IPv6 payload header. The IPv4 destination is obtained from a combination of bits in the payload header and information on the router. Furthermore, the IPv4 address is not at a fixed location in the IPv6 header as it is in 6to4.
- The 6rd delegated prefix is derived from the provider prefix and the IPv4 address bits, and is used by the Customer Premises Equipment (CPE) for hosts within its site.

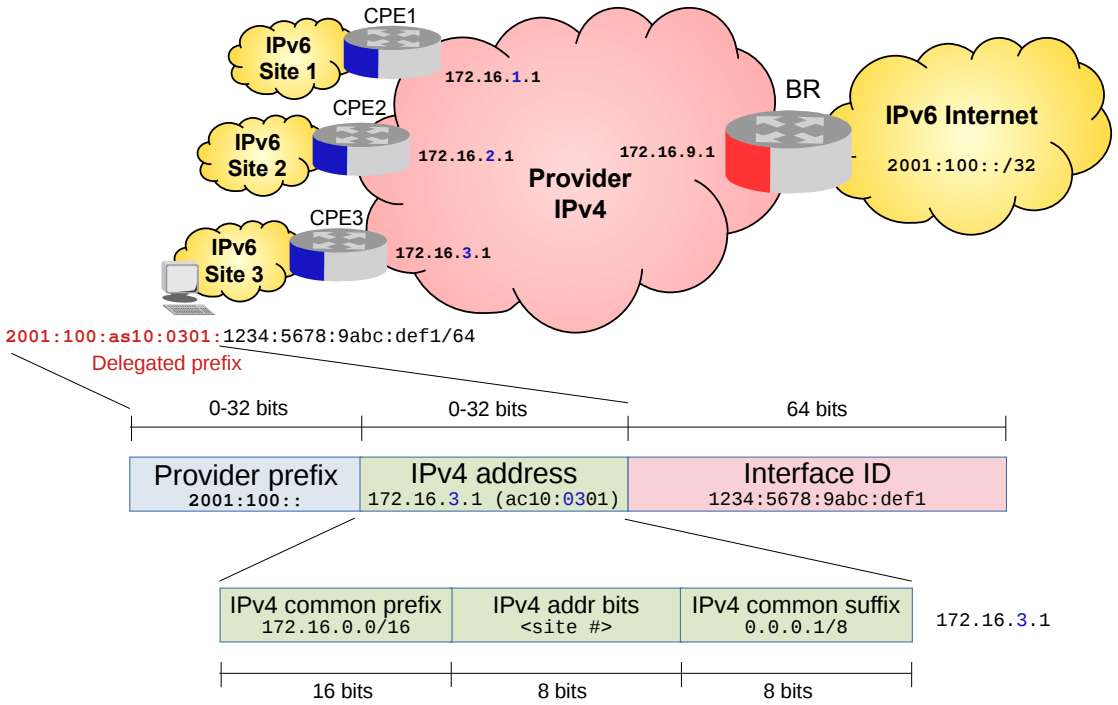


Illustration 27: IPv6 Rapid Deployment (6rd)

Taking the example 6rd example in Illustration 27, a tunnel and an IPv6 address is applied to it from the providers assigned block. The source of the tunnel is tied to the loopback interface associated with each site. The tunnel mode is defined as 6rd and the common IPv6 prefix is defined for the tunnels. Finally the prefix length and suffix length of the IPv4 transport address common to all the 6rd routers in a domain. The bits between the IPv4 prefix length and the suffix length (the third octet in this case) holds the site number. The following is an example for site 3.

```
6rd(config)# interface loopback 3
6rd(config-if)# ip address 10.3.3.3/32
6rd(config)# interface tunnel 3
6rd(config-if)# ipv6 address 2001:100::1/32
6rd(config-if)# tunnel source loopback 3
6rd(config-if)# tunnel mode ipv6ip 6rd
6rd(config-if)# tunnel 6rd prefix 2001:100:as10:0301::/64
6rd(config-if)# tunnel 6rd ipv4 prefix-length 16 suffix 8
```

12.2.6 Dual Stack Lite (DS-Lite)

DS-Lite technology described in RFC 6333 does not involve allocating an IPv4 address to the CPE for providing Internet access. It enables a broadband provider to share IPv4 addresses among customers by combining two well-known technologies IP in IP (IPv4-in-IPv6) and NAT.

Two elements:

- DS-Lite Basic Bridging BroadBand (B4)
- DS-Lite Address Family Transition Router (AFTR)
  (Also called Carrier Grade NAT (CGN) or Large Scale NAT (LSN))

The B4 distributes private IPv4 addresses for LAN clients, typically via DHCP. It encapsulates IPv4 packets within IPv6 packets, these are called *softwires*. The tunnel is a multipoint-to-point IPv4-in-IPv6 tunnel ending on the providers AFTR. There is a well-known range, 192.0.0.0/29 reserved for the IPv4 tunnel with 192.0.0.1 reserved for the AFTR element, and 192.0.0.2 is reserved for the B4 element. The original IPv4 packet is recovered at the AFTR and NAT is performed upon the IPv4 packet so it can be routed to the public IPv4 Internet based on the AFTR's global IPv4 address. AFTR uniquely identifies traffic flows by recording the B4 public IPv6 address, the private IPv4 address, and TCP or UDP port number as a session.

The best way to deliver necessary information to B4 is using DHCPv6. RFC6334 defined a new option called AFTR_NAME that conveys the Fully Qualified Domain Name (FQDN) to the AFTR. This allows operators to use a name that resolves to a different address for different clients, thus providing load balancing.

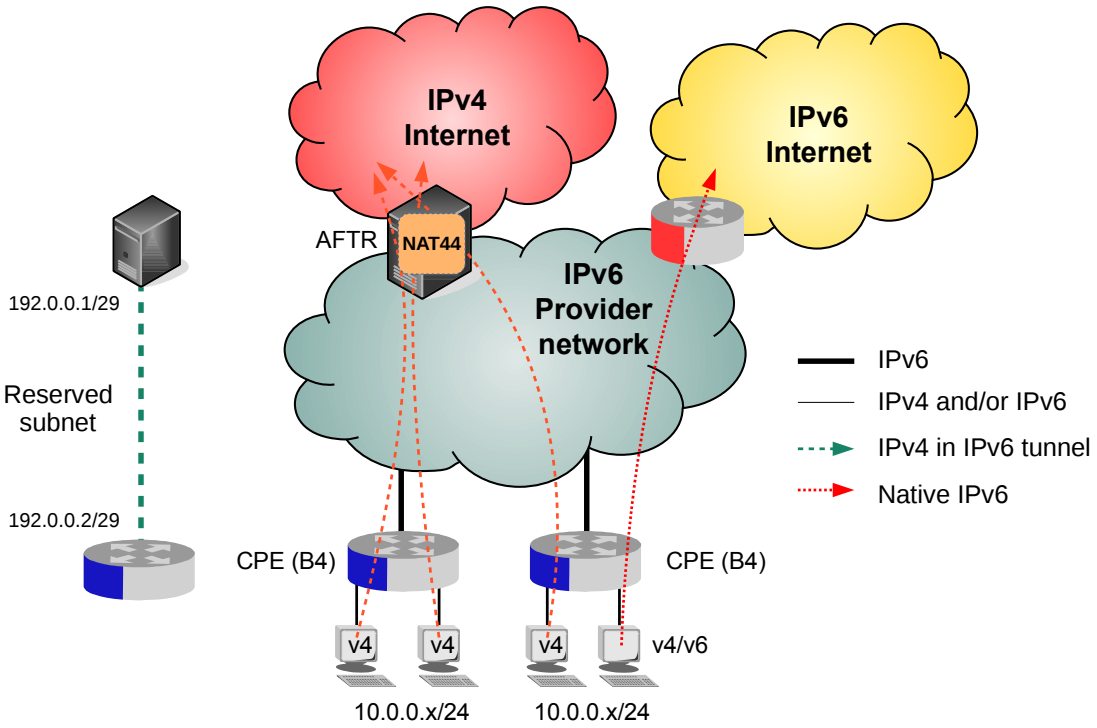


Illustration 28: DS-Lite

12.2.7 Lightweight 4over6 (lw4o6)

Lightweight 4over6 (lw4o6) extends DS-Lite by moving the NAT functionality from the ISP side to the CPE, eliminating the need to implement CGN. It is described in RFC 7596 and can be seen in Illustration 30. It is accomplished by allocating a port range for a shared IPv4 address to each CPE. Moving the NAT functionality to the CPE allows the ISP to reduce the amount of state tracked for each subscriber, which improves the scalability of the translation infrastructure.

There are three main components in the lw4o6 architecture:

- The lwB4, which performs the Network and Port Translation (NAPT) function and IPv4/IPv6 encapsulation/decapsulation
- The lwAFTR, which performs the IPv4/IPv6 encapsulation/decapsulation
- The Provisioning system, which tells the lwB4 which IPv4 address and port set to use.

lwB4

The lwB4 now performs the NAPT functionality and therefore must be provisioned with the public IPv4 address and a port set it is allowed to use. This information is provided through a provisioning mechanism such as DHCP, the Port Control Protocol (PCP) (RFC6887), or Technical Report 069 (TR-69). The lwB4 must be provisioned with:

- The IPv6 address for the lwAFTR
- The IPv4 external (public) address for NAPT44
- The restricted port set to use for NAPT44
- The IPv6 binding prefix.

The lwB4 MUST implement DHCPv6-based configuration using the OPTION_S46_CONT_LW.

This means that the lifetime of the *softwire* and the derived configuration information (e.g., IPv4 shared address, IPv4 address) are bound to the lifetime of the DHCPv6 lease.

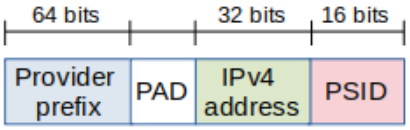


Illustration 29: IPv6 address for source of lw4o6 tunnel

lwB4 performs a longest-prefix match between the IPv6 binding prefix and its currently active IPv6 prefixes and forms the subnet to be used for the source of the lw4o6 tunnel. The full /128 address is then constructed in the same manner as demonstrated in Illustration 29. The Port Set ID (PSID) identifies the separate part of the transport-layer port space assigned to each lwB4.

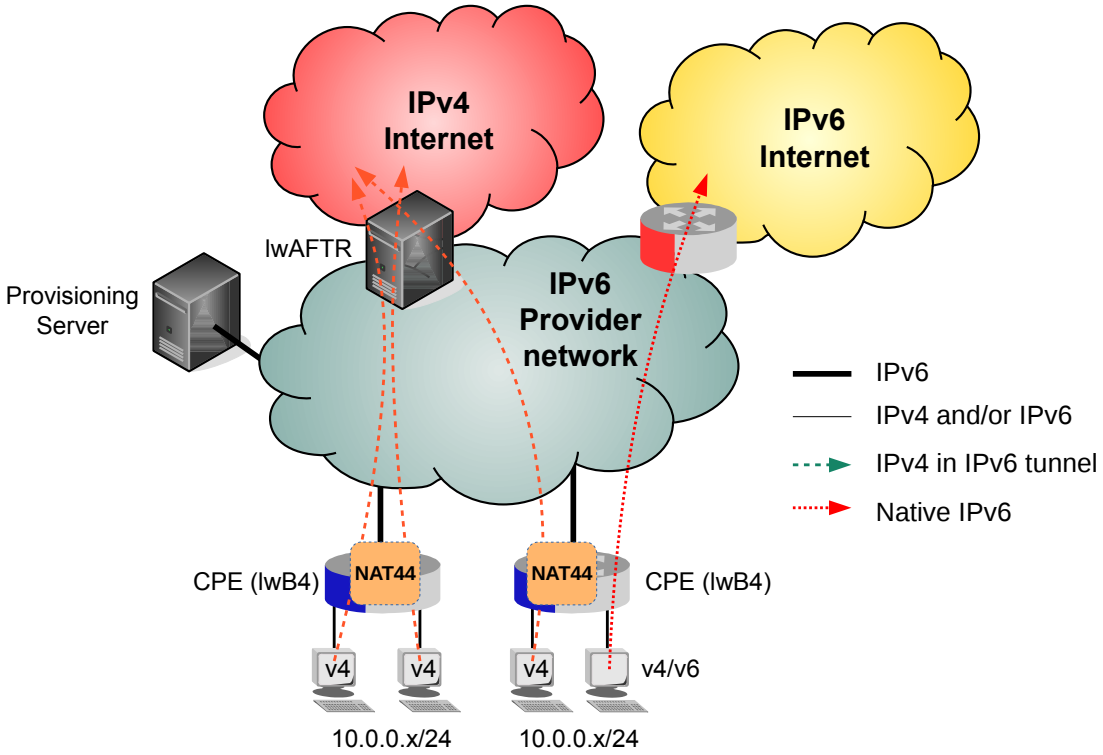


Illustration 30: lw4o6 system

lwAFTR

The lwAFTR needs to know the per-subscriber binding state between the IPv6 address of each subscriber as well as the IPv4 address and port set allocated to each subscriber. The lwAFTR maintains an address binding table containing:

- IPv6 address for a single lwB4
- Public IPv4 address
- Restricted port set.

The lwAFTR synchronises the binding information with the Provisioning server. The lifetime of binding table entries is synchronised with the lifetime of address allocations.

12.3 Translation mechanisms

12.3.1 NAT64 and DNS64

NAT64 is an IPv6 transition mechanism that facilitates communication between IPv6 and IPv4 hosts by using a form of NAT. The NAT64 gateway is a translator between IPv4 and IPv6 protocols. NAT64 requires at least one IPv4 address and an IPv6 network segment comprising a 32 bit address space to cater for the IPv4 network.

An IPv6 client embeds the IPv4 address it wishes to communicate with using the host part of the IPv6 network segment, resulting in an IPv4 embedded IPv6 addresses, and sends packets to the resulting address. The NAT64 gateway creates a mapping between the IPv6 and the IPv4 addresses, which may be manually configured or determined automatically.

Typically, NAT64 is designed to be used when the communication is initiated by IPv6 hosts. Some mechanisms, including static address mapping, exist to allow the inverse scenario.

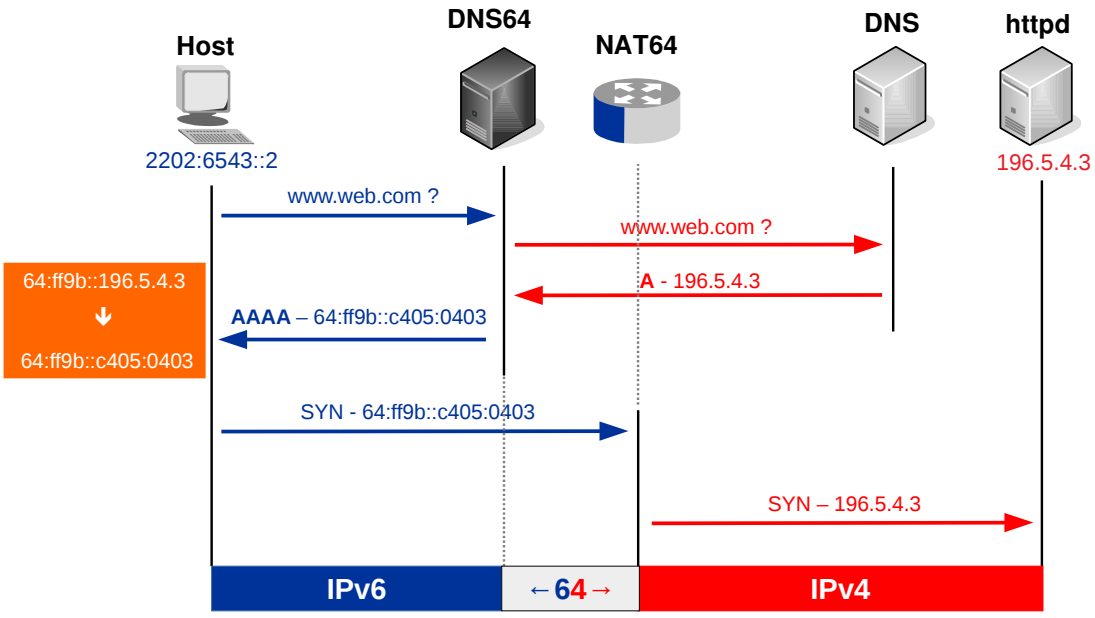


Illustration 31: NAT64 / DNS64

Illustration 31 demonstrates how an IPv6 host accesses an IPv4 service using NAT64/DNS64. The IPv4 host makes a IPv4 DNS request to the DNS64 server which forwards the request on the IPv4 network to the DNS server. The DNS server responds with an A record, 196.5.4.3 for the webserver. The DNS64 server translates the IPv4 address by combining it with the NAT64 prefix 64:ff9b::/96 to form an IPv6 address 64:ff9b::c405:0403 and forwards this as an AAAA record to the requesting host.

The host forms a packet and sends it to the NAT64 router, it being its gateway which translates the IPv6 packet header to an IPv4 packet header changing the IP address to the IPv4 format. It continues to translate the packet headers in each direction for the duration of the connection.

12.3.2 XLAT / 464LAT

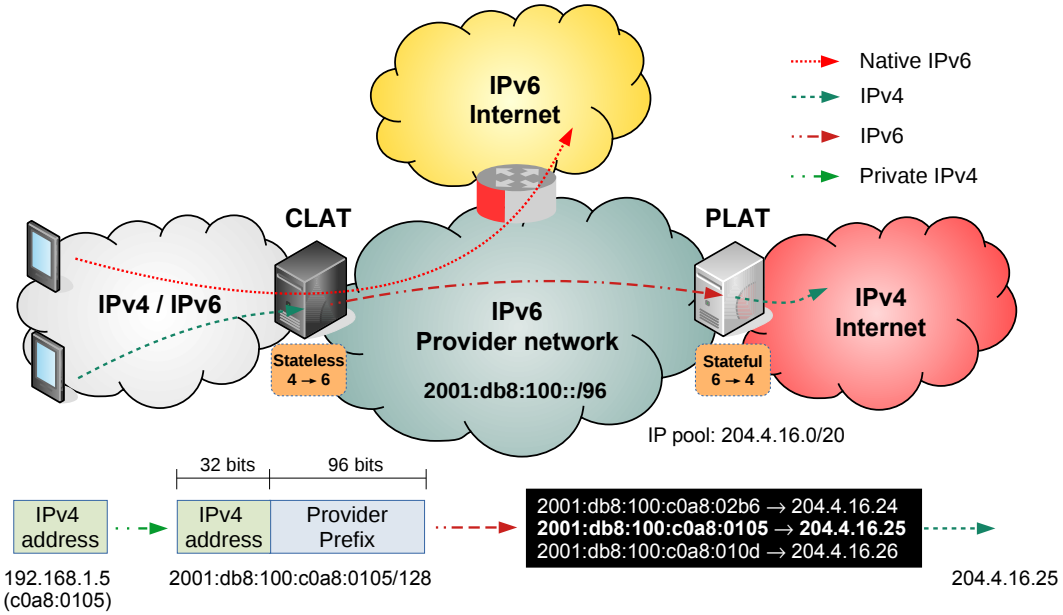


Illustration 32: xLAT / 464LAT system

464XLAT is a hub and spoke architecture focused on enabling IPv4-only services over IPv6-only networks. It is defined in RFC 6877. It has two elements:

- **Customer-side transLATor (CLAT)**: translates 1:1 private IPv4 addresses to global IPv6 addresses
- **Provider-side transLATor (PLAT)**: translates N:1 global IPv6 addresses to global IPv4 addresses.

Referring to the Illustration 32 a dual stack device accessing a native IPv6 service does so using IPv6 routing. In the case of an IPv6 device requiring connection to an IPv4 service then there is only a requirement for stateful NAT to be performed at the PLAT; however, in the case of an IPv4 device requiring connection to an IPv4 service, the CLAT provides it with a private IPv4 address. The CLAT translates this private IPv4 address into an IPv6 address by appending the IPv4 address to the provider IPv6 prefix using Stateless IP/ICMP Translation (SIIT). This is then forwarded to the PLAT. The PLAT maintains a pool of public IPv4 addresses and performs NAT between the IPv6 address and a public IPv4 address thereby providing access to the IPv4 public service.

These are three traffic treatment scenarios are summarised in the table.

Application and host	Server	Traffic Treatment	Location of Translation
IPv6	IPv6	End-to-End IPv6	None
IPv6	IPv4	Stateful Translation	PLAT
IPv4	IPv4	464XLAT	CLAT/PLAT