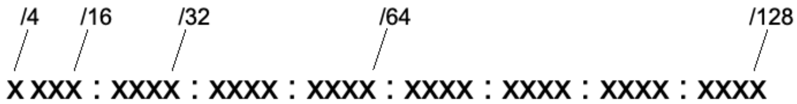
IPv6 Overview

- IPv6 core protocol definition is RFC 2460
- RIPng, EIGRPv6, OSPFv3, ICMPv6, traceroute6, ping6, MP-BGP-4 (mulitprotocol BGP v4)
- ARP is replaced with Neighbor discovery, Neighbor solicitation/ duplicated address detection, multiprotocol router discovery and router advertisement.

IPv6 Numbering Summary:
Each character is 4 bits, in hexadecimal, so every block of 4 (a quartet) of those is 16 bits. Typically, IPv6 addresses refer to a certain length or place using the backslash notation. This diagram shows the structure of an IPv6 address to demonstrate bit placement/ordering 0-128 bits.



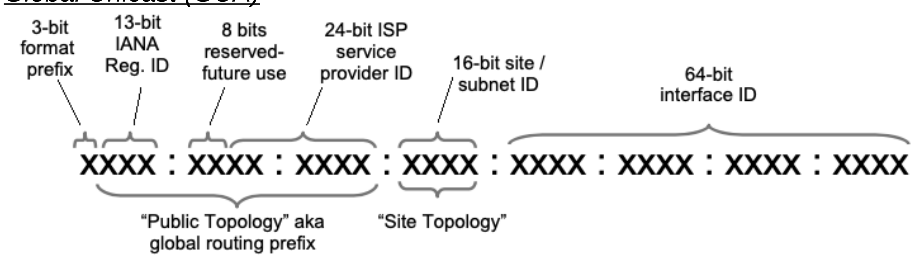
So, a prefix ending at the /48th bit, we can say something like 2001:1234:5678::/48 (how many bits are used)

You'll see IPv6 addresses shortened often. Here's how that works:
Start with expanded address: 2001:0DB8:0000:0000:0000:FF00:0042:8329
- Remove any leading zeros - All leading zeros within each group of 4 hexadecimal digits can go. So, 0DB8 becomes DB8, 0000 becomes 0, and 0042 becomes 42 - 2001:DB8:0:0:0:FF00:42:8329
- Replace consecutive groups of zeros with double colons (::). You can only do this once per address, so do it to the largest grouping or the leftmost if there are a few the same size. Now we have 2001:DB8::FF00:42:8329
- This is more a style thing instead of shortening: make all letters lowercase. It helps with both ease of reading, but also with case-sensitive searching or filtering.   We finally get: 2001:db8::ff00:42:8329
- Expanding is easy: inverse the rules and pad out so there are 8 quartets of 4 bits each.

Address Types

If asked what the 3 types of IPv6 addresses there typically the expected answer is unicast, multicast and anycast. It's most efficient to start of with unicast, of which there are also 3 types, one routable on the internet and the other two that have a local scope. It is also the one that has the most components so it makes the others easier.

Global Unicast (GUA)



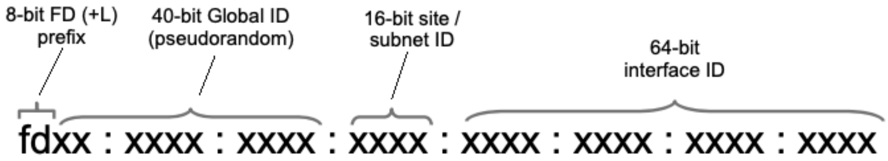
So, using the global unicast address as a starting point, take 2001:0db8:56a4:0001:0000:0000:0000:0002 - or - 2001:db8:56a4:1::2 In this example, the generalized prefix would be 2001:0db8:56a4::/48, the subnet would be 0001, and interface ID (simplified) 2

Global unicast addresses are routable and what you need to use to do anything internet. The other two can't leave your local nets. The also can have a variety of prefixes (the other 2 have the same ones and are easy to spot). As of Feb 2023, IANA lists 2001:0000::/23 – 2c00:0000::/12 as being currently registered/available global unicast prefix blocks ffrom regional registries (RIRs). For more info, see https://www.iana.org/assignments/ipv6-unicast-address-assignments.

If your company was only given one /64, you will likely have a subnet ID assigned by the ISP along with the rest of the prefix illustrated above. The interface ID represents a single interface on any given device. Typically you shouldn't need to worry about altering anything in the prefix your ISP gave you including the subnet ID. You may have been given a larger address space to work with like /32 or /48. There is a section on subnetting those after talking about addresses and such.

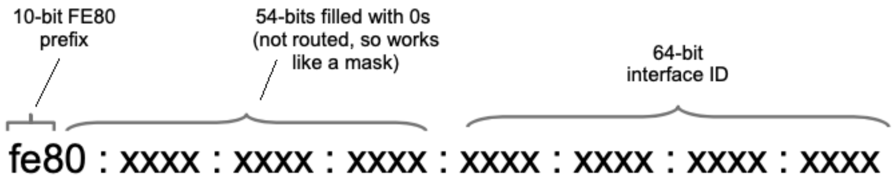
2001:1234:3333::/48 -Company B
2001:1234:3333:0001::/64 -Company B's subnet 1
2001:1234:3333:0002::/64 -Company B's subnet 2
2001:1234:3333:2::1 -A host in Company B's subnet 2

Unique Local Addresses (ULA) Unicast



Unique Locals are like the reserved IPv4 addresses for private networks (192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8). They aren't routable over the internet (we'll talk about NAT later) but they can move around the local network, just one notch down from being “internet-ready”. Generally these will start with the prefix FD. They replace an old one named “site local”. These don't use the IANA/ISP's prefix like global unicast, and instead is a “random” replacement (still often called a global ID... out of habit?). This can be autogenerated by the device or done manually, which will be discussed soon.

Link Local (Unicast), fe80::/10



Link locals are auto-generated and only useful inside one subnet- can't be routed. They mainly help with the functions of neighbor discovery and next hop configuration. A great way to check out all 3 of these unicast types is to run ifconfig on your workstation.

Multicast address ff00::/8
Identifies a group of nodes or interfaces with traffic forwarded to all the nodes in the group. Addresses are all assigned out of the FF00::/8 block; also have a scope associated: link local being just like the scope of unicast LL, 'organization' with same scope as unique local, and global.

Anycast address
Identifies a group of nodes or interfaces, with traffic forwarded to the nearest node in the group. An anycast address is essentially a unicast address assigned to multiple devices with a host ID = 0000:0000:0000:0000. (Anycast addresses are not widely used today.)

Key IPv6 Multicast Addresses

ff01::1	All nodes in the interface-local	
ff02::1	All nodes in the link-local, RA/RS	
ff0X::1	All nodes address, identify the group of all IPv6 nodes	
ff0X::2	All routers	1 (interface-local), 2 (link-local), 5 (site-local)
ff01::2	All routers in the interface-local	
ff02::2	All routers in the link-local	
ff05::2	All routers in the site-local	
ff02::5	OSPFv3	2 (link-local)
ff02::6	OSPFv3 Designated Routers	2 (link-local)
ff02::9	RIP Routers	2 (link-local)
ff02::a	EIGRPv6 Routers	2 (link-local)
ff02::d	All PIM Routers	2 (link-local)
ff02::1a	All RPL Routers	2 (link-local)
ff0X::fb	mDNSv6	All scopes
ff0X::101	All NTP servers	All scopes
ff02::1:1	Link Name	2 (link-local)
ff02::1:2	All-dhcp-agents	2 (link-local)
ff02::1:3	Link-local Multicast Name Resolution	2 (link-local)
ff05::1:3	All-dhcp-servers	5 (site-local)
ff02::1:ff00:0/104	Solicited-node multicast address	2 (link-local)
ff02::2:ff00:0/104	Node Information Queries	2 (link-local)

Host Addressing: SLAAC and EUI-64 (RFC 2373)
Stateless Address Auto-Configuration (SLAAC) enables hosts to generate a unique **routable** address on their own. For this the router sends out a Router Advertisement (RA) periodically, and a host can send a Router Solicitation(RS) in order to trigger an RA. The RA has the prefix and length to use, default gateway. Before using the address it makes, the host uses Duplicate Address Detection (DAD) to make sure it's unique. Since things like DNS servers aren't in RAs, DHCPv6 still has to be used. (DAD takes the place of ARP in IPv6)

For the interface ID, the host either uses a EUI-64 (Extended Unique Identifier) made from the interface's MAC address (most likely), or can optionally generate it randomly. The EUI-64 creation works like this:

1. Split the 6-byte (12-hex digit) MAC address in two down the middle.
2. Insert FFFE in between the two halves.
3. Invert the seventh bit from the beginning.

Examples:
Interface MAC address is aa12:bcbc:1234
10101010 represents the first 8 bits of the MAC address (aa) - when inverting the 7th bit it becomes 10101000.
The answer is A8 and the EUI-64 address: 2001:0db8:0:1:a812:bcff:febc:1234 EUI-64

MAC address 0b34:ba12:1234
0b in binary is 00001011, the first 8 bits of the MAC address, which then becomes 00001001
The answer is 09, and the IPv6 EUI-64 address: 2001:0db8:0:1:0934:baff:fe12:1234 EUI-64

Generating EUI-64 is usually done for us:
ipv6 address 2001:db8:1111:1::/64 eui-64

```
interface GigabitEthernet0/0
  ipv6 address 2001:db8:1111:1::/64 eui-64
interface serial0/0/0
  ipv6 address 2001:db8:1111:2::/64 eui-64
show ipv6 interface brief
    GigabitEthernet0/0 [up/up]
       fe80::1ff:fe01:101                <---link local, employing eui64
       2001:db8:1111:1:0:1ff:fe01:101    <---EUI64 global unicast address
    Serial0/0 [up/up]
       fe80::1ff:fe01:101
       2001:db8:1111:2:0:1ff:fe01:101
```

Cisco router, setting up SLAAC

```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-routing
Router(config)#interface interface
Router(config-if)#ipv6 address ipv6-address/prefix-length
Router(config-if)#no shutdown
```

Problems with EUI-64
It has been pointed out that a user cannot connect anonymously to any network if someone knows the EUI-64 interface identifier of that device, which could be exploited such as websites and apps associating different IPv6 addresses to a particular device or user (whether malicious, monitoring, or benign). It is recommended that if the host OS or router does not support autoconfig with Random Interface Identifiers, that static IPv6 address should be used. (MS Windows generates by default a random interface ID for SLAAC)

Best answer to how to make clients go random instead of using EUI-64:
https://superuser.com/questions/243669/how-to-avoid-exposing-my-mac-address-when-using-ipv6

Stateless and Stateful DHCPv6 in Cisco IOS
DHCPv6 Stateless mode
- provides network info not in an RA, (no IPv6 address since already provided by SLAAC).
- DNS domain name and server(s), other DHCP options.

```
ipv6 unicast-routing
ipv6 dhcp pool IPV6_DHCPPOOL
address prefix 2001:db8:5:10::/64
domain-name cisco.com
dns-server 2001:db8:6:6::1
interface Vlan20
description IPv6-DHCP-Stateless
ip address 192.168.20.1 255.255.255.0
ipv6 nd other-config-flag
ipv6 dhcp server IPV6_DHCPPOOL
ipv6 address 2001:DB8:0:20::1/64
```

DHCPv6 Stateful aka managed mode assigns unique addresses instead of the client generating one

```
ipv6 unicast-routing
ipv6 dhcp pool IPV6_DHCPPOOL
address prefix 2001:db8:5:10::/64
domain-name cisco.com
dns-server 2001:db8:6:6::1
interface Vlan20
description IPv6-DHCP-Stateful
ip address 192.168.20.1 255.255.255.0
ipv6 address 2001:DB8:0:20::1/64
ipv6 nd prefix 2001:DB8:0:20::/64 no-advertise
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 dhcp server IPV6_DHCPPOOL
```

This interface configuration is for a Cisco IOS IPv6 router implementing stateful DHCPv6 on an external DHCP server:

```
ipv6 unicast-routing
domain-name cisco.com
dns-server 2001:db8:6:6::1
interface Vlan20
description IPv6-DHCP-Stateful
ip address 192.168.20.1 255.255.255.0
ipv6 address 2001:DB8:0:20::1/64
ipv6 nd prefix 2001:DB8:0:20::/64 no-advertise
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 dhcp_relay destination 2001:DB8:0:20::2
```

Cisco IOS Routing in IPv6

Static routing in IPv6

```
ipv6 unicast-routing
interface serial0/0/0
  ipv6 address 2001:5432:1111:4::1/64
interface serial0/0/1
  ipv6 address 2001:db8:1111:5::1/64
interface gigabitethernet0/0
  ipv6 address 2001:db8:1111:1::1/64
```

Static IPv6 with Next-Hop Address

```
!First command is on R1, listing R2's global unicast
R1(config)#ipv6 route 2001:db8:1111:2::/64 2001:db8:1111:4::2
!This command is on R2, listed R1's global unicast
R2(config)#ipv6 route 2001:db8:1111:1::/64 2001:db8:1111:4::1
!Verify routes with show ipv6 route static
```

Default Route

```
B1(config)#ipv6 route ::0 S0/0/1
```

```
show ipv6 route
show ipv6 route static
show ipv6 route local
```

OSPFv3 Routing - PID and area goes on interfaces

```
ipv6 unicast-routing
interface serial0/0/0
  no ip address
  ipv6 address 2001:abcd:1234:4::1/64
interface s0/0/1
  no ip address
  ipv6 address 2001: abcd 1234:5::1/64
interface GigabitEthernet0/0
  no ip address
  ipv6 address 2001:abcd:1234:5::1/64
ipv6 router ospf 1
     router-id 1.1.1.1
```

```
interface s0/0/0
    ipv6 ospf 1 area 0
interface s0/0/1
    ipv6 ospf 1 area 0
int gi0/0
    ipv6 ospf 1 area 0

ipv6 unicast-routing
ipv6 router ospf 2
  router-id 2.2.2.2
int s0/0/1
  ipv6 address 2001: abcd:1234:4::2
  ipv6 ospf 2 area 0
int gi0/0
  ipv6 address 2001:abcd:1234:2::2
  ipv6 ospf 2 area 0

show ipv6 ospf
show ipv6 protocols
show ipv6 ospf interface
show ipv6 ospf interface brief
show ipv6 ospf neighbor
show ipv6 ospf database
show ipv6 route ospf
```

Subnetting IPv6

In IPv4, subnetting is done to get more addresses for hosts,optimize network distribution to help with that.
In IPv6 we have tons of addresses, so it becomes more of an issue of things like network organization.
Bulk blocks of IPv6 space purchased from RIRs are bought in lengths of /32 - /48 that can be subnetted into multiple /64 subnets. Subnets sized over /64 (like /84 or /96) are not advised since IPv6 features like addressing autoconfiguration rely on being used in a /64 subnet.

Prefix	Appearance	# of hosts	Number of /64s
/32	xxxx:xxxx::	2^{96}	4294967296
/36	xxxx:xxxx:x::	2^{92}	268435456
/40	xxxx:xxxx:xx::	2^{88}	16777216
/44	xxxx:xxxx:xxx::	2^{84}	1048576
/48	xxxx:xxxx:xxxx::	2^{80}	65536
/52	xxxx:xxxx:xxxx:x::	2^{76}	4096
/56	xxxx:xxxx:xxxx:xx::	2^{72}	256
/60	xxxx:xxxx:xxxx:xxx::	2^{68}	16
/64	xxxx:xxxx:xxxx:xxxx::	2^{64}	1

A /64 holds 18,446,744,073,709,551,616 addresses