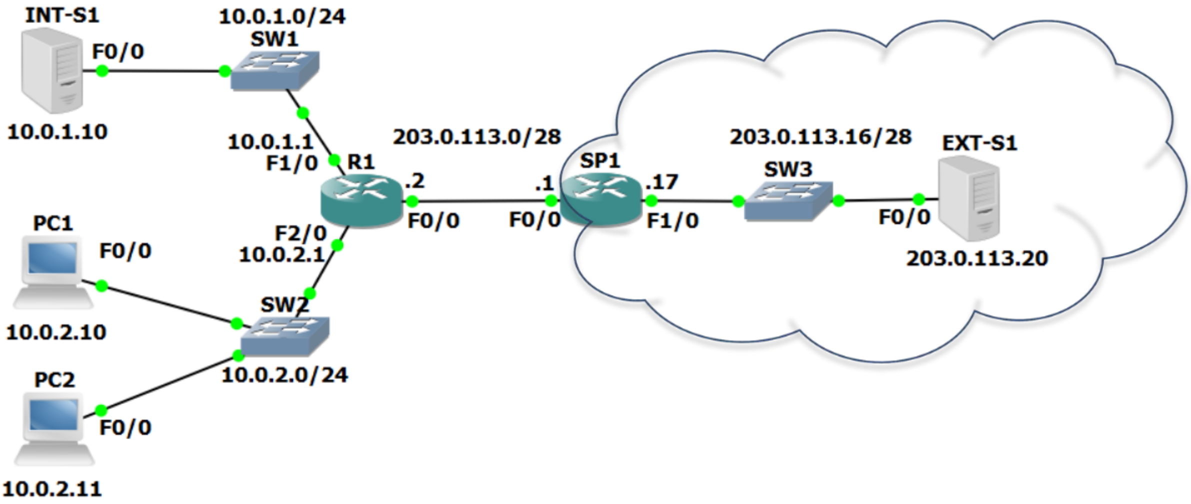
NAT Types

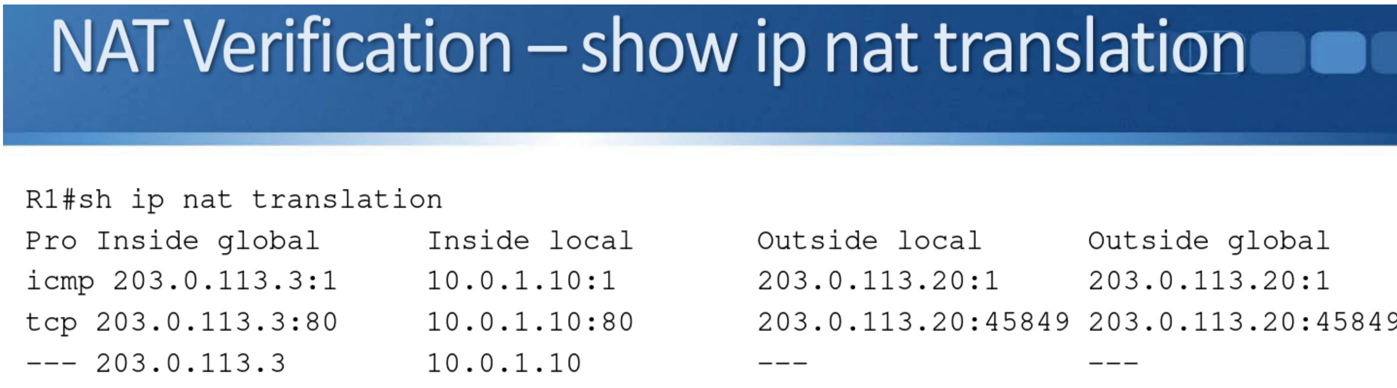
- **Static NAT** – permanent one-to-one mapping usually between a public and private IP address. Used for servers which must accept incoming connections.
- **Dynamic NAT** – uses a pool of public addresses which are given out on an as needed first come first served basis. Usually used for internal hosts which need to connect to the Internet but do not accept incoming connections.
- **PAT** (Port Address Translation) – allows the same IP address to be reused.

Static NAT

- We have bought the range of public IP addresses 203.0.113.0/28 from our service provider
- 203.0.113.2 is used on the outside interface on our Internet edge router R1
- 203.0.113.1 is used as the default gateway address. It is the SP1 router on the other side of the link
- 203.0.113.3 – 203.0.113.14 remain available



```
R1(config)#int f0/0
R1(config-if)#ip nat outside
R1(config)#int f1/0
R1(config-if)#ip nat inside
R1(config)#ip nat inside source static 10.0.1.10 203.0.113.3
```



NAT Verification – show ip nat translation

```
R1#sh ip nat translation
Pro Inside global       Inside local        Outside local       Outside global
icmp 203.0.113.3:1      10.0.1.10:1         203.0.113.20:1      203.0.113.20:1
tcp 203.0.113.3:80      10.0.1.10:80        203.0.113.20:45849 203.0.113.20:45849
--- 203.0.113.3         10.0.1.10           ---                 ---
```

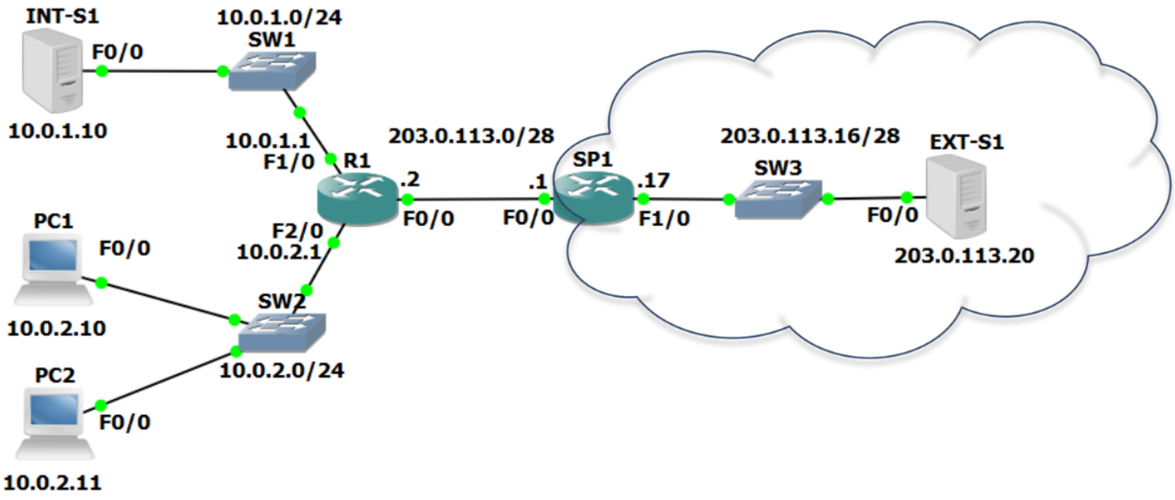
NAT Definitions

- Inside local address—The IP address actually configured on the inside host's Operating System.
- Inside global address— The NAT'd address of the inside host as it will be reached by the outside network.
- Outside local address—The IP address of the outside host as it appears to the inside network.
- Outside global address—The IP address assigned to the host on the outside network by the host's owner.

Outside Local vs Outside Global

- Router R1 in our example knows one address to reach the outside host (203.0.113.20) and does not translate that address.
- For one way NAT, the Outside Local and Outside Global addresses will be reported as being the same.

Dynamic NAT



- The hosts in the 10.0.2.0/24 network do not accept incoming connections so they don’t need a fixed public IP address with a static NAT translation
- They do need outbound connectivity to the Internet so need to be translated to a public IP address
- We will use the remaining public addresses 203.0.113.4 - 14 as a NAT pool
- The inside hosts will be translated to the public IP addresses on a first come first served basis when they send traffic out
- The first host to send traffic out will be translated to 203.0.113.4, the second host to 203.0.113.5 etc., up to 203.0.113.14 at the end of the pool

- With standard dynamic NAT you need a public IP address for every inside host which needs to communicate with the outside
- If you have 30 hosts, you need 30 public IP addresses
- When all the addresses in the pool have been used, new outbound connections from other inside hosts will fail because there will be no addresses left to translate them to
- These hosts would have to wait for existing connections to be torn down and the translations to be released back into the pool when they time out

Dynamic NAT Configuration

```
R1(config)#int f0/0
R1(config-if)#ip nat outside
R1(config)#int f2/0
R1(config-if)#ip nat inside
```

Configure the pool of global addresses.

```
R1(config)#ip nat pool Flackbox 203.0.113.4 203.0.113.14 netmask 255.255.255.240
```

Create an access list which references the internal IP addresses we want to translate.

```
R1(config)#access-list 1 permit 10.0.2.0 0.0.0.255
```

Associate the access list with the NAT pool to complete the configuration.

```
R1(config)#ip nat inside source list 1 pool Flackbox
```

- R1#clear ip nat translation can be used to remove translations from the translation table
- This can be useful when troubleshooting
- It is also often required if you want to edit your NAT configuration – the router will not allow changes when there are active translations
- `clear ip nat translation *` will remove all dynamic translations

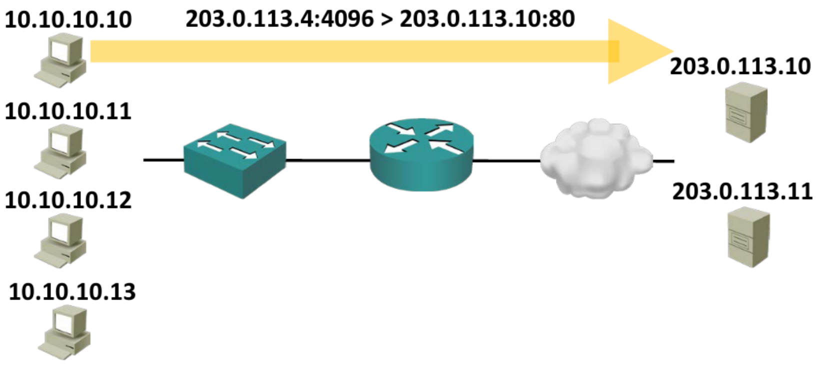
R1#show ip nat statistics

PAT

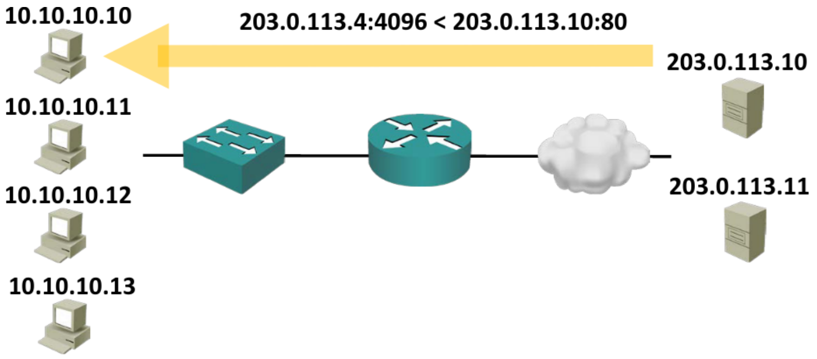
- Port Address Translation (PAT) is an extension to NAT that permits multiple devices to be mapped to a single public IP address
- With PAT you do not need a public IP address for every inside host
- The router tracks translations by IP address and Layer 4 port number
- Because different inside hosts are assigned different port numbers, the router knows which host to send return traffic to, even when the public IP address is the same

- **Dynamic NAT with Overload**(is a type of PAT) uses PAT to allow more clients to be translated than IP addresses are available in the NAT pool
- If the NAT pool is 203.0.113.4 to 203.0.113.6 for example, the first 2 hosts which initiate outbound connections will be translated to 203.0.113.4 and 203.0.113.5

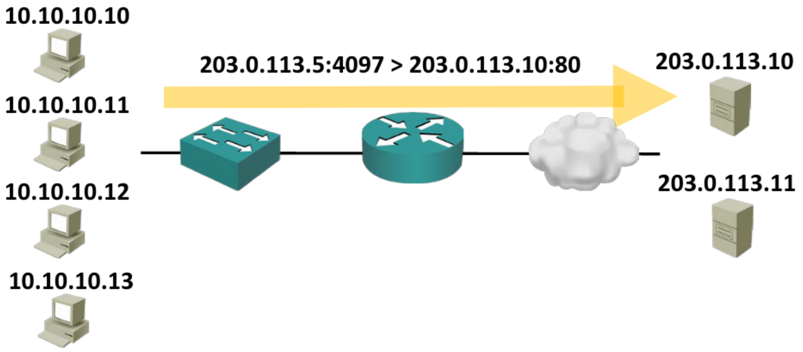
- The 3rd host will be translated to 203.0.113.6 and the router will track which source port number was used in the translation table
- The 4th and 5th etc. hosts will also be translated to 203.0.113.6 but with different source port numbers
- When the return traffic is sent back the router checks the destination port number to see which host to forward it to



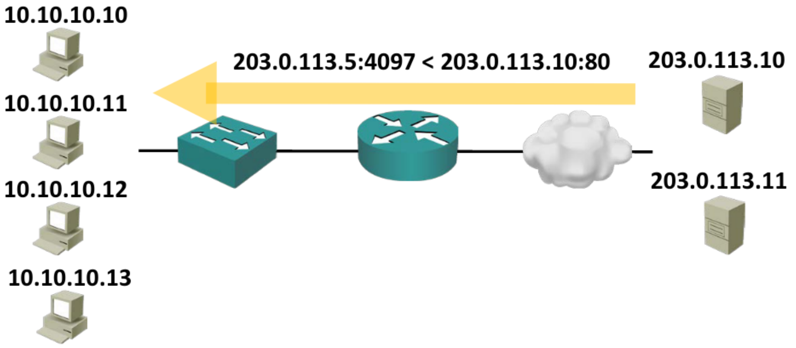
Inside Local Source	Inside Global Source	Outside Local Destination	Outside Global Destination
10.10.10.10:49165	203.0.113.4:4096	203.0.113.10:80	203.0.113.10:80



Inside Local Source	Inside Global Source	Outside Local Destination	Outside Global Destination
10.10.10.10:49165	203.0.113.4:4096	203.0.113.10:80	203.0.113.10:80



Inside Local Source	Inside Global Source	Outside Local Destination	Outside Global Destination
10.10.10.11:49158	203.0.113.5:4097	203.0.113.10:80	203.0.113.10:80



Inside Local Source	Inside Global Source	Outside Local Destination	Outside Global Destination
10.10.10.11:49158	203.0.113.5:4097	203.0.113.10:80	203.0.113.10:80

Dynamic NAT with Overload Configuration

```
R1(config)#int f0/0
R1(config-if)#ip nat outside
R1(config)#int f2/0
R1(config-if)#ip nat inside
```

Configure the pool of global addresses.

```
R1(config)#ip nat pool Flackbox 203.0.113.4 203.0.113.6 netmask 255.255.255.240
```

Create an access list which references the internal IP addresses we want to translate.

```
R1(config)#access-list 1 permit 10.0.2.0 0.0.0.255
```

Associate the access list with the NAT pool to complete the configuration.

```
R1(config)#ip nat inside source list 1 pool Flackbox overload
```

PAT with Single IP Address

- The last NAT scenario to cover is a small office which has not purchased a range of public IP addresses
- In this case the outside interface will most likely get its IP address via DHCP from the service provider
- PAT can be used to allow multiple inside hosts to share the single outside public IP address
- The configuration is very similar to Dynamic NAT with Overload but translates to the outside interface address rather than a pool of addresses
- You must translate to the outside interface rather than a specific IP address because a DHCP address can change

```
R1(config)#int f0/0
R1(config-if)#ip address dhcp
R1(config-if)#ip nat outside

R1(config)#int f1/0
R1(config-if)#ip nat inside

R1(config)#access-list 1 permit 10.0.2.0 0.0.0.255
R1(config)#ip nat inside source list 1 interface f0/0 overload
```

IPv6

- IPv6 uses a 128 bit address compared to IPv4's 32 bit address
- The address is written as X:X:X:X:X:X:X:X
- Each 'X' is a 16 bit hexadecimal field (hex values are 0-9,A-F)
- Eg. 2001:0DB8:0000:0001:0000:0000:0000:0001
- Each segment is 16 bits but there isn't an official name for them ('hexadectet' is too hard to pronounce)
- They are sometimes called 'hextets', 'pieces' or 'quartets'

Address Shortening

- The IPv6 address is very long. There are a couple of ways we can shorten it to make things more convenient
- Address shortening is a standard convention and supported by all vendor's devices
- Leading zeros in each field can be removed
- 2001:0DB8:0000:0001:0000:0000:0000:0001 can be written as 2001:DB8:0:1:0:0:0:1

- Successive all zero fields can be shortened to '::'
- 2001:0DB8:0000:0001:0000:0000:0000:0001 can be written as 2001:DB8:0:1:0:0:0:1 (leading zeros removed)
- And 2001:DB8:0:1:0:0:0:1 can be written as 2001:DB8:0:1::1

- Successive all zero fields can be shortened only once in an address to avoid confusion
- 2001:0:0:1:0:0:0:B can be shortened to
- 2001::1:0:0:0:B or
- 2001:0:0:1::B
- It can't be shortened to 2001::1::B

IPv6 Address Types

1. Global Unicast
2. Unique Local
3. Link Local

Global Unicast

- Global Unicast Addresses are similar to IPv4 public addresses
- They are assigned to an individual host and have global reachability (unless blocked by security policy such as on a firewall)
- They are assigned from the range 2000::/3

- Internet authorities assign blocks from the overall 2000::/3 range to organisations
- A common assignment for a company is a /48 block, eg 2001:10:10::/48
- A smaller or larger size block can be assigned depending on the size of the company

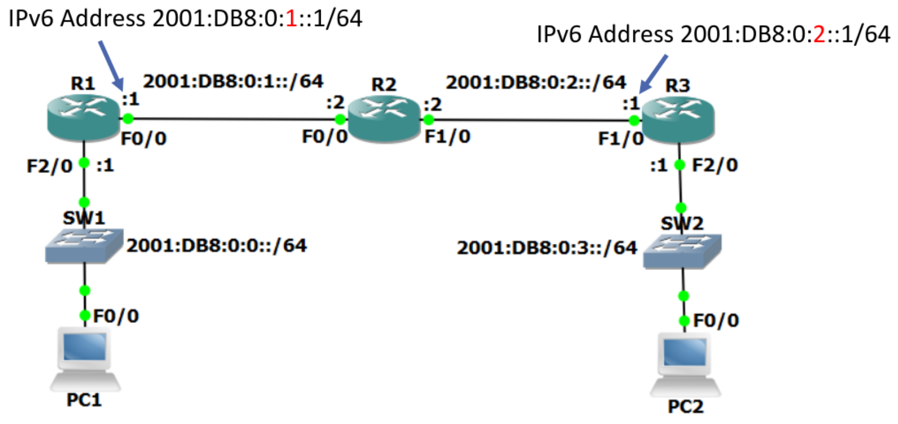
- IPv6 standards state that addresses assigned to individual hosts should use a /64 mask
- The IPv6 address is 128 bits so /64 splits it in half for the network and host portions of the address

- X:X:X:X : X:X:X:X
  Network : Host

Example:

- If a company is assigned a /48 address by the Internet authorities and uses /64 host addresses, that leaves 16 bits the company can assign to its internal subnets
- For example, if the company was assigned 2001:10:10::/48 by the Internet authorities, it can assign subnets 2001:10:10:0::/64 to 2001:10:10:FFFF::/64 to its internal network segments
- 16 bits = 65,535 possible subnets
- 64 bits left over = 18,446,744,073,709,551,616 hosts per subnet

- In this example the company has been assigned 2001:DB8:0::/48 by the Internet authorities



- Using a /64 for all network subnets including point-to-point links and loopback addresses can seem wasteful, but the official declaration is that the IPv6 address space is so large that it does not create a problem
- Using /64 everywhere simplifies the addressing and enables the use of EUI-64 addresses

Enable IPv6 routing first

```
R1(config)#ipv6 unicast-routing
R1(config-if)#int f0/0
R1(config-if)#ipv6 add 2001:db8:0:1::1/64
R1(config-if)#int f2/0
R1(config-if)#ipv6 add 2001:db8:0:0::1/64
R1#sh ipv6 interface brief
```

Broadcast and Multicast

- IPv4 supports broadcast to all hosts on 255.255.255.255
- Routers do not forward broadcast traffic so this stays on the local subnet
- IPv6 does not support broadcast traffic
- It does however support multicast to all hosts on the local subnet (ff02::1) which is functionally equivalent
- Many services which use broadcast to 255.255.255.255 in IPv4 use more specific multicast addresses in IPv6 (eg ff05::1:3 for all DHCP servers)

EUI-64 Addresses

- A Cisco router can generate full IPv6 addresses for itself when given the interface and /64 network to use
- The host portion of the address is derived from the interface's MAC address, which is guaranteed to be globally unique
- A MAC address is a /48 address compared to the /64 host portion of the IPv6 address
- FF:FE is injected in the middle of the /48 MAC address to bring it up to 64 bits. Also, the 7th bit is inverte

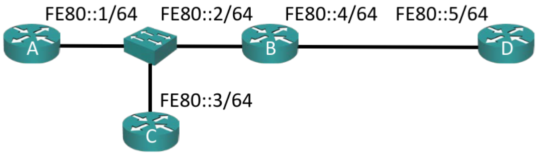
```
R1(config)#int f0/0
R1(config-if)#ipv6 address 2001:db8:0:1::/64 eui-64
R1(config)#int f2/0
R1(config-if)#ipv6 address 2001:db8:0::/64 eui-64
```

Unique Local

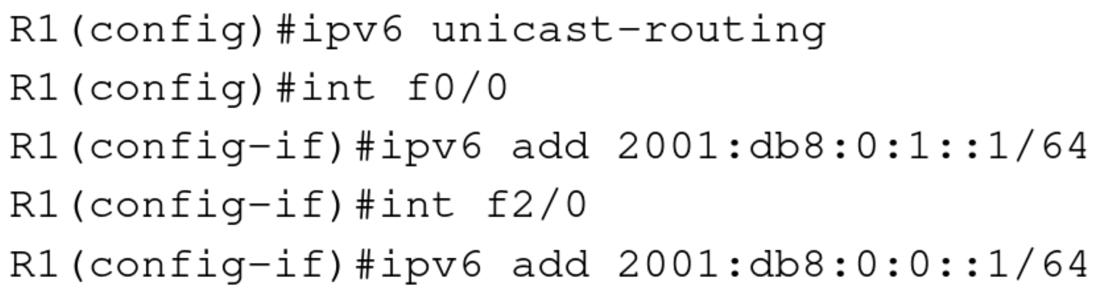
- Unique Local Addresses are similar to IPv4 RFC 1918 private addresses
- They are not publicly reachable
- They are assigned from the range FC00::/7
- Hosts should be assigned /64 addresses

Link Local

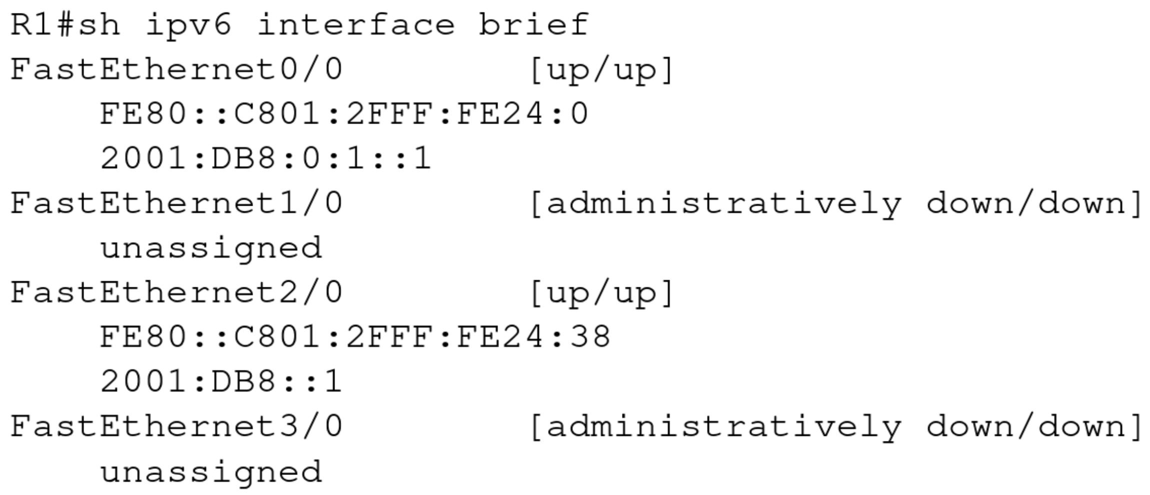
- Link local addresses are valid for communications on that link only
- They are assigned from the range FE80::/10 – FEB0::/10
- Hosts should be assigned /64 addresses
- A, B and C have connectivity to each other via the FE80::1, FE80::2 and FE80::3 link local addresses on the same segment
- B and D have connectivity to each other via the FE80::4 and FE80::5 link local addresses on the same segment
- FE80::1, FE80::2 and FE80::3 do not have connectivity to FE80::4 or FE80::5



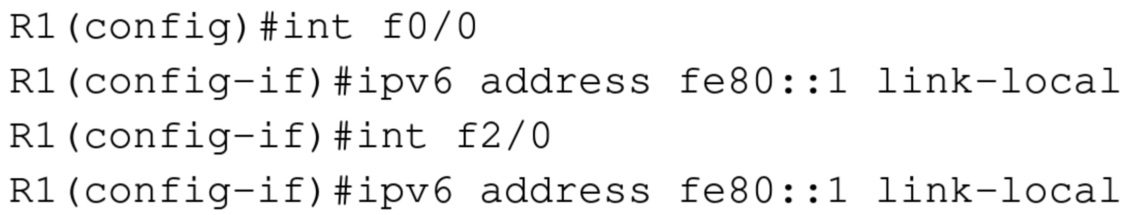
- Link local addresses can be used for communications which should not be forwarded beyond the local link, like routing protocol hello packets and updates
- They are mandatory on IPv6 enabled Cisco router interfaces



```
R1(config)#ipv6 unicast-routing
R1(config)#int f0/0
R1(config-if)#ipv6 add 2001:db8:0:1::1/64
R1(config-if)#int f2/0
R1(config-if)#ipv6 add 2001:db8:0:0::1/64
```



```
R1#sh ipv6 interface brief
FastEthernet0/0        [up/up]
    FE80::C801:2FFF:FE24:0
    2001:DB8:0:1::1
FastEthernet1/0        [administratively down/down]
    unassigned
FastEthernet2/0        [up/up]
    FE80::C801:2FFF:FE24:38
    2001:DB8::1
FastEthernet3/0        [administratively down/down]
    unassigned
```



```
R1(config)#int f0/0
R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#int f2/0
R1(config-if)#ipv6 address fe80::1 link-local
```

Multiple IPv4 Addresses

R1(config)#int f0/0
R1(config-if)#ip address 10.10.10.1 255.255.255.0
R1(config-if)#ip address 192.168.10.1 255.255.255.0

R1#sh run int f0/0

```
interface FastEthernet0/0
ip address 192.168.10.1 255.255.255.0
```

R1(config)#int f0/0
R1(config-if)#ip address 172.16.0.1 255.255.255.0 secondary
R1#sh run int f0/0

```
interface FastEthernet0/0
ip address 172.16.0.1 255.255.255.0 secondary
ip address 192.168.10.1 255.255.255.0
```

Multiple IPv6 Addresses

R1(config)#int f0/0
R1(config-if)#ipv6 address FE80::1 link-local
R1(config-if)#ipv6 add 2001:db8:0:0::1/64
R1(config-if)#ipv6 add 2001:db8:0:1::1/64

R1#sh run int f0/0

```
interface FastEthernet0/0
ip address 172.16.0.1 255.255.255.0 secondary
ip address 192.168.10.1 255.255.255.0
ipv6 address FE80::1 link-local
ipv6 address 2001:DB8::1/64
ipv6 address 2001:DB8:0:1::1/64
```

- Link local addresses are mandatory on IPv6 enabled interfaces
- Global unicast and Unique local addresses are optional
- You can have multiple addresses on the same interface
- One link local address for routing protocol traffic and one global unicast address for normal routing is typical

(SLAAC)-Stateless Address AutoConfiguration

- Hosts can be assigned IPv6 addresses through static addressing, DHCPv6, or SLAAC
- DHCP servers track their MAC address to IP address assignments, so this is ‘stateful’ addressing

- With SLAAC, hosts learn the /64 subnet which their interface is on from their local router and then use this information to generate their own IPv6 EUI-64 address
- (Modern Operating Systems randomise the host portion of the address rather than using standard EUI-64 for privacy reasons)
- The router does not track which hosts have which IP address so this is ‘stateless’ addressing

SLAAC – Router Advertisements

- When a global unicast IPv6 address is configured on an interface then Router Advertisements advertising the network prefix are sent out by default
- These ICMP messages are sent to the ‘All Nodes’ multicast address from the interface’s link-local address
- Hosts can also send a ‘Router Solicitation’ message to request the information

- As well as telling the hosts which subnet to generate their IP address on, the router tells the hosts to use itself as their default gateway
- **The original implementation did not support any information other than the default gateway address**

- In practice a DHCP server is still required to give out information such as DNS server
- If the IP address is assigned by SLAAC and the DNS server is assigned by DHCP this results in a stateless configuration, where the DHCP server does not retain information about the hosts

The Unspecified Address

- :: is the Unspecified address or Unknown address
- An IPv6 route to ::/0 is a default route equivalent to 0.0.0.0 0.0.0.0 in IPv4
- Also, :: is used as the source when an interface is trying to acquire an address

Neighbor Discovery

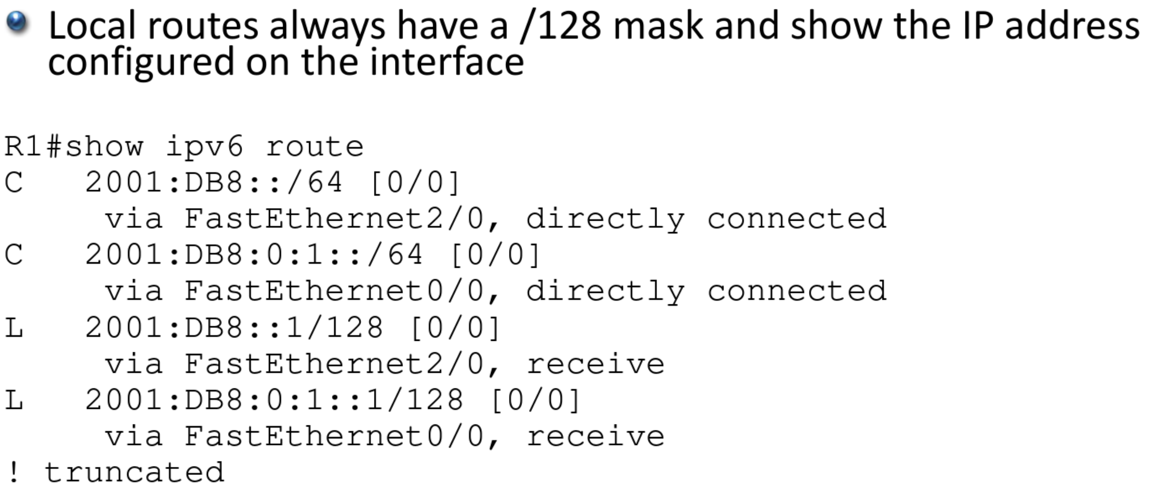
- Neighbor Discovery is the IPv6 version of ARP and works in the same way
- Rather than using ARP requests and replies, Neighbor Discovery uses ICMP Neighbor Solicitations and Neighbor Advertisements
- Neighbor Solicitation messages are sent to the Solicited-Node multicast address which reaches all hosts on the subnet

IPv6 Routing

- IPv6 routing works the same way as IPv4 routing, but the processes are separate, and there are separate IPv4 and IPv6 routing tables
- If a router receives an IPv4 packet, it will route it according to its IPv4 routing table
- If a router receives an IPv6 packet, it will route it according to its IPv6 routing table
- The routing tables are built in the same way, through static routes or dynamic routing protocols

- Updated versions of the existing IPv4 routing protocols were released to support IPv6.
- The configuration and operation is very similar for IPv6 as for IPv4.
    - RIPng (RIP next generation)
    - EIGRP for IPv6
    - OSPFv3
    - IS-IS
    - MP-BGP4 (MultiProtocol BGP-4)

- IPv4 routing is enabled by default on a Cisco IOS router
- IPv6 routing is disabled by default
- Enter the command ‘ipv6 unicast-routing’ to enable it
- You can still configure IPv6 addresses on a router without ipv6 unicast-routing enabled and send and receive IPv6 traffic, but the router will not forward IPv6 traffic to other networks

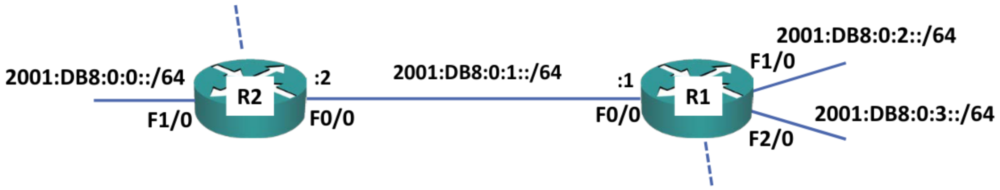


- Local routes always have a /128 mask and show the IP address configured on the interface

```
R1#show ipv6 route
C   2001:DB8::/64 [0/0]
     via FastEthernet2/0, directly connected
C   2001:DB8:0:1::/64 [0/0]
     via FastEthernet0/0, directly connected
L   2001:DB8::1/128 [0/0]
     via FastEthernet2/0, receive
L   2001:DB8:0:1::1/128 [0/0]
     via FastEthernet0/0, receive
! truncated
```

IPv6 Static Routes

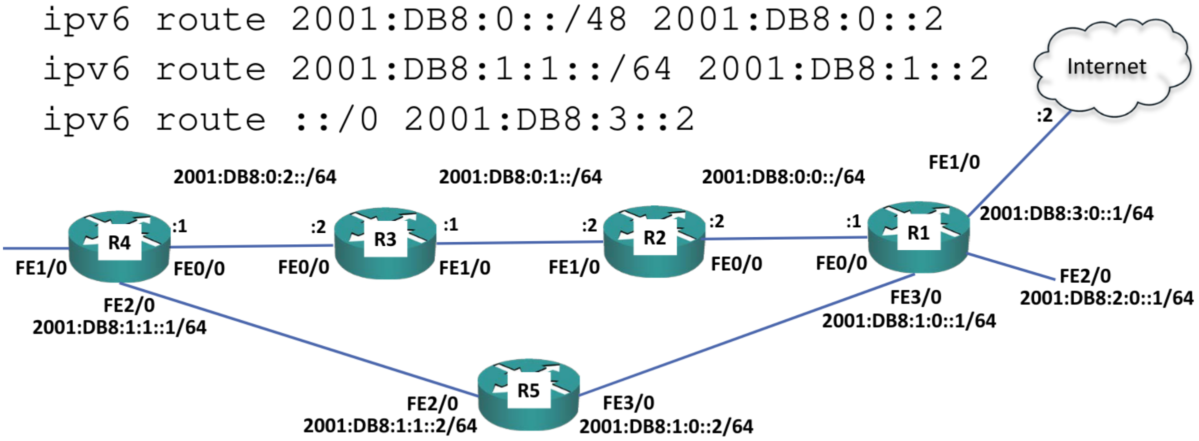
```
ipv6 route 2001:DB8:0:2::/64 2001:DB8:0:1::1
ipv6 route 2001:DB8:0:3::/64 2001:DB8:0:1::1
```



```
ipv6 route 2001:DB8::/64 2001:DB8:0:1::2
```

IPv6 Summary and Default Route

```
ipv6 route 2001:DB8:0::/48 2001:DB8:0::2
ipv6 route 2001:DB8:1:1::/64 2001:DB8:1::2
ipv6 route ::/0 2001:DB8:3::2
```



Internet
:2
FE1/0
2001:DB8:0:2::/64
2001:DB8:0:1::/64
2001:DB8:0:0::/64
2001:DB8:3:0::1/64
R4
:1
:2
R3
:1
:2
R2
:2
:1
R1
FE1/0
FE0/0
FE0/0
FE1/0
FE1/0
FE0/0
FE0/0
FE2/0
2001:DB8:2:0::1/64
FE2/0
2001:DB8:1:1::1/64
FE3/0
2001:DB8:1:0::1/64
R5
FE2/0
2001:DB8:1:1::2/64
FE3/0
2001:DB8:1:0::2/64