



IPv6 Integration & Transition

ISP/IXP Workshops

IETF Working Groups

- “v6ops”

Define the processes by which networks can be transitioned from IPv4 to IPv6

<http://www.ietf.org/dyn/wg/charter/v6ops-charter.html>

- “behave”

Designs solutions for the IPv4 to IPv6 translations scenarios

<http://www.ietf.org/dyn/wg/charter/behave-charter.html>

- “softwires”

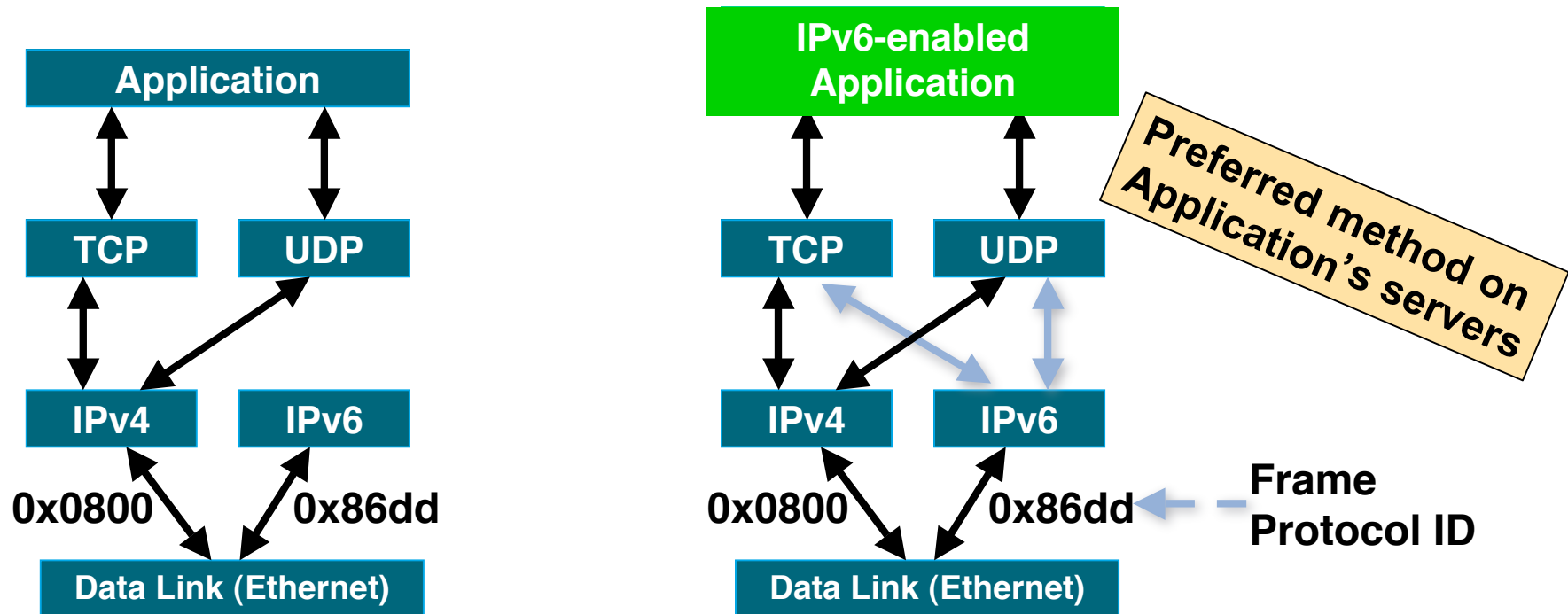
Specifies the standardisation of discovery, control and encapsulation methods for connecting IPv4 networks across IPv6 networks and IPv6 networks across IPv4 networks in a way that will encourage multiple, inter-operable implementations

<http://www.ietf.org/dyn/wg/charter/softwire-charter.html>

IPv4-IPv6 Co-existence/Transition

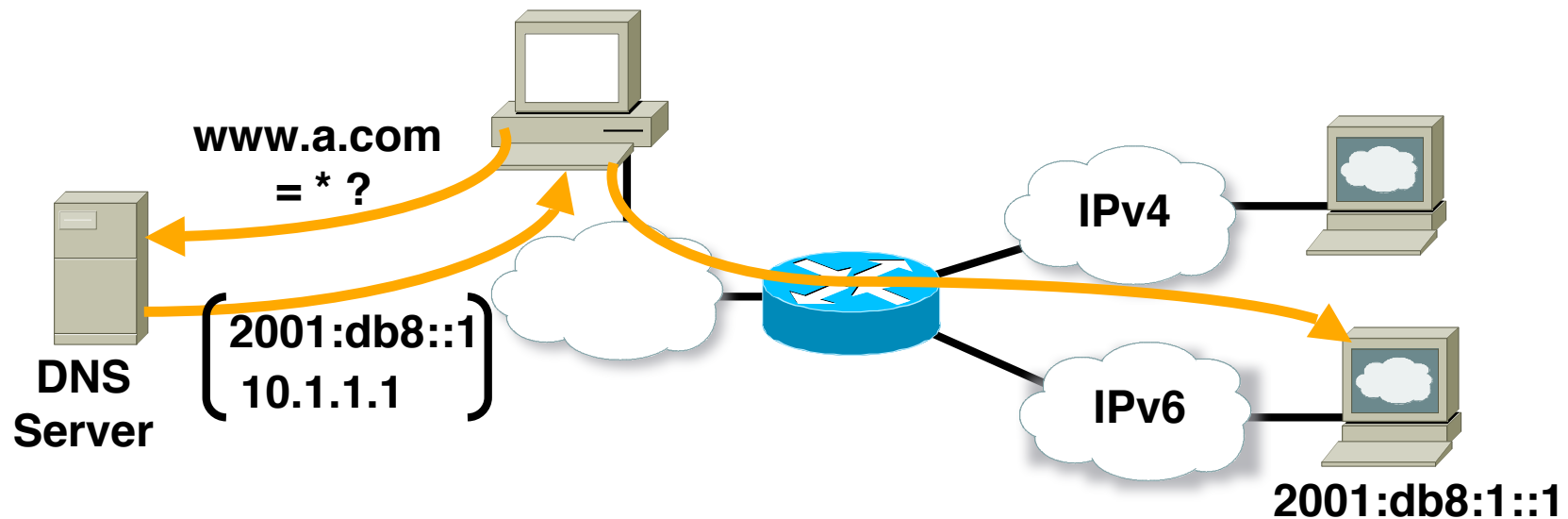
- A wide range of techniques have been identified and implemented, basically falling into three categories:
 - Dual-stack techniques, to allow IPv4 and IPv6 to co-exist in the same devices and networks
 - Tunneling techniques, to avoid order dependencies when upgrading hosts, routers, or regions
 - Translation techniques, to allow IPv6-only devices to communicate with IPv4-only devices
- Expect all of these to be used, in combination

Dual Stack Approach



- Dual stack node means:
 - Both IPv4 and IPv6 stacks enabled
 - Applications can talk to both
 - Choice of the IP version is based on name lookup and application preference

Dual Stack Approach & DNS

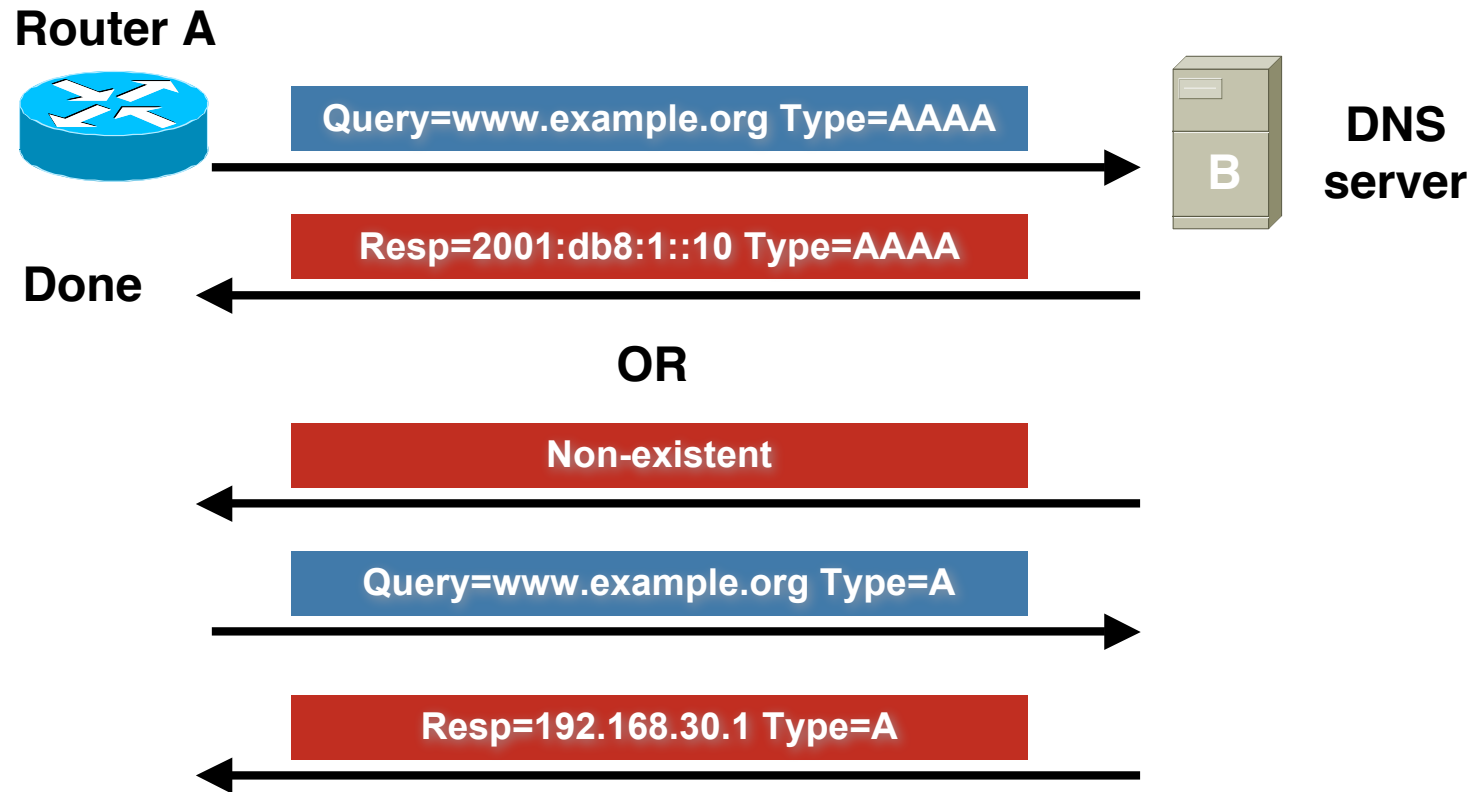


- In a dual stack case, an application that:
 - Is IPv4 and IPv6-enabled
 - Asks the DNS for all types of addresses
 - Chooses one address and, for example, connects to the IPv6 address

IOS IPv6 DNS Client Support

- IOS supports IPv6 DNS client
- Queries DNS servers for IPv6/IPv4:
 - First tries queries for an IPv6 address (AAAA record)
 - If no IPv6 address exists, then query for an IPv4 address (A record)
 - When both IPv6 and IPv4 records exists, the IPv6 address is picked first
- Static hostname to IPv6 address can also be configured
- Note: IPv6 stacks on Windows XP, Linux, FreeBSD, etc also pick IPv6 address before IPv4 address if both exist
 - Check out www.kame.net for example

Example of DNS query



- DNS resolver picks IPv6 AAAA record first

IOS DNS configuration

- DNS commands for IPv6

Define static name for IPv6 addresses

```
ipv6 host <name> [<port>] <v6addr> [<v6addr> ...]
```

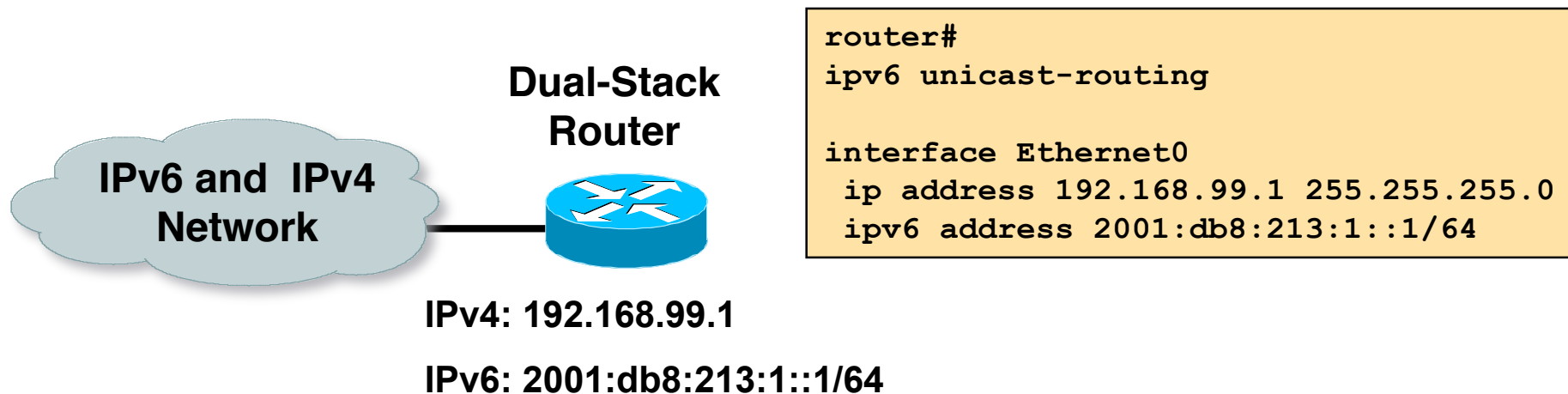
Example: `ipv6 host router1 2001:db8:1::10`

Configuring DNS servers to query

```
ip name-server <address>
```

Example: `ip name-server 2001:db8:1::10`

A Dual Stack Configuration



- IPv6-enabled router

If IPv4 and IPv6 are configured on one interface, the router is dual-stacked

Telnet, Ping, Traceroute, SSH, DNS client, TFTP,...

Using Tunnels for IPv6 Deployment

- Many techniques are available to establish a tunnel:

- Manually configured

- Manual Tunnel (RFC 2893)

- GRE (RFC 2473)

- Semi-automated

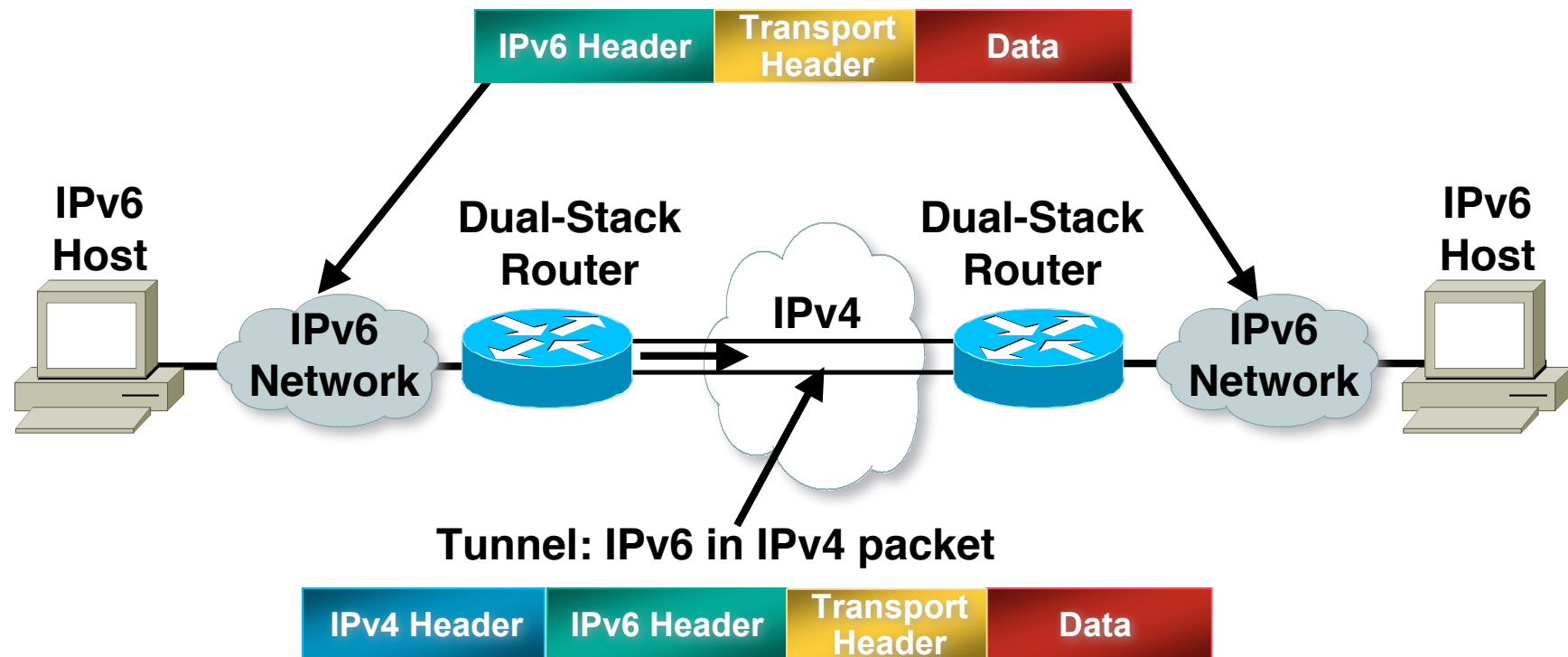
- Tunnel broker

- Automatic

- 6to4 (RFC 3056)

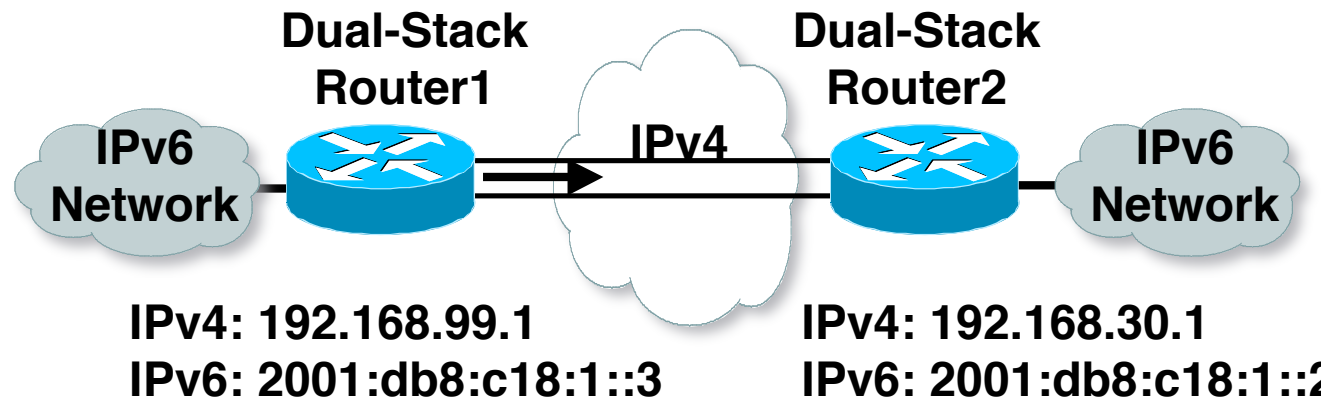
- ISATAP

IPv6 over IPv4 Tunnels



- Tunneling is encapsulating the IPv6 packet in the IPv4 packet
- Tunneling can be used by routers and hosts

Manually Configured Tunnel (RFC2893)

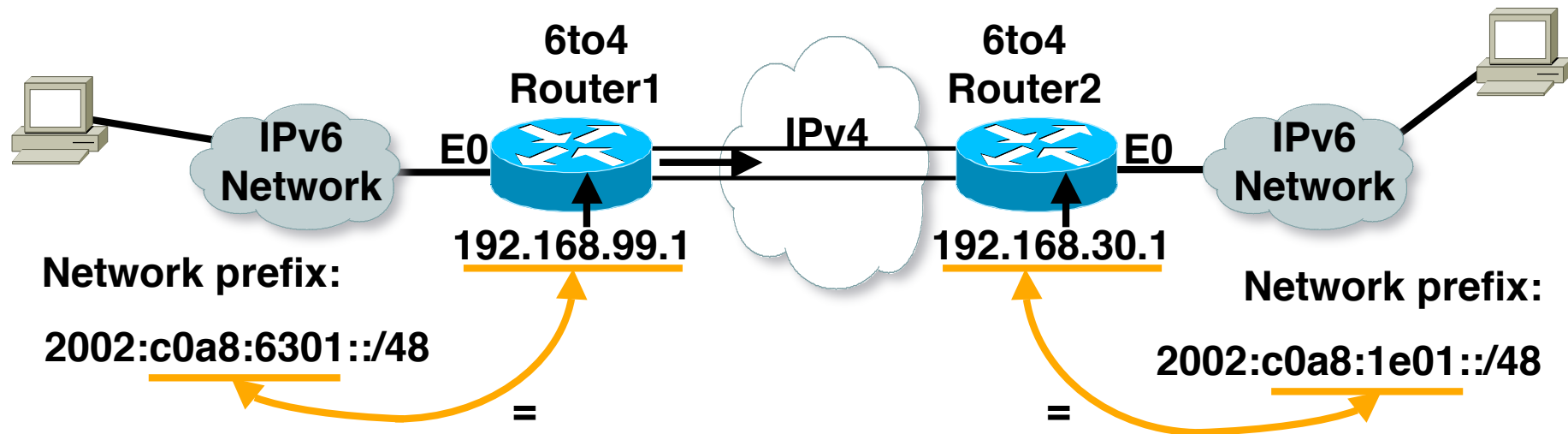


```
router1#  
  
interface Tunnel0  
  ipv6 address 2001:db8:c18:1::3/64  
  tunnel source 192.168.99.1  
  tunnel destination 192.168.30.1  
  tunnel mode ipv6ip
```

```
router2#  
  
interface Tunnel0  
  ipv6 address 2001:db8:c18:1::2/64  
  tunnel source 192.168.30.1  
  tunnel destination 192.168.99.1  
  tunnel mode ipv6ip
```

- Manually Configured tunnels require:
 - Dual stack end points
 - Both IPv4 and IPv6 addresses configured at each end

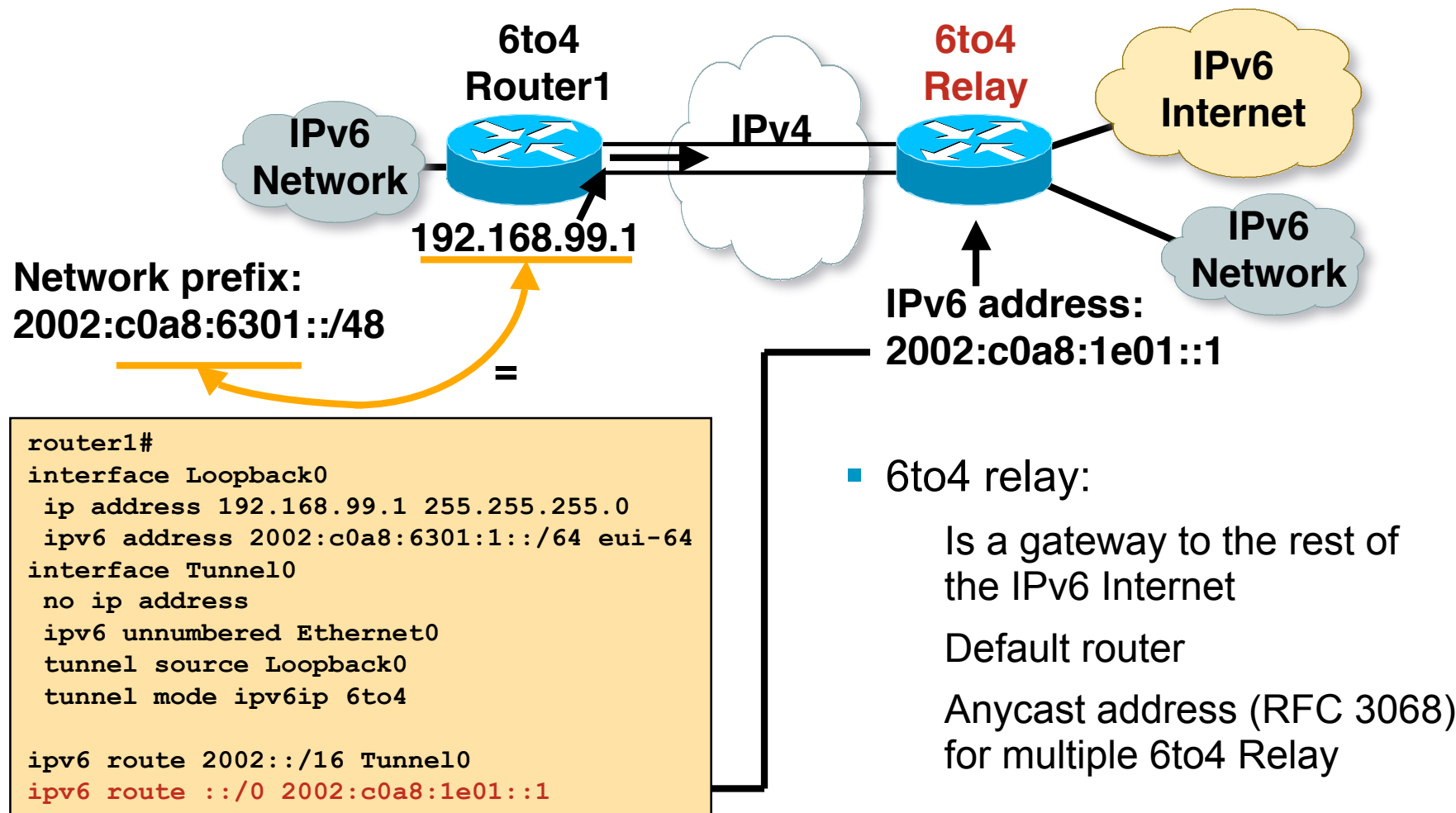
6to4 Tunnel (RFC 3056)



- 6to4 Tunnel:
 - Is an automatic tunnel method
 - Gives a prefix to the attached IPv6 network
 - 2002::/16 assigned to 6to4
 - Requires one global IPv4 address on each Ingress/Egress site

```
router2#  
interface Loopback0  
 ip address 192.168.30.1 255.255.255.0  
 ipv6 address 2002:c0a8:1e01:1::/64 eui-64  
interface Tunnel0  
 no ip address  
 ipv6 unnumbered Ethernet0  
 tunnel source Loopback0  
 tunnel mode ipv6ip 6to4  
  
ipv6 route 2002::/16 Tunnel0
```

6to4 Relay



6to4 in the Internet

- 6to4 prefix is 2002::/16
- 192.88.99.0/24 is the IPv4 anycast network for 6to4 routers
- 6to4 relay service

An ISP who provides a facility to provide connectivity over the IPv4 Internet between IPv6 islands

Is connected to the IPv6 Internet and announces 2002::/16 by BGP to the IPv6 Internet

Is connected to the IPv4 Internet and announces 192.88.99.0/24 by BGP to the IPv4 Internet

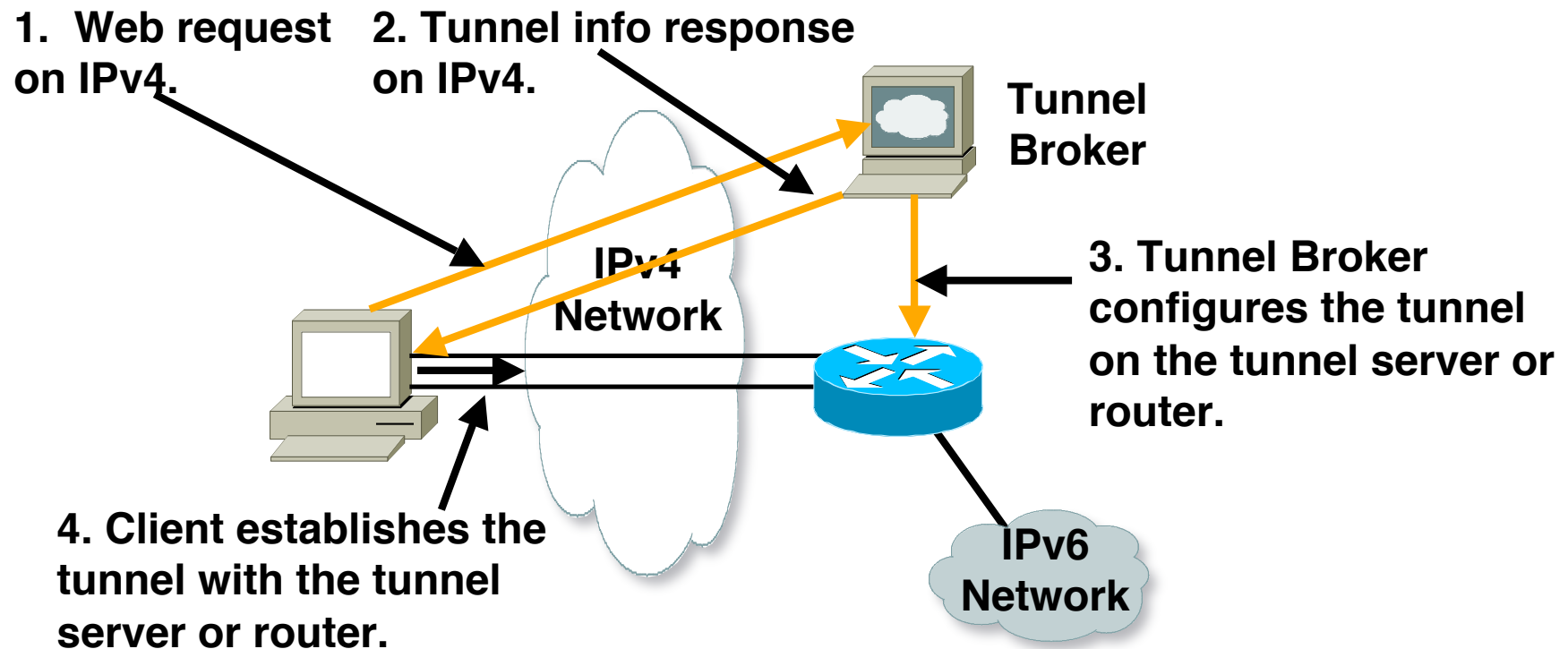
Their router is configured with local IPv4 address of 192.88.99.1 and local IPv6 address of 2002:c058:6301::1

6to4 in the Internet

relay router configuration

```
interface loopback0
  ip address 192.88.99.1 255.255.255.255
  ipv6 address 2002:c058:6301::1/128
!
interface tunnel 2002
  no ip address
  ipv6 unnumbered Loopback0
  tunnel source Loopback0
  tunnel mode ipv6ip 6to4
  tunnel path-mtu-discovery
!
interface FastEthernet0/0
  ip address 105.3.37.1 255.255.255.0
  ipv6 address 2001:db8::1/64
!
router bgp 100
  address-family ipv4
    neighbor <v4-transit> remote-as 101
    network 192.88.99.0 mask 255.255.255.0.
  address-family ipv6
    neighbor <v6-transit> remote-as 102
    network 2002::/16
!
ip route 192.88.99.0 255.255.255.0 null0 254
ipv6 route 2002::/16 tunnel2002
```


Tunnel Broker



- Tunnel broker:

Tunnel information is sent via http-ipv4

ISATAP – Intra Site Automatic Tunnel Addressing Protocol

- Tunnelling of IPv6 in IPv4
- Single Administrative Domain
- Creates a virtual IPv6 link over the full IPv4 network
- Automatic tunnelling is done by a specially formatted ISATAP address which includes:
 - A special ISATAP identifier
 - The IPv4 address of the node
- ISATAP nodes are dual stack

ISATAP Addressing Format

- An ISATAP address of a node is defined as:

A /64 prefix dedicated to the ISATAP overlay link

Interface identifier:

Leftmost 32 bits = 0000:5EFE:

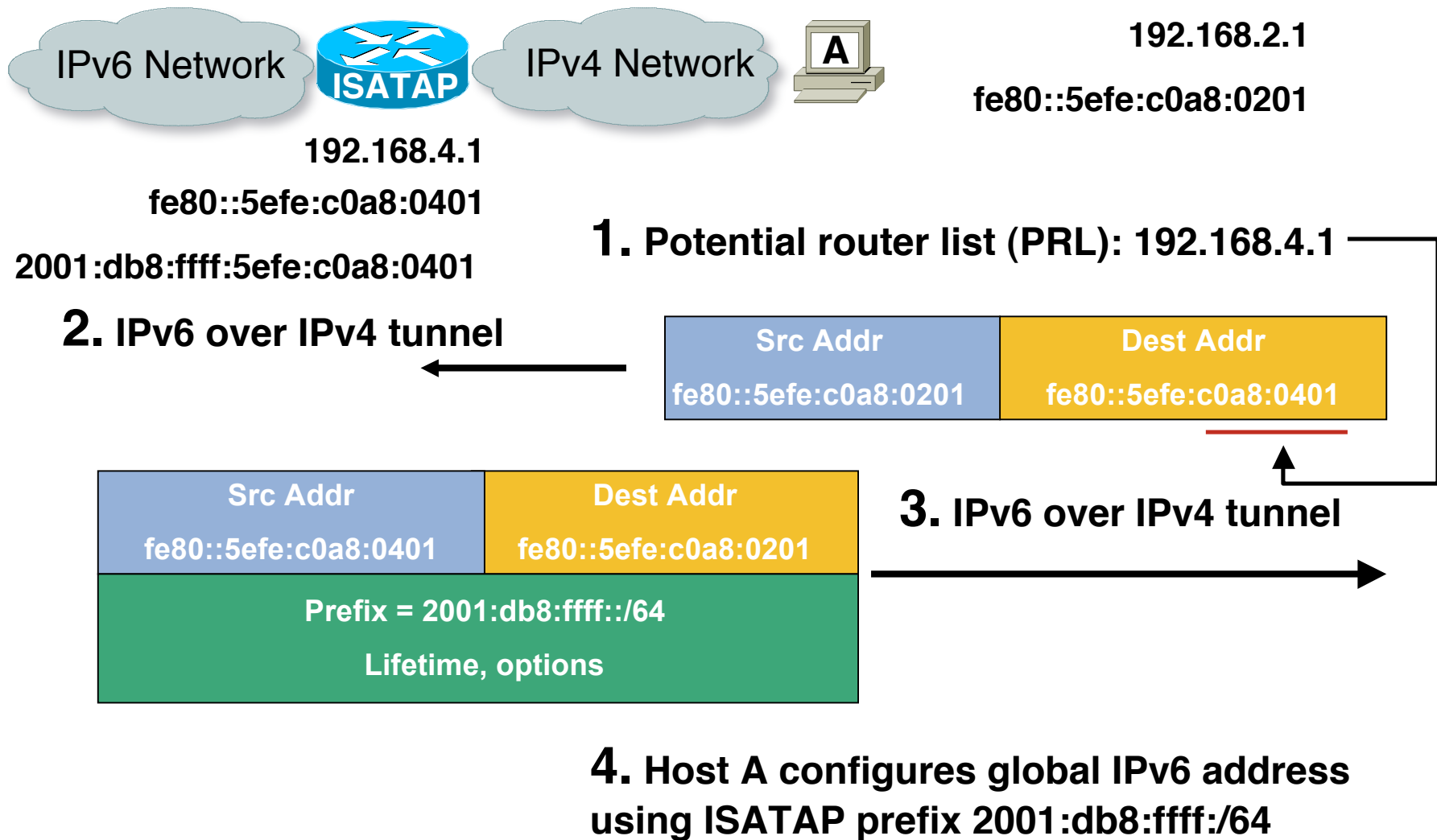
Identify this as an ISATAP address

Rightmost 32 bits = <ipv4 address>

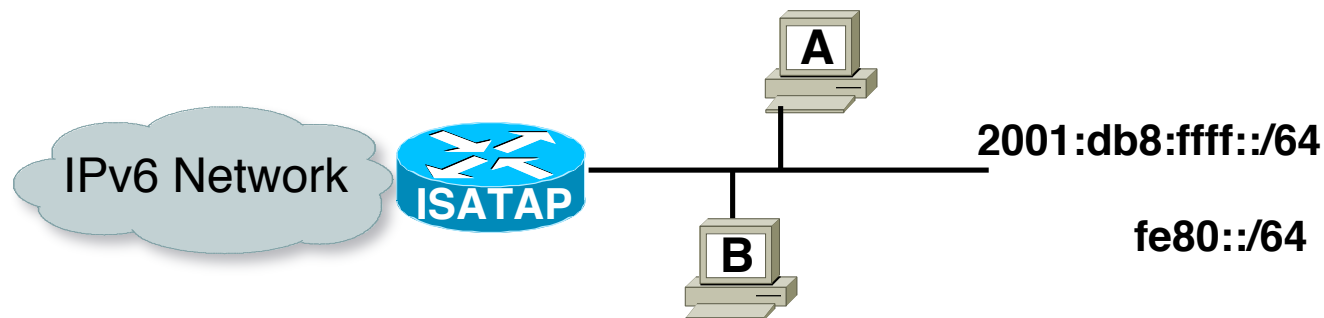
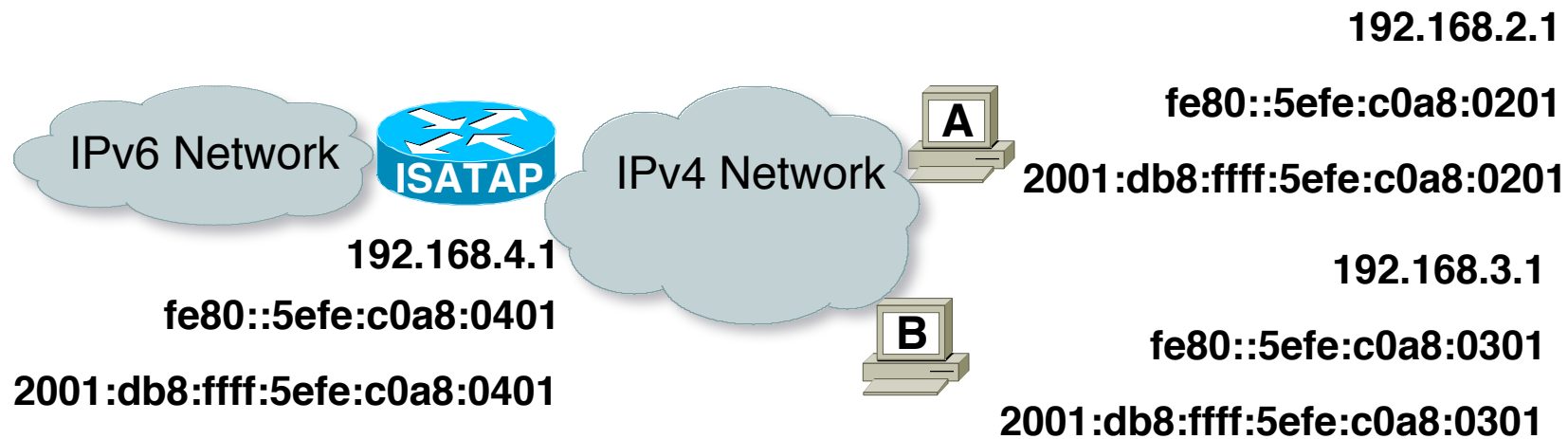
The IPv4 address of the node

ISATAP dedicated prefix	0000:5EFE	IPv4 address
--------------------------------	------------------	---------------------

ISATAP prefix advertisement



ISATAP configuration example



NAT-PT for IPv6

- NAT-PT

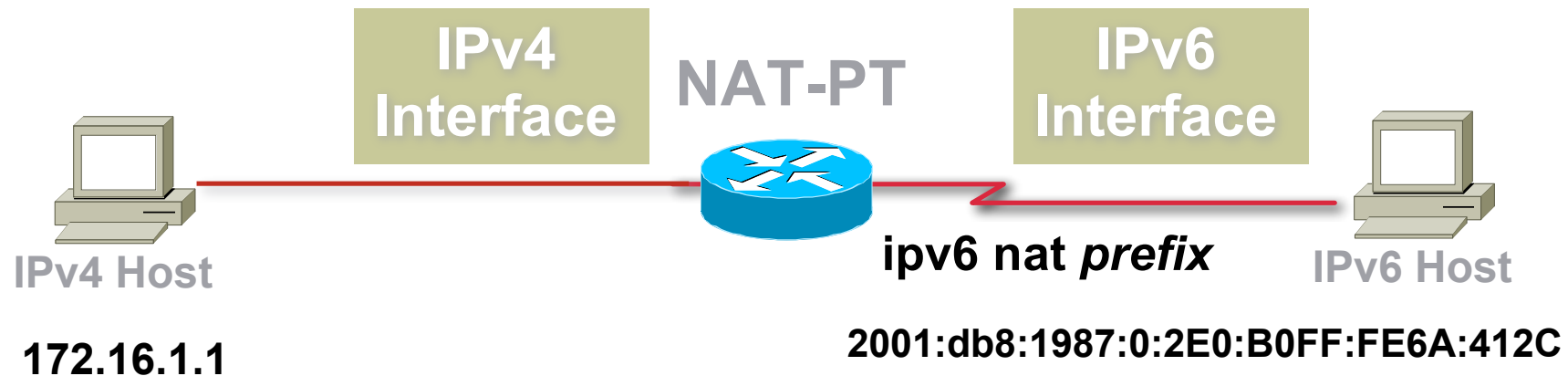
(Network Address Translation – Protocol Translation)

RFC 2766 & RFC 3152

Obsoleted by IETF (RFC4966) but implementations still in use

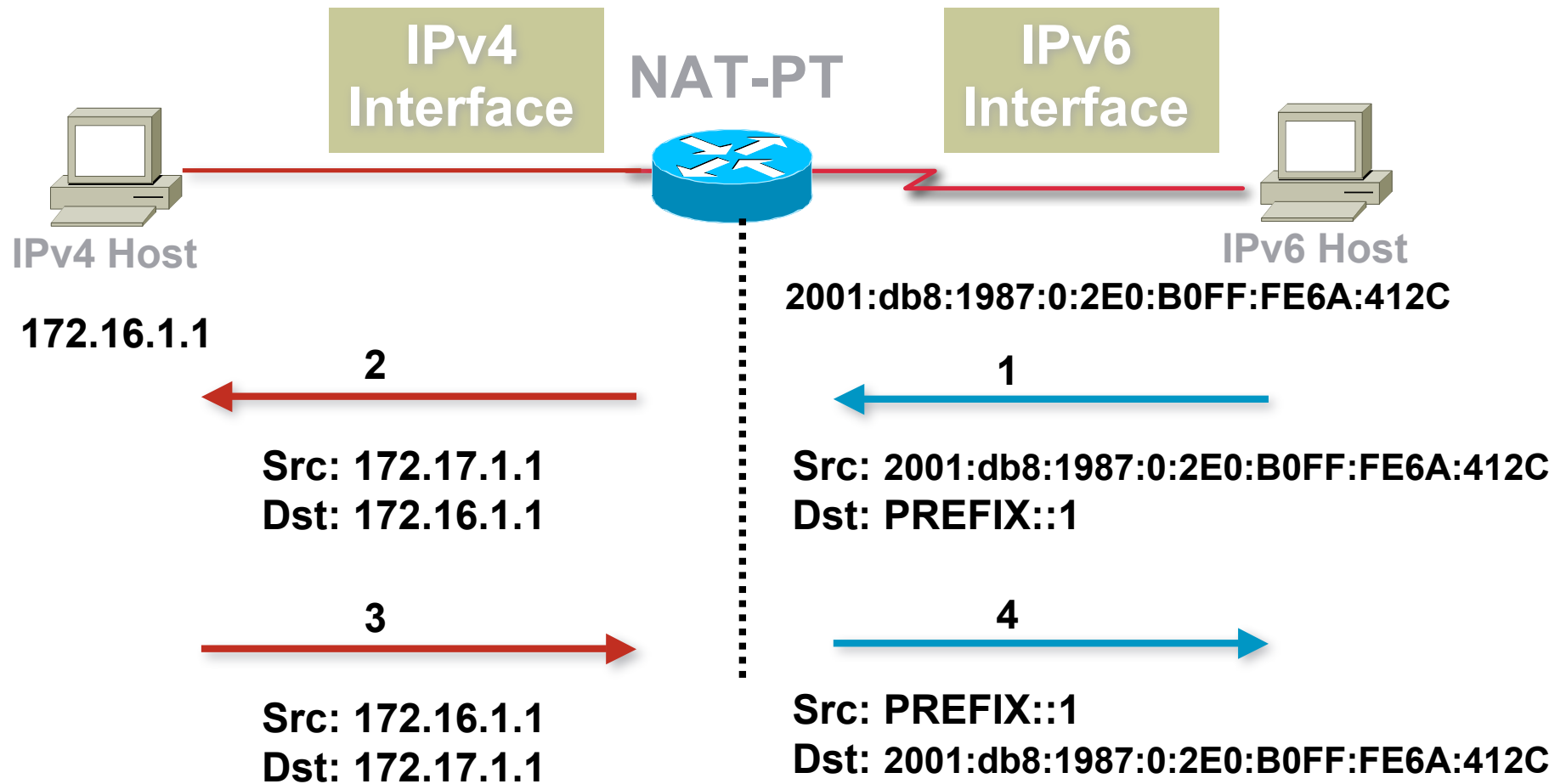
- Allows native IPv6 hosts and applications to communicate with native IPv4 hosts and applications, and vice versa
- Easy-to-use transition and co-existence solution

NAT-PT Concept



- *prefix* is a 96-bit field that allows routing back to the NAT-PT device

NAT-PT packet flow



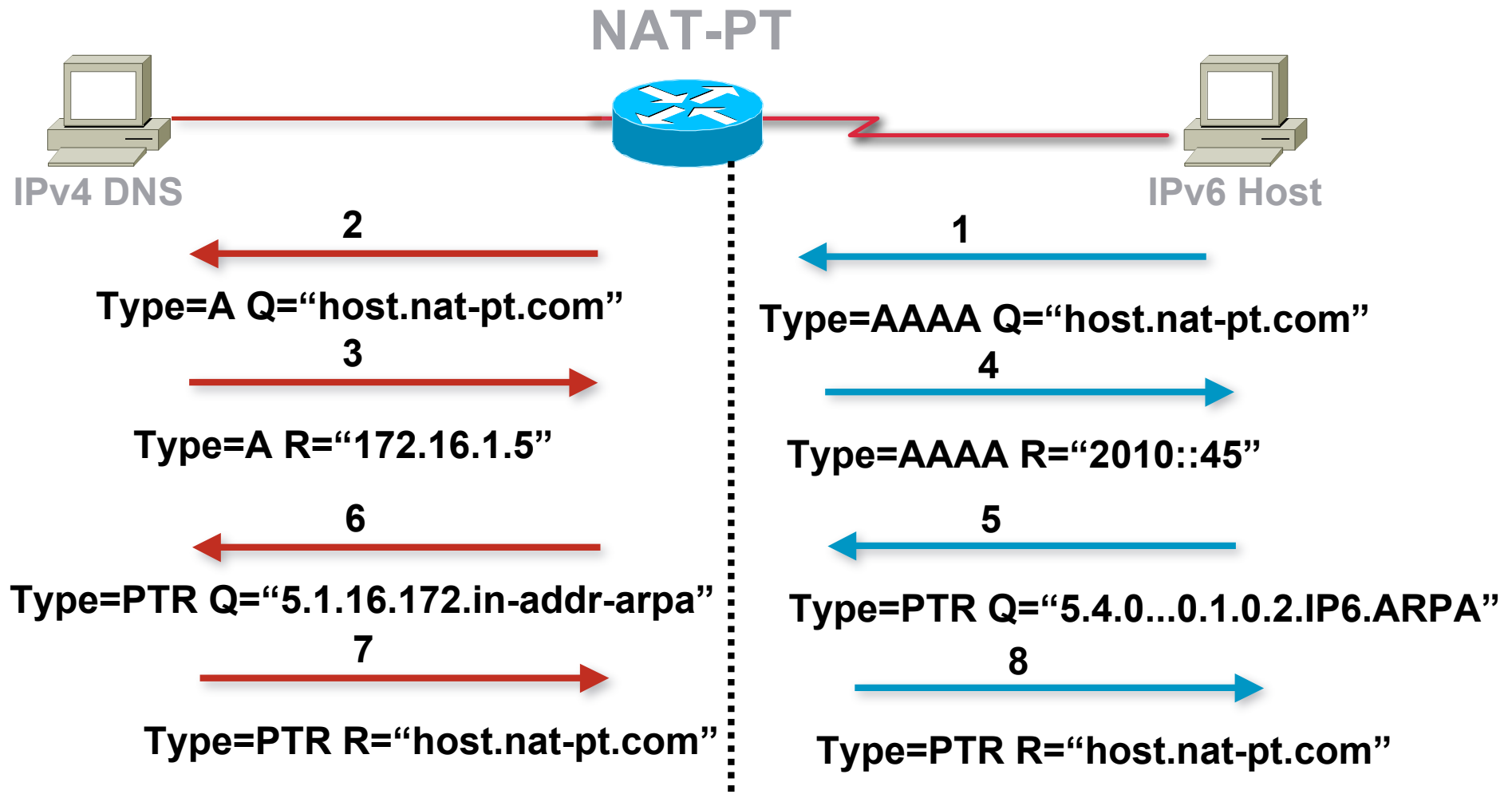
Cisco IOS NAT-PT features

- IP Header and Address translation
- Support for ICMP and DNS embedded translation
- Auto-aliasing of NAT-PT IPv4 Pool Addresses
- Future developments will add FTP ALG, Address Overload and fragmentation support

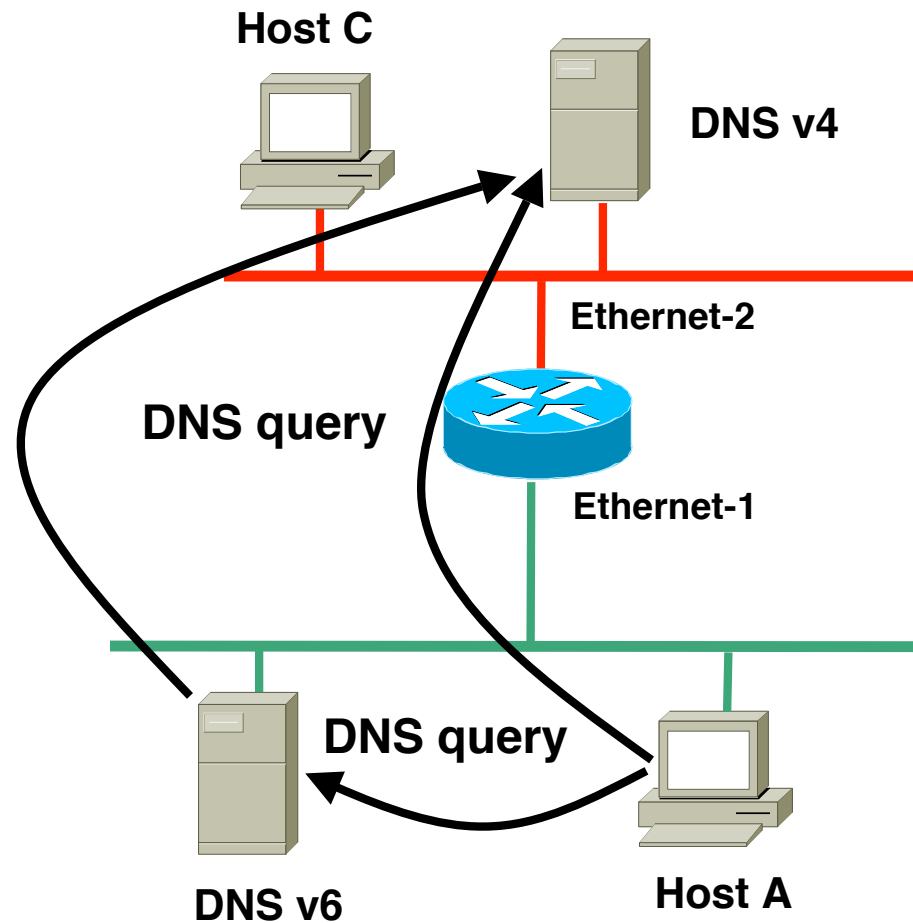
Stateless IP ICMP Translation

<i>IPv6 field</i>	<i>IPv4 field</i>	<i>Action</i>
Version = 6	Version = 4	Overwrite
Traffic class	DSCP	Copy
Flow label	N/A	Set to 0
Payload length	Total length	Adjust
Next header	Protocol	Copy
Hop limit	TTL	Copy

DNS Application Layer Gateway



DNS ALG address assignment



- TTL value in DNS Resource Record = 0

Configuring NAT-PT (1)

- Enabling NAT-PT

```
[no] ipv6 nat
```

- Configure global/per interface NAT-PT prefix

```
[no] ipv6 nat prefix <prefix>::/96
```

- Configuring static address mappings

```
[no] ipv6 nat v6v4 source <v6 address> <v4 address>
```

```
[no] ipv6 nat v4v6 source <v4 address> <v6 address>
```

Configuring NAT-PT (2)

- Configuring dynamic address mappings

```
[no] ipv6 nat v6v4 source <list,route-map> <ipv6  
list, route-map> pool <v4pool>
```

```
[no] ipv6 nat v6v4 pool <v4pool> <ipv4 addr>  
<ipv4addr> prefix-length <n>
```

- Configure Translation Entry Limit

```
[no] ipv6 nat translation max-entries <n>
```

- Debug commands

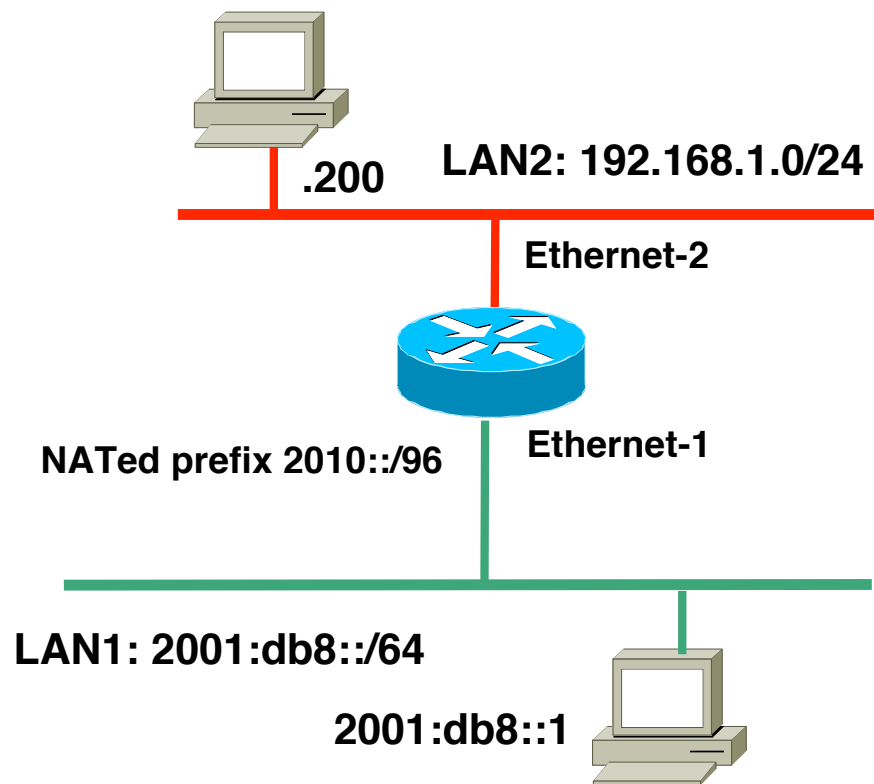
```
debug ipv6 nat
```

```
debug ipv6 nat detailed
```

NAT-PT translation timeouts

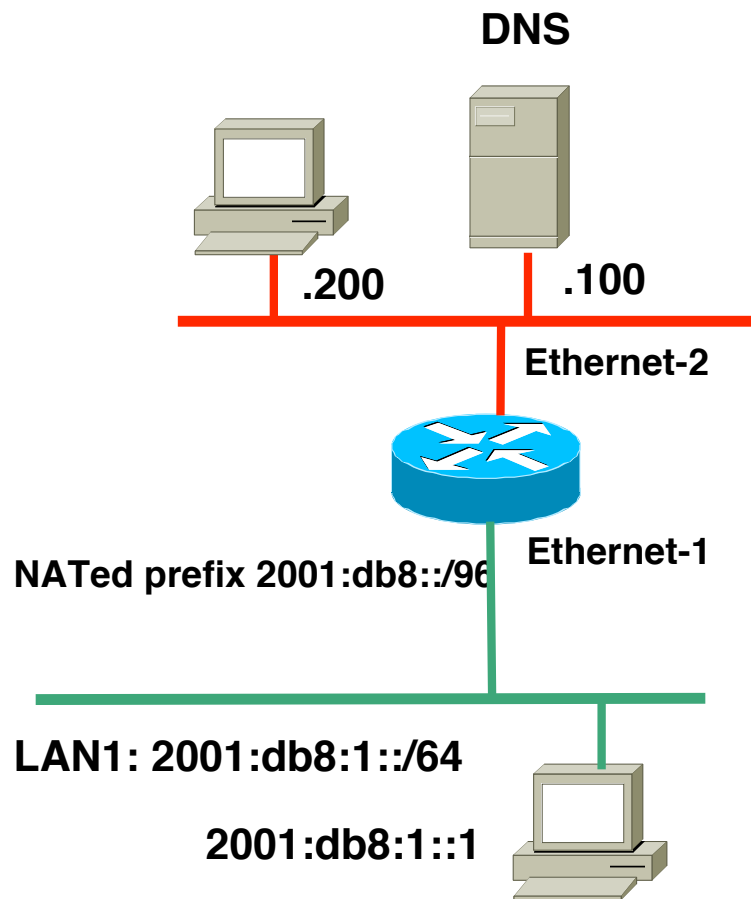
- Dynamic translations time out after 24 hours
`[no] ipv6 nat translation timeout <seconds>`
- Non-DNS UDP translations time out after 5 minutes
`[no] ipv6 nat translation udp-timeout <seconds>`
- DNS translations time out after 1 minute
`[no] ipv6 nat translation dns-timeout <seconds>`
- TCP translations time out after 24 hours, unless a RST or FIN is seen on the stream, in which case it times after 1 minute
`[no] ipv6 nat translation tcp-timeout <seconds>`
`[no] ipv6 nat translation finrst-timeout <seconds>`
`[no] ipv6 nat translation icmp-timeout <seconds>`

Cisco IOS NAT-PT configuration example



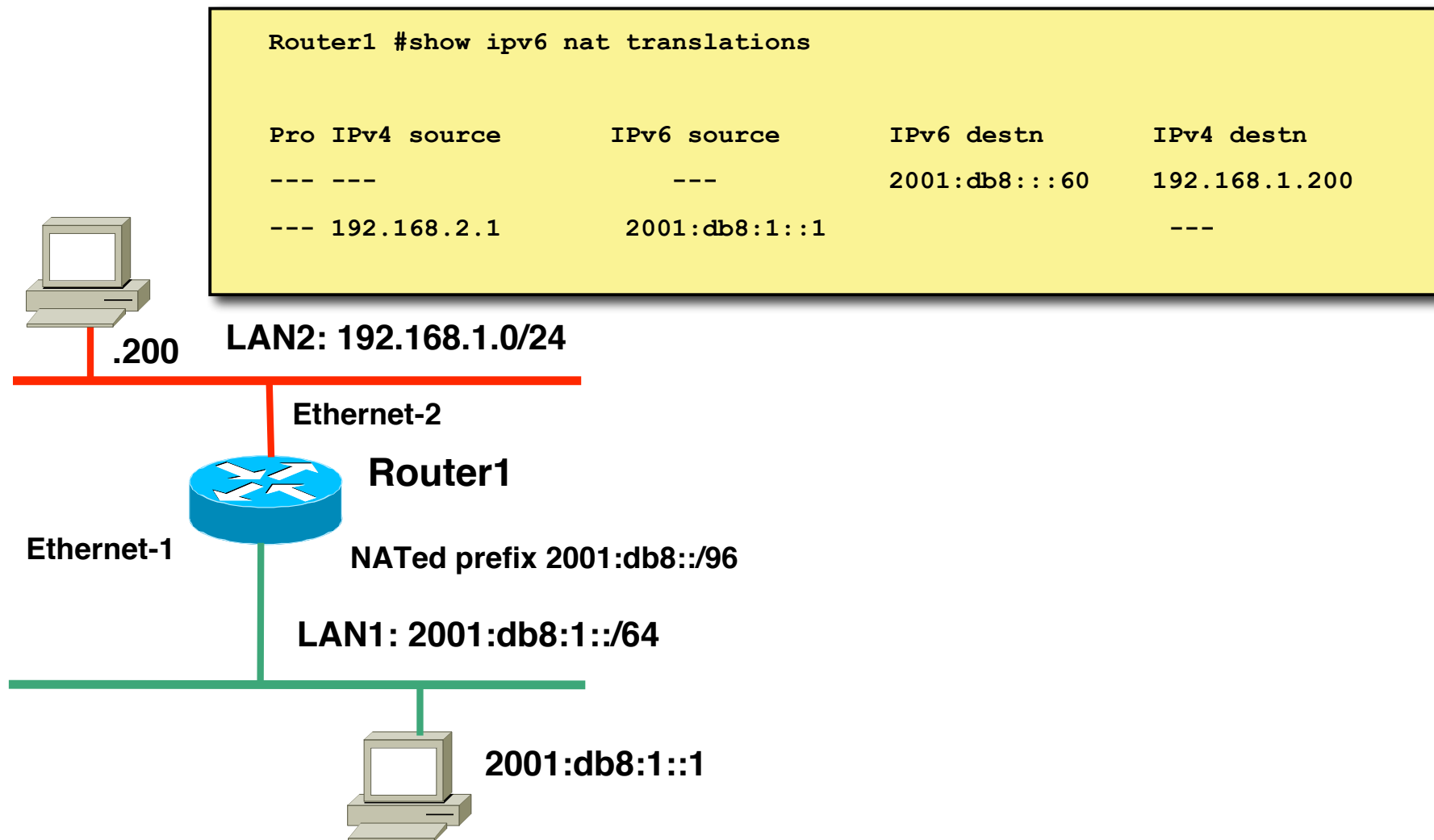
```
interface ethernet-1
  ipv6 address 2001:db8::10/64
  ipv6 nat
!
interface ethernet-2
  ip address 192.168.1.1 255.255.255.0
  ipv6 nat prefix 2010::/96
  ipv6 nat
!
ipv6 nat v6v4 source 2001:db8::1 192.168.2.1
ipv6 nat v4v6 source 192.168.1.200 2001:db8::60
!
```


Cisco IOS NAT-PT w/ DNS ALG Configuration

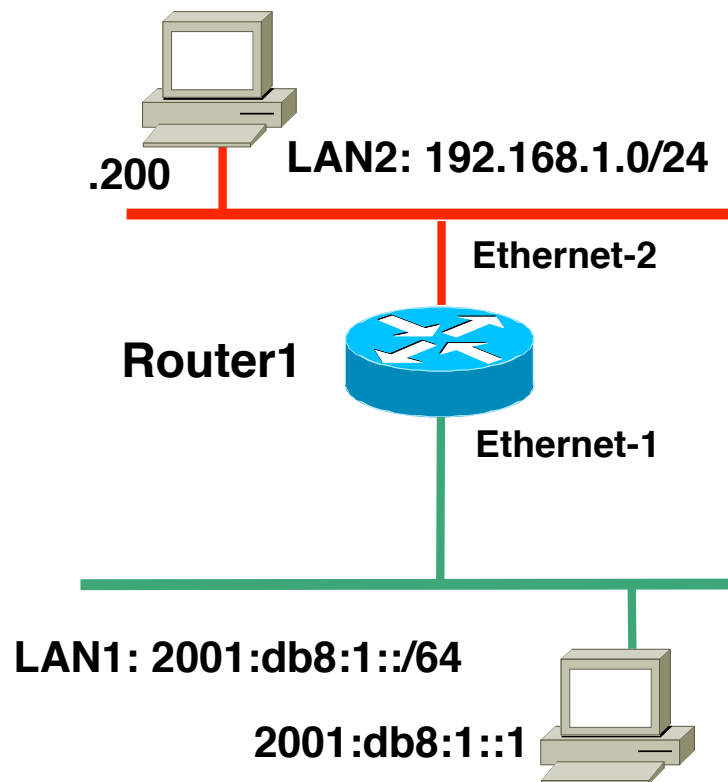


```
interface ethernet-1
  ipv6 address 2001:db8:1::10/64
  ipv6 nat
!
interface ethernet-2
  ip address 192.168.1.1 255.255.255.0
  ipv6 nat
!
ipv6 nat v4v6 source 192.168.1.100 2010::1
!
ipv6 nat v6v4 source list v6-list map1 pool v4pool1
ipv6 nat v6v4 pool v4pool1 192.168.2.1 192.168.2.10
prefix-length 24
ipv6 nat service dns
ipv6 nat prefix 2001:db8::/96
!
ipv6 access-list v6-list
  permit 2001:db8:1::/64 any
```

Cisco IOS NAT-PT display (1)



Cisco IOS NAT-PT display (2)



```
Router1#show ipv6 nat statistics
```

```
Total active translations: 15 (2 static, 3 dynamic;  
10 extended)
```

```
NAT-PT interfaces:
```

```
Ethernet-1, Ethernet-2
```

```
Hits: 10 Misses: 0
```

```
Expired translations: 0
```

NAT-PT Summary

- Points of note:

- ALG per application carrying IP address

- No End to End security

- No DNSsec

- No IPsec because different address realms

- Conclusion

- Easy IPv6 / IPv4 co-existence mechanism

- Enable applications to cross the protocol barrier



IPv6 Servers and Services

Unix Webserver

- Apache 2.x supports IPv6 by default
- Simply edit the **httpd.conf** file

HTTPD listens on all IPv4 interfaces on port 80 by default

For IPv6 add:

```
Listen [2001:db8:10::1]:80
```

So that the webserver will listen to requests coming on the interface configured with 2001:db8:10::1/64

Unix Nameserver

- BIND 9 supports IPv6 by default
- To enable IPv6 nameservice, edit /etc/named.conf:

```
options {  
    listen-on-v6 { any; };  
};  
zone "workshop.net" {  
    type master;  
    file "workshop.net.zone";  
};  
zone "8.b.d.0.1.0.0.2.ip6.arpa" {  
    type master;  
    file "workshop.net.rev-zone";  
};
```

**Tells bind to listen
on IPv6 ports**



**Forward zone contains
v4 and v6 information**



**Sets up reverse
zone for IPv6 hosts**



Unix

Sendmail

- Sendmail 8 as part of a distribution is usually built with IPv6 enabled

But the configuration file needs to be modified

- If compiling from scratch, make sure NETINET6 is defined
- Then edit `/etc/mail/sendmail.mc` thus:
Remove the line which is for IPv4 only and enable the IPv6 line thus (to support both IPv4 and IPv6):
`DAEMON_OPTIONS(`Port=smtp, Addr::, Name=MTA-v6, Family=inet6')`
Remake `sendmail.cf`, then restart sendmail

Unix FTP Server

- Vsftpd is covered here
 - Standard part of many Linux distributions now
- IPv6 is supported, but not enable by default
 - Need to run two vsftpd servers, one for IPv4, the other for IPv6
- IPv4 configuration file: `/etc/vsftpd/vsftpd.conf`
 - `listen=YES`
 - `listen_address=<ipv4 addr>`
- IPv6 configuration file: `/etc/vsftpd/vsftpdv6.conf`
 - `listen=NO`
 - `listen_ipv6=YES`
 - `listen_address6=<ipv6 addr>`

Unix Applications

- OpenSSH

Uses IPv6 transport before IPv4 transport if IPv6 address available

- Mozilla/Firefox/Thunderbird

Supports IPv6, but still hampered by broken IPv6 nameservers and IPv6 connectivity

In `about:config` the value `network.dns.disableIPv6` is set to `true` by default

Change to `false` to enable IPv6

MacOS X

- IPv6 installed
- IPv6 enabled by default

Will use autoconfiguration by default

Enter **System Preferences** and then **Network** to enter static IPv6 addresses (depends on MacOS X version)

- Applications will use IPv6 transport if IPv6 address offered in name lookups

FreeBSD – client

- IPv6 installed, but disabled by default
- To enable using autoconfiguration:

Simply edit `/etc/rc.conf` to include these lines

```
ipv6_enable="YES"  
ipv6_network_interfaces="em0"
```

Where

`em0` should be replaced with the name of the Ethernet interface on the device

- And then reboot the system

FreeBSD – server

- IPv6 installed, but disabled by default
- To enable using static configuration:

Edit `/etc/rc.conf` to include these lines

```
ipv6_enable="YES"
ipv6_network_interfaces="em0"
ipv6_ifconfig_em0="2001:db8::1 prefixlen 64"
ipv6_defaultrouter="fe80::30%em0"
```

Where

`em0` should be replaced with the name of the Ethernet interface on the device

`2001:db8::1` should be replaced with the IPv6 address

`fe80::30` should be replaced with the default gateway

- And then reboot the system

RedHat/Fedora/CentOS Linux – client

- IPv6 installed, but disabled by default
- To enable:
 - Edit `/etc/sysconfig/network` to include the line
NETWORKING_IPV6=yes
 - Edit `/etc/sysconfig/network-scripts/ifcfg-eth0` to include:
IPV6INIT=yes
 - And then `/sbin/service network restart` or reboot
- Other Linux distributions will use similar techniques

RedHat/Fedora/CentOS Linux – server

- To enable:

Edit `/etc/sysconfig/network` to include:

```
NETWORKING_IPV6=yes
```

```
IPV6_DEFAULTGW=FE80::30
```

```
IPV6_DEFAULTDEV=eth0
```

Edit `/etc/sysconfig/network-scripts/ifcfg-eth0` to include:

```
IPV6ADDR=2001:db8::1/64
```

```
IPV6INIT=yes
```

```
IPV6_AUTOCONF=no
```

Where

`eth0` should be replaced with the name of the Ethernet interface on the device

`2001:db8::1` should be replaced with the IPv6 address

`fe80::30` should be replaced with the default gateway

And then `/sbin/service network restart` or reboot

Windows XP & Vista

- XP

IPv6 installed, but disabled by default

To enable, start command prompt and run “**ipv6 install**”

- Vista

IPv6 installed, enabled by default

- Most apps (including IE) will use IPv6 transport if IPv6 address offered in name lookups



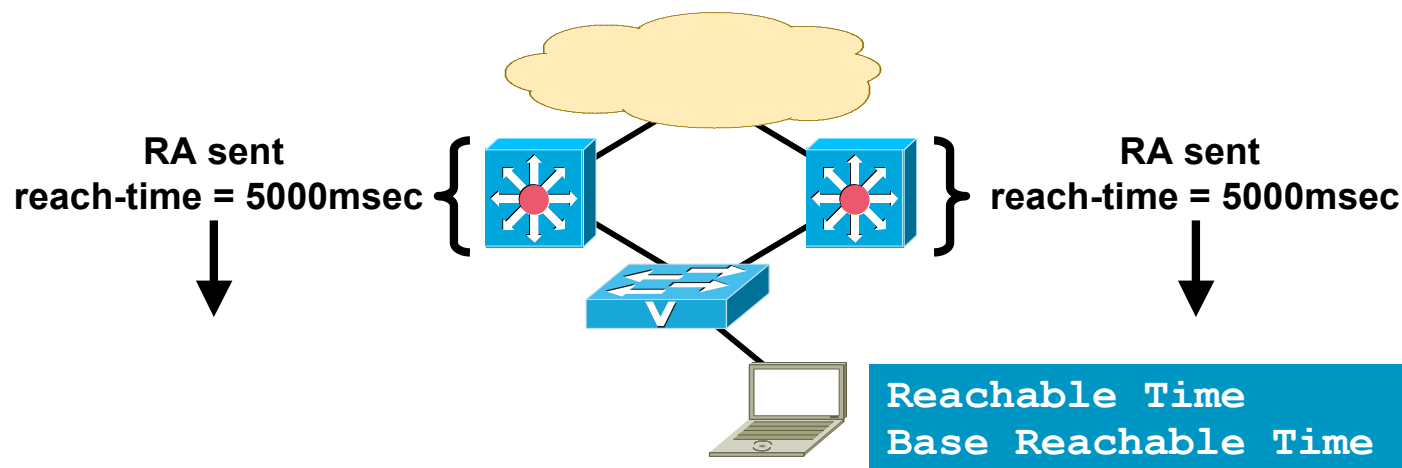
Other IOS Features

Redundancy, Radius, DHCP,...

First-Hop Redundancy

- When HSRP, GLBP and VRRP for IPv6 are not available
- NUD can be used for rudimentary HA at the first-hop (today this only applies to the Campus/DC...HSRP is available on routers)
`(config-if)#ipv6 nd reachable-time 5000`
- Hosts use NUD “reachable time” to cycle to next known default gateway (30 seconds by default)

```
Default Gateway . . . . . : 10.121.10.1  
                           fe80::211:bcff:fec0:d000%4  
                           fe80::211:bcff:fec0:c800%4
```

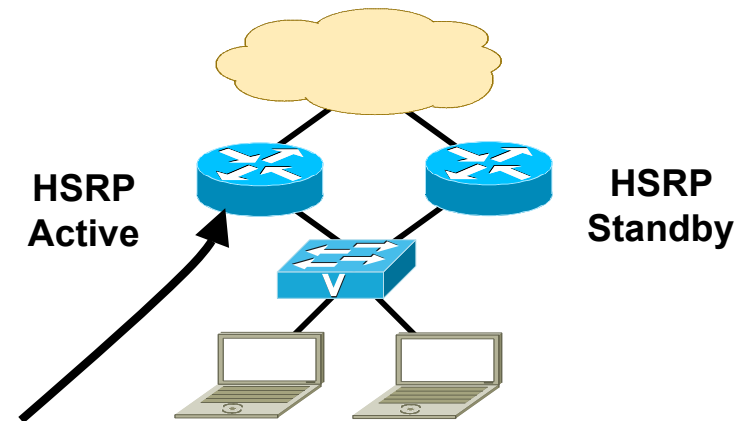


HSRP for IPv6

- Many similarities with HSRP for IPv4
- Changes occur in Neighbor Advertisement, Router Advertisement, and ICMPv6 redirects
- No need to configure GW on hosts (RAs are sent from HSRP Active router)
- Virtual MAC derived from HSRP group number and virtual IPv6 Link-local address
- IPv6 Virtual MAC range:
0005.73A0.0000 - 0005.73A0.0FFF
(4096 addresses)
- HSRP IPv6 UDP Port Number 2029 (IANA Assigned)
- No HSRP IPv6 secondary address
- No HSRP IPv6 specific debug

Host with GW of Virtual IP

```
#route -A inet6 | grep ::/0 | grep eth2
::/0      fe80::207:85ff:fef3:2f60      UGDA 1024 3      0 eth2
::/0      fe80::205:9bff:febf:5ce0      UGDA 1024 0      0 eth2
::/0      fe80::5:73ff:fea0:1
```

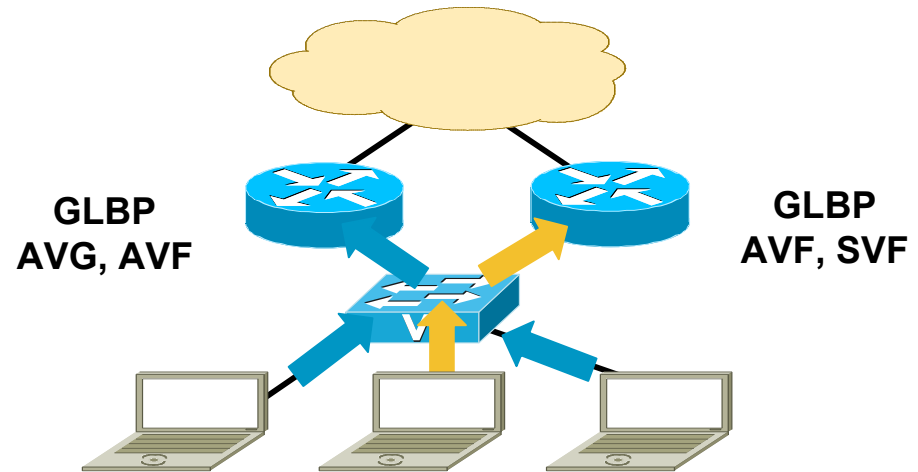


```
interface FastEthernet0/1
  ipv6 address 2001:DB8:66:67::2/64
  ipv6 cef
  standby version 2
  standby 1 ipv6 autoconfig
  standby 1 timers msec 250 msec 800
  standby 1 preempt
  standby 1 preempt delay minimum 180
  standby 1 authentication md5 key-string cisco
  standby 1 track FastEthernet0/0
```

GLBP for IPv6

- Many similarities with GLBP for IPv4 (CLI, Load-balancing)
- Modification to Neighbor Advertisement, Router Advertisement
- GW is announced via RAs
- Virtual MAC derived from GLBP group number and virtual IPv6 Link-local address

AVG=Active Virtual Gateway
AVF=Active Virtual Forwarder
SVF=Standby Virtual Forwarder



```
interface FastEthernet0/0
  ipv6 address 2001:DB8:1::1/64
  ipv6 cef
  glbp 1 ipv6 autoconfig
  glbp 1 timers msec 250 msec 750
  glbp 1 preempt delay minimum 180
  glbp 1 authentication md5 key-string cisco
```

IPv6 General Prefix

- Provides an easy/fast way to deploy prefix changes
- Example: 2001:db8:cafe::/48 = General Prefix
- Fill in interface specific fields after prefix

"office ::11:0:0:0:1" = 2001:db8:cafe:11::1/64

```
ipv6 unicast-routing
ipv6 cef
ipv6 general-prefix office
2001:DB8:CAFE::/48
!
interface GigabitEthernet3/2
  ipv6 address office ::2/127
  ipv6 cef
!
interface GigabitEthernet1/2
  ipv6 address office ::E/127
  ipv6 cef
```

```
interface Vlan11
  ipv6 address office ::11:0:0:0:1/64
  ipv6 cef
!
interface Vlan12
  ipv6 address office ::12:0:0:0:1/64
  ipv6 cef
```

```
6k-agg-1#sh ipv6 int vlan 11 | i Global|2001
Global unicast address(es):
  2001:DB8:CAFE:11::1, subnet is 2001:DB8:CAFE:11::/64
```

AAA/RADIUS

- RADIUS attributes and IPv6 (RFC3162)—Cisco IOS® 12.3(4)T
- RADIUS Server support requires an upgrade (supporting RFC3162)

Few RADIUS solutions support RFC3162 functionality today

- IPv6 AAA/RADIUS Configuration
www.cisco.com/warp/public/cc/pd/iosw/prodlit/ipv6a_wp.htm

RADIUS Configuration with permanently assigned /64:

```
Auth-Type = Local, Password = "foo"  
User-Service-Type = Framed-User,  
Framed-Protocol = PPP,  
cisco-avpair = "ipv6:prefix=2001:DB8:1:1::/64"
```

Interface Identifier attribute (Framed-Interface-Id) can be used:

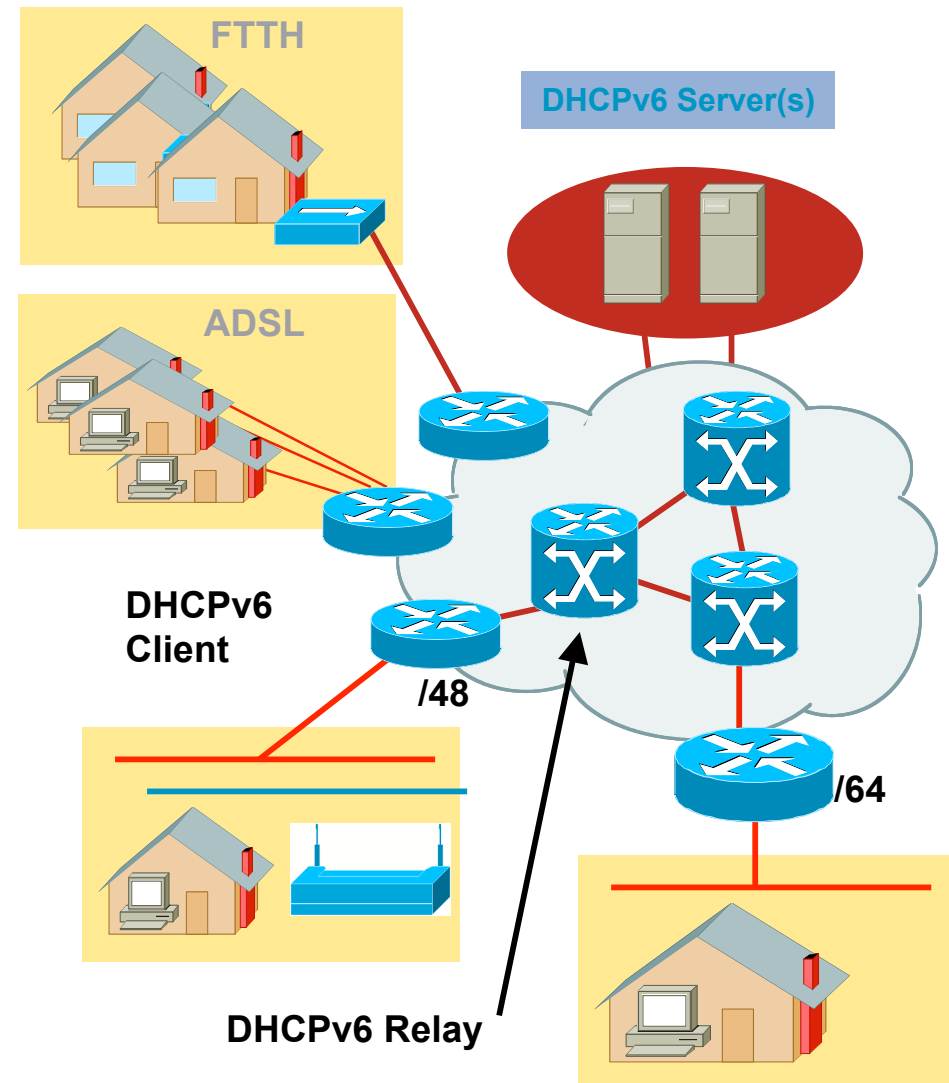
```
Interface-Id = "0:0:0:1",
```

DHCPv6 Overview

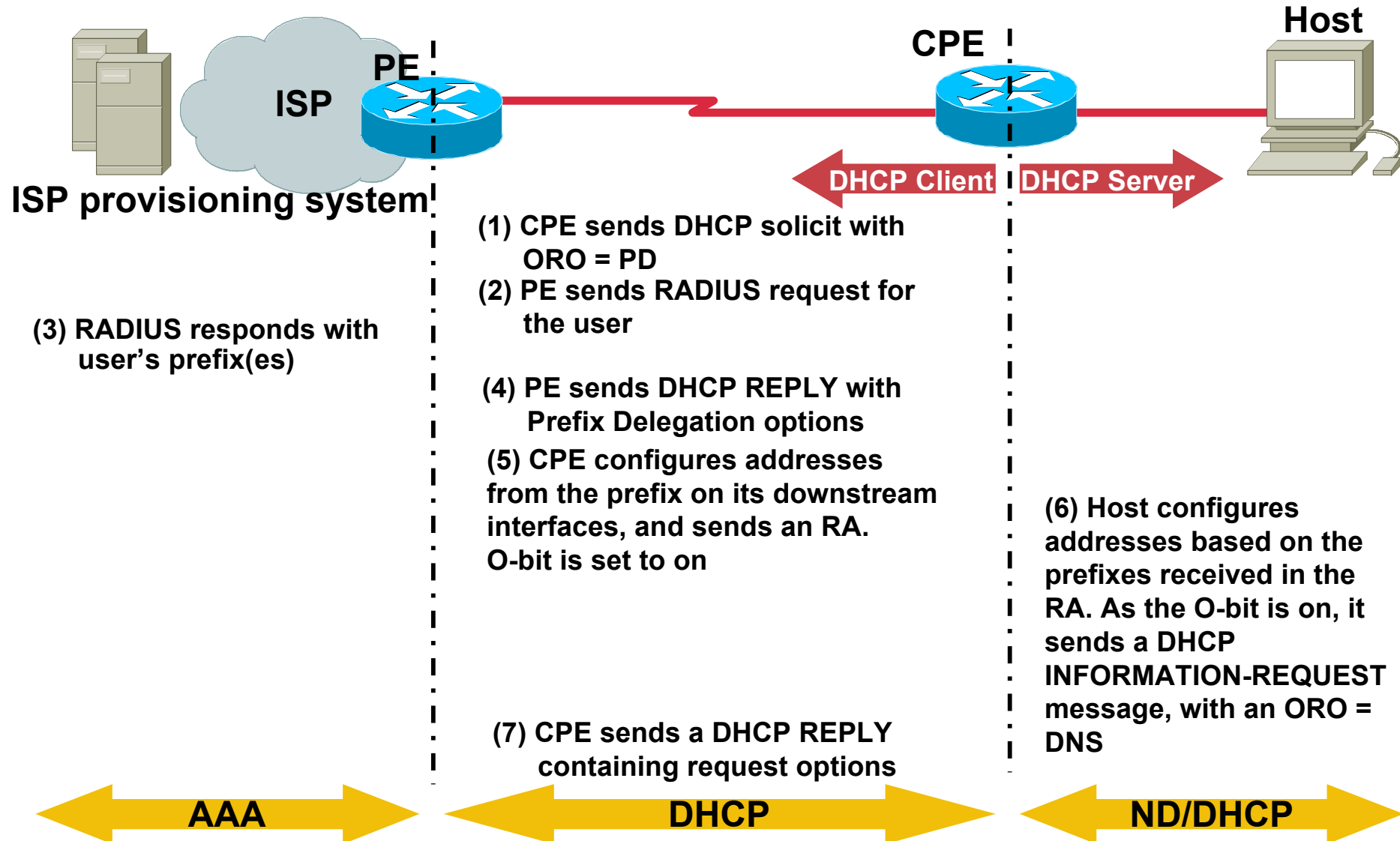
- Operational model based on DHCPv4, but details differ:
 - Client uses link-local address for message exchanges
 - Server can assign multiple addresses per client through Identity Associations
 - Clients and servers identified by DUID
 - Address assignment & Prefix delegation
 - Message exchanges similar, but will require new protocol engine
 - Server-initiated configuration, authentication part of the base specification
 - Extensible option mechanism & Relay-agents
- Allows both stateful and stateless configuration
- RFC 3315 (DHCPv6) has additional options:
 - DNS configuration—RFC 3646
 - Prefix delegation—RFC 3633
 - NTP servers
 - Stateless DHCP for IPv6—RFC 3736

DHCPv6 PD: RFC 3633

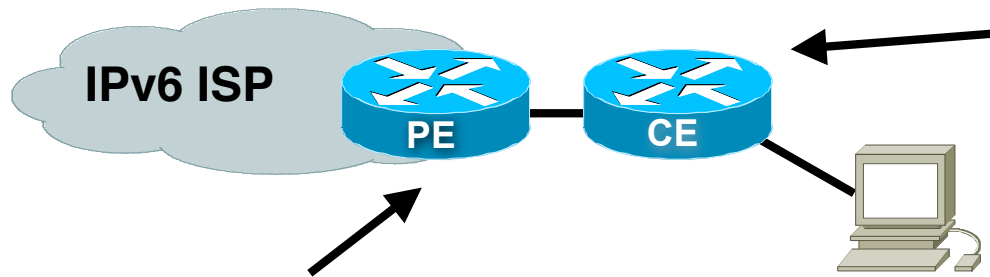
- Media independence
e.g., ADSL, FTTH
Only knows identity of requesting router
- Leases for prefixes
- Flexible deployments
Client/Relay/Server model
- Requesting router includes request for prefixes in DHCP configuration request
- Delegating router assigns prefixes in response along with other DHCP configuration information



Prefix/Options Assignment



DHCPv6 Prefix Delegation



```
vpdn enable
!
vpdn-group pppoe
 accept-dialin
 protocol pppoe
 virtual-template 1
!
ipv6 dhcp pool FOO
 prefix-delegation 2001:7:7::/48 0003000100055FAF2C08
 prefix-delegation 2001:8:8::/48 0003000100055FAC1808
 dns-server 2001:4::1
 domain-name cisco.com
!
interface Virtual-Template1
 ipv6 enable
 no ipv6 nd suppress-ra
 ipv6 dhcp server FOO
 ppp authentication chap
!
interface FastEthernet1/0
 pppoe enable
```

```
vpdn enable
!
vpdn-group 1
 request-dialin
 protocol pppoe
!
interface FastEthernet0/1
 ipv6 address DH-PREFIX 0:0:0:1::/64 eui-64
!
interface FastEthernet0/0
 pppoe enable
 pppoe-client dial-pool-number 1
!
interface Dialer1
 encapsulation ppp
 dialer pool 1
 dialer-group 1
 ipv6 address autoconfig
 ipv6 dhcp client pd DH-PREFIX
 ppp authentication chap callin
 ppp chap hostname dhcp
 ppp chap password 7 0300530816
!
ipv6 route ::/0 Dialer1
```

http://www.cisco.com/en/US/tech/tk872/technologies_white_paper09186a00801e199d.shtml



IPv6 Integration & Transition

ISP/IXP Workshops