IPv6 Lanjut

1

Muhammad Zen Samsono Hadi, ST. MSc.

Lab. Komunikasi Digital
Gedung D4 Lt. 1
EEPIS-ITS

Deployment scenarios

- Many ways to deliver IPv6 services to End Users
  - Most important is End to End IPv6 traffic forwarding
- Service Providers and Enterprises may have different deployment needs
- IPv6 over IPv4 tunnels
- Dedicated Data Link layers for native IPv6
  - no impact on IPv4 traffic & revenues
- Dual stack Networks
  - IPv6 over MPLS or IPv4-IPv6 Dual Stack Routers

Media - Interface Identifier

- IEEE interfaces - EUI-64
  - MAC-address: 0050.a218.0c38
  - Interface ID: 250:A2FF:FE18:C38
- P2P links (HDLC, PPP)
  - Interface ID: 50:A218:C00:D
  - 48 bits from the first MAC address in the box + 16 bit interface index.
- IPv4 tunnels
  - Interface ID: ::a.b.c.d

ICMP Informational Messages

- Echo request & reply (same as IPv4)
- Multicast listener discovery messages:
  query, report, done (like IGMP for IPv4):

Type	Code	Checksum
Maximum Response Delay		Reserved
Multicast Address		

Neighbor Discovery

ICMP message types:

- router solicitation
- router advertisement
- neighbor solicitation
- neighbor advertisement
- redirect

Functions performed:

- router discovery
- prefix discovery
- autoconfiguration of address & other parameters
- duplicate address detection (DAD)
- neighbor unreachability detection (NUD)
- link-layer address resolution
- first-hop redirect

Neighbor Discovery Messages

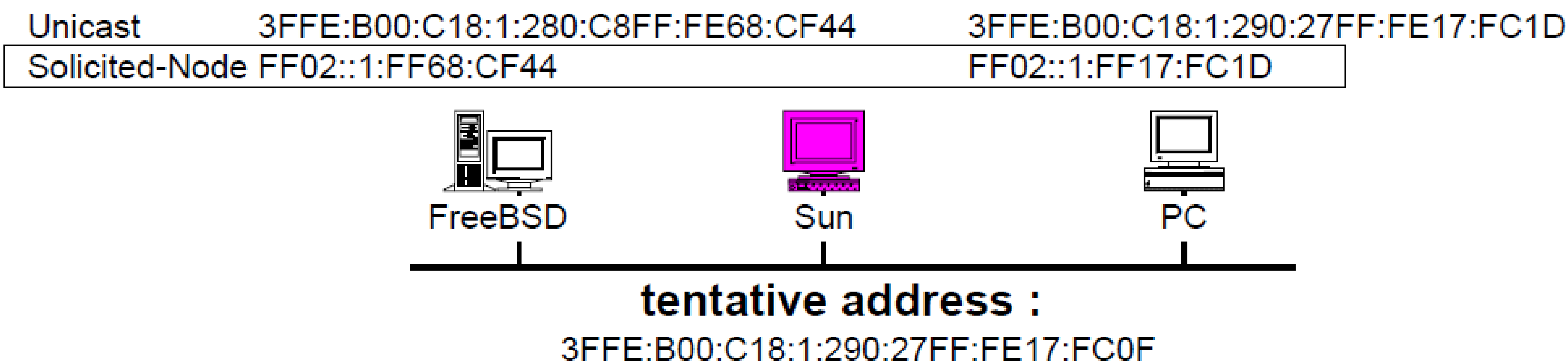
- Router advertisements

  Periodically multicast by router to all-nodes multicast address (link scope)
- Router solicitations
  - sent only at host start-up, to solicit immediate router advert.
  - sent to all-routers multicast address (link scope)
- Neighbor solicitations
  - for address resolution: sent to “solicited node” multicast addr.
  - for unreachability detection: sent to neighbor’s unicast addr.
- Neighbor advertisements
  - for address resolution: sent to unicast address of solicitor
  - for link-layer address change: sent to all-nodes multicast addr.
  - usable for proxy responses (detectable)
  - includes router/host flag

Serverless Autoconfiguration ("Plug-n-Play")

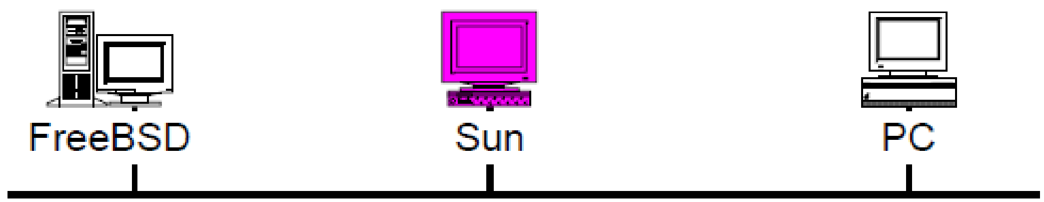
- Hosts generally will construct addresses from RA:
  - subnet prefix(es) learned from periodic multicast advertisements from neighboring router(s)
  - interface IDs generated locally
  - MAC addresses : pseudo-random temporary
- Other IP-layer parameters also learned from router adverts (e.g., router addresses, recommended hop limit, etc.)
- Higher-layer info (e.g., DNS server and NTP server addresses) discovered by multicast / anycast-based service-location protocol [details being worked out]
- DHCP is available for those who want explicit control

IPv6 autoconfiguration



IPv6 autoconfiguration

	FreeBSD	PC
Unicast	3FFE:B00:C18:1:280:C8FF:FE68:CF44	3FFE:B00:C18:1:290:27FF:FE17:FC1D
Solicited-Node	FF02::1:FF68:CF44	FF02::1:FF17:FC1D



tentative address :

3FFE:B00:C18:1:290:27FF:FE17:FC0F

Join FF02::1 (All Nodes)

Join FF02::1:FF17FC0F

Send Neighbor Solicitation to FF02::1:FF17:FC0F
Listen for response on FF02::1

IPv4-IPv6 Transition / Co-Existence

A wide range of techniques have been identified and implemented, basically falling into three categories:

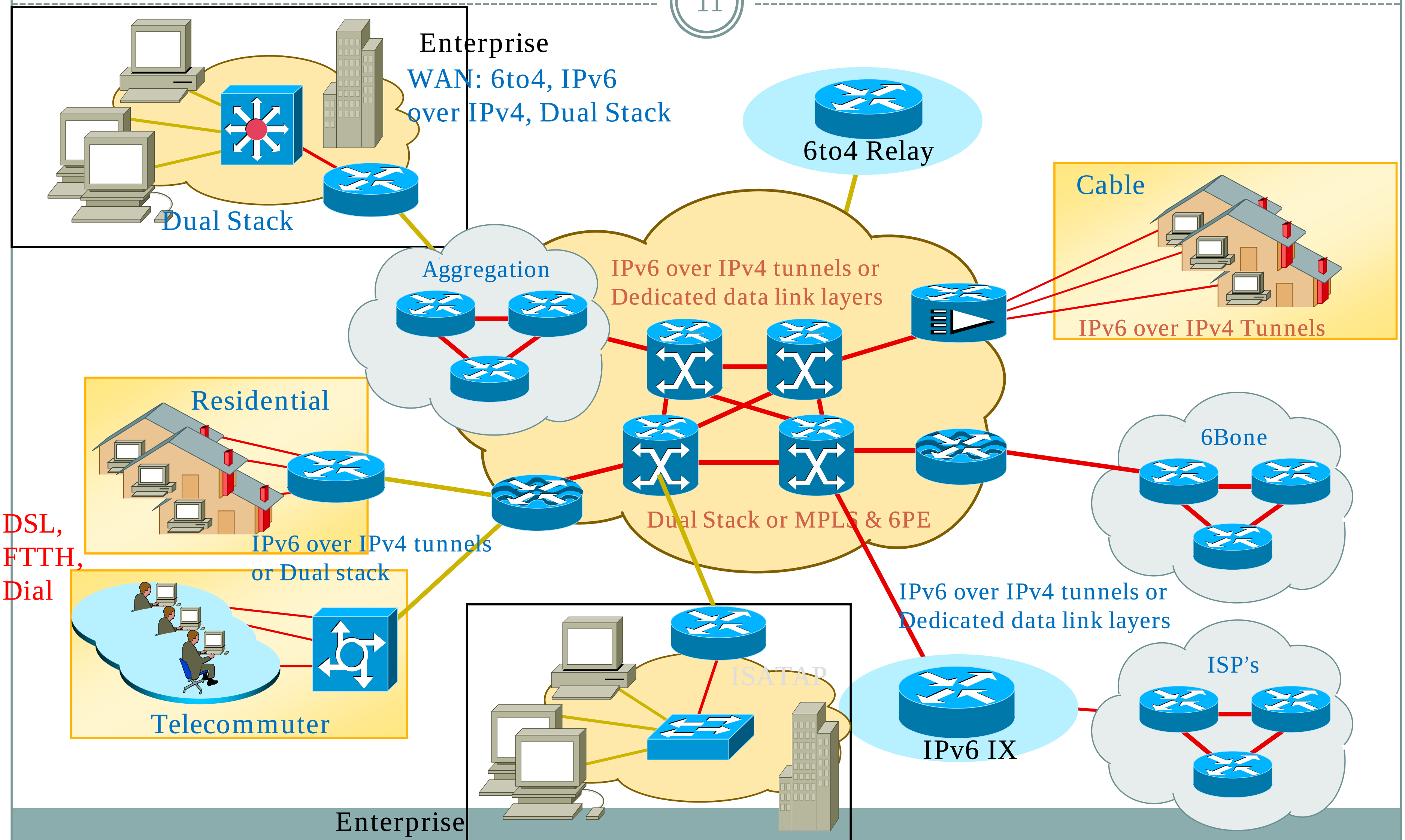
(1) Dual-stack techniques, to allow IPv4 and IPv6 to co-exist in the same devices and networks

(2) Tunneling techniques, to avoid order dependencies when upgrading hosts, routers, or regions

(3) Translation techniques, to allow IPv6-only devices to communicate with IPv4-only devices

Expect all of these to be used, in combination

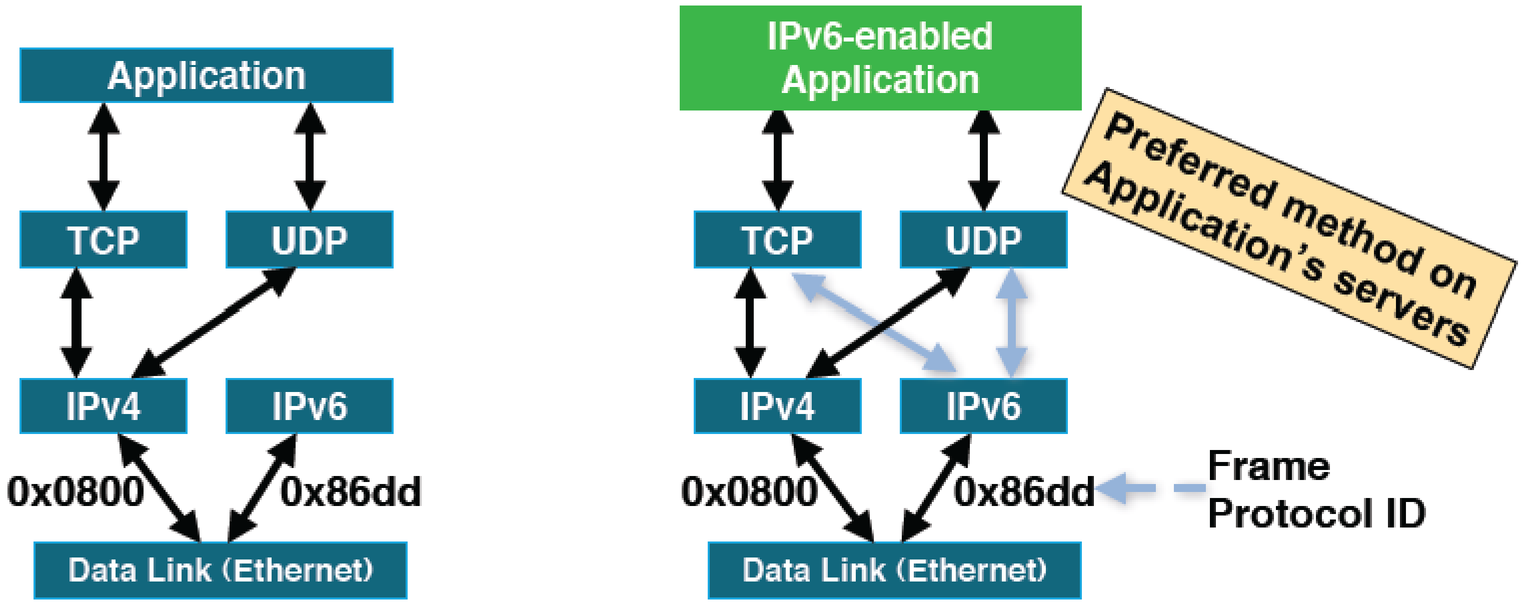
Transition environments



I. Dual-Stack Approach

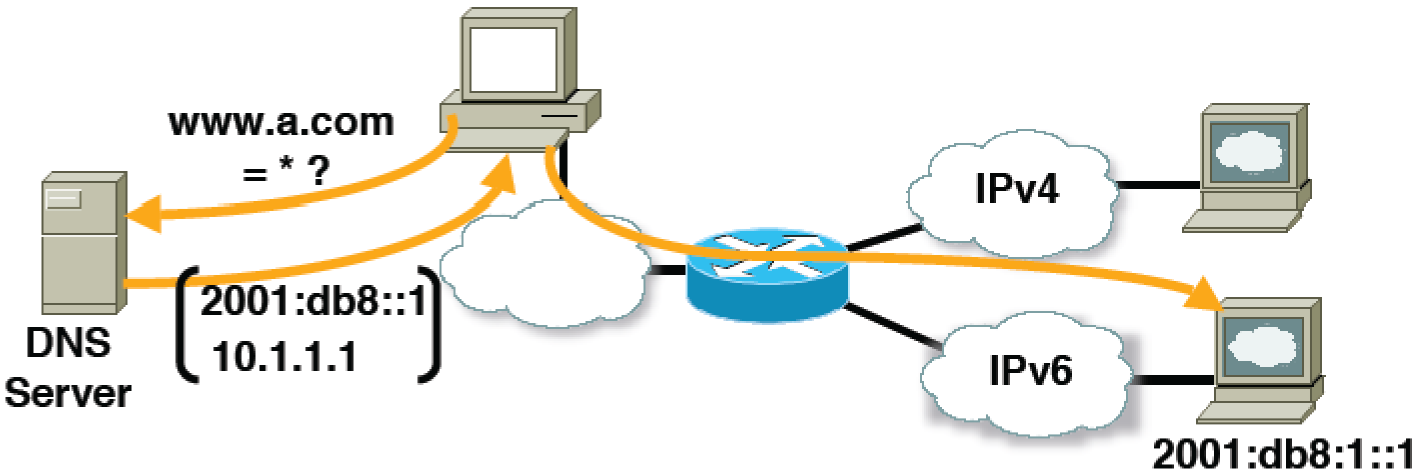
- When adding IPv6 to a system, do not delete IPv4
  - this multi-protocol approach is familiar and well-understood (e.g., for AppleTalk, IPX, etc.)
  - note: in most cases, IPv6 will be bundled with new OS releases, not an extra-cost add-on
- Applications (or libraries) choose IP version to use
  - when initiating, based on DNS response:
    - Prefer scope match first, when equal IPv6 over IPv4
  - when responding, based on version of initiating packet
- This allows indefinite co-existence of IPv4 and IPv6, and gradual app-by-app upgrades to IPv6 usage

Dual Stack Approach



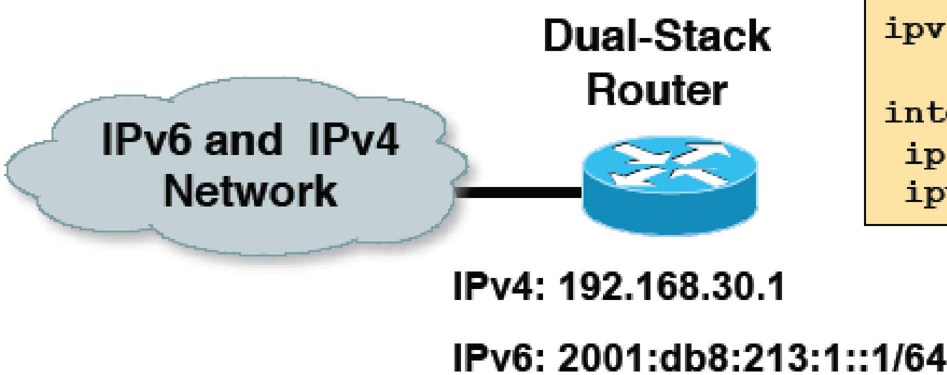
- Dual stack node means:
  - Both IPv4 and IPv6 stacks enabled
  - Applications can talk to both
  - Choice of the IP version is based on name lookup and application preference

Dual Stack & DNS



- On a system running dual stack, an application that is both IPv4 and IPv6 enabled will:
    - Ask the DNS for an IPv6 address (AAAA record)
    - If that exists, IPv6 transport will be used
    - If it does not exist, it will then ask the DNS for an IPv4 address (A record) and use IPv4 transport instead

Sample Dual Stack Configuration




```
router#
ipv6 unicast-routing

interface Ethernet0
 ip address 192.168.30.1 255.255.255.0
 ipv6 address 2001:db8:213:1::1/64
```

- IPv6-enabled router

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

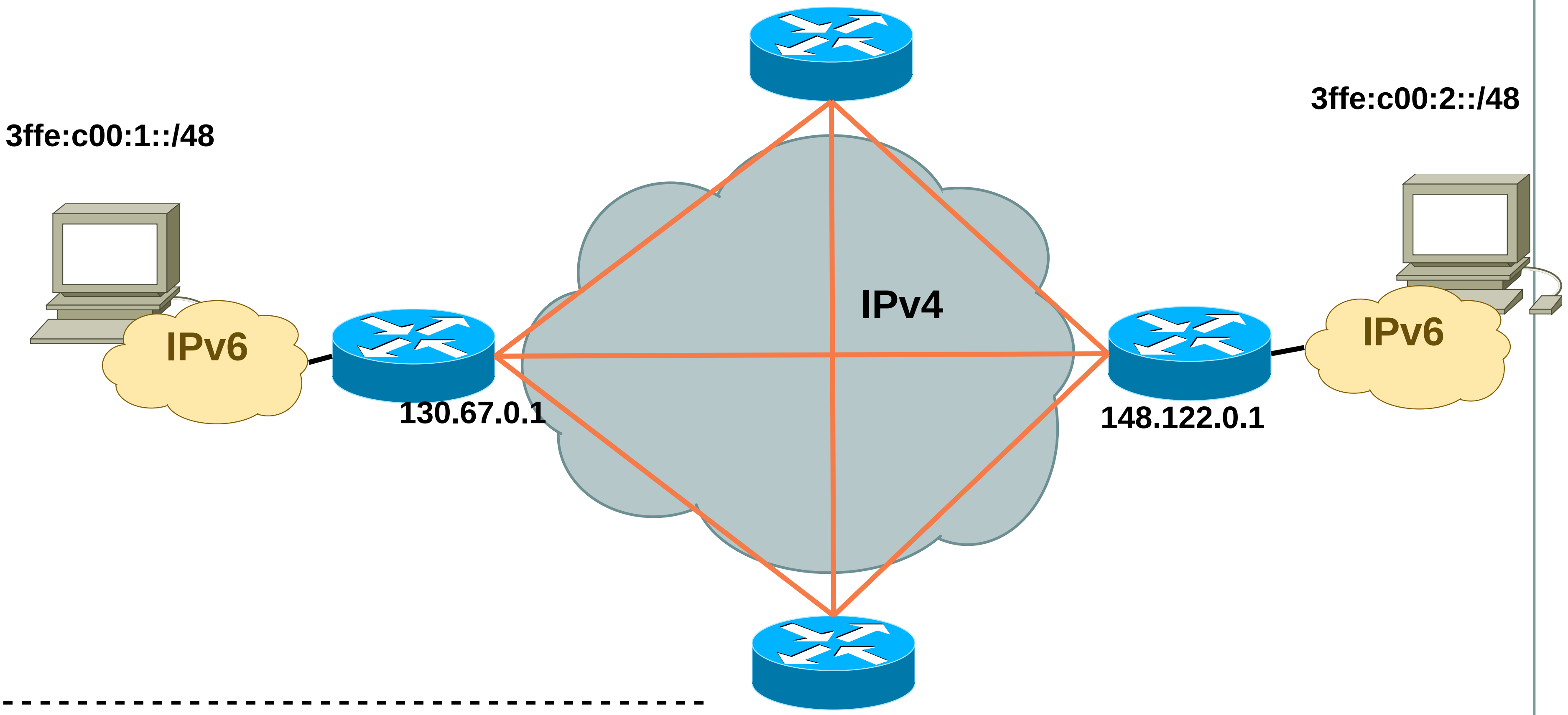
  Telnet, Ping, Traceroute, SSH, DNS client, TFTP etc will all use IPv6 if transport and destination are available

II. Using Tunnels for IPv6 Deployment

- Many techniques are available to establish a tunnel:
  - Manually configured
    - Manual Tunnel (RFC 4213)
    - GRE (RFC 2473)
  - Semi-automated
    - Tunnel broker
  - Automatic
    - 6to4 (RFC 3056)
    - ISATAP (RFC 4214)
    - TEREDO (RFC 4380)

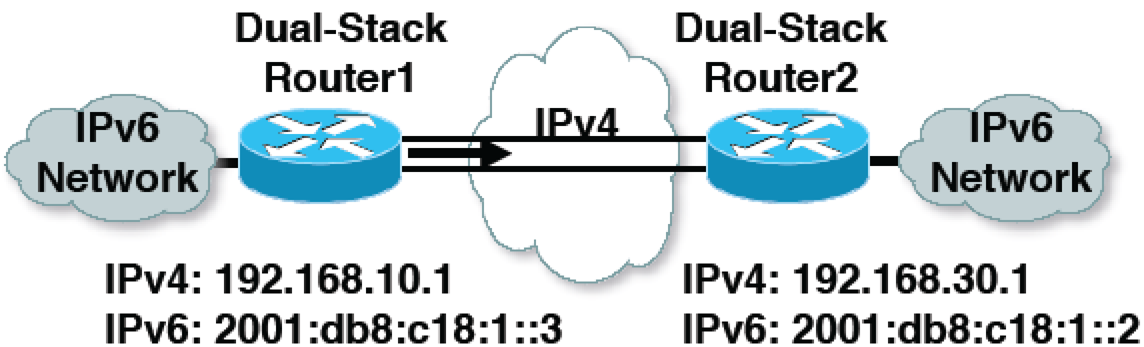
ISATAP & TEREDO are more useful for Enterprises than for Service Providers

Configured tunnels (Manual)




```
--------------------------------------
|IPv4 header|IPv6 header IPv6 payload|
--------------------------------------
IPv4 protocol type = 41
```

Manually Configured Tunnel (RFC4213)

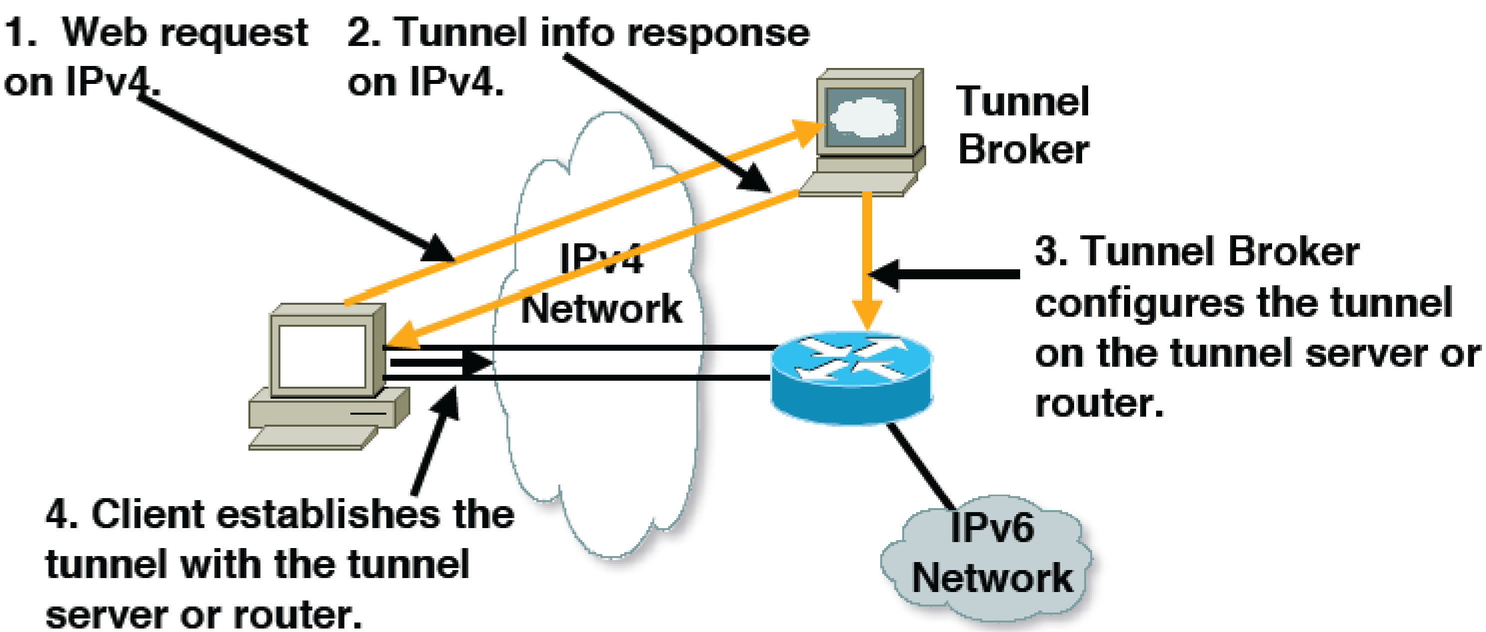



```
router1#

interface Tunnel0
 ipv6 address 2001:db8:c18:1::3/64
 tunnel source 192.168.10.1
 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 192.168.10.1
 tunnel mode ipv6ip
```

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

Tunnel Broker

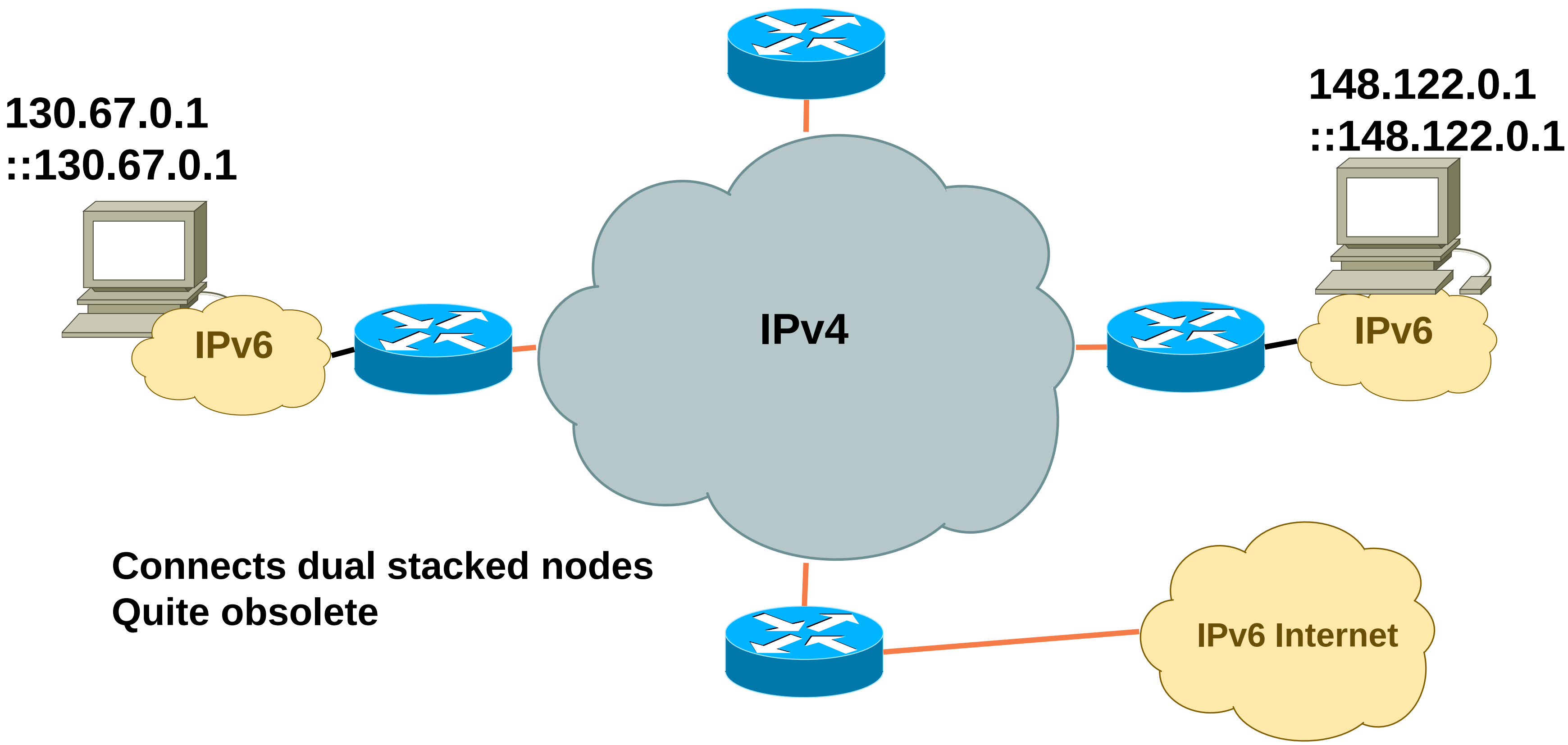


- Tunnel broker:

  Tunnel information is sent via http-ipv4

Automatic tunnels

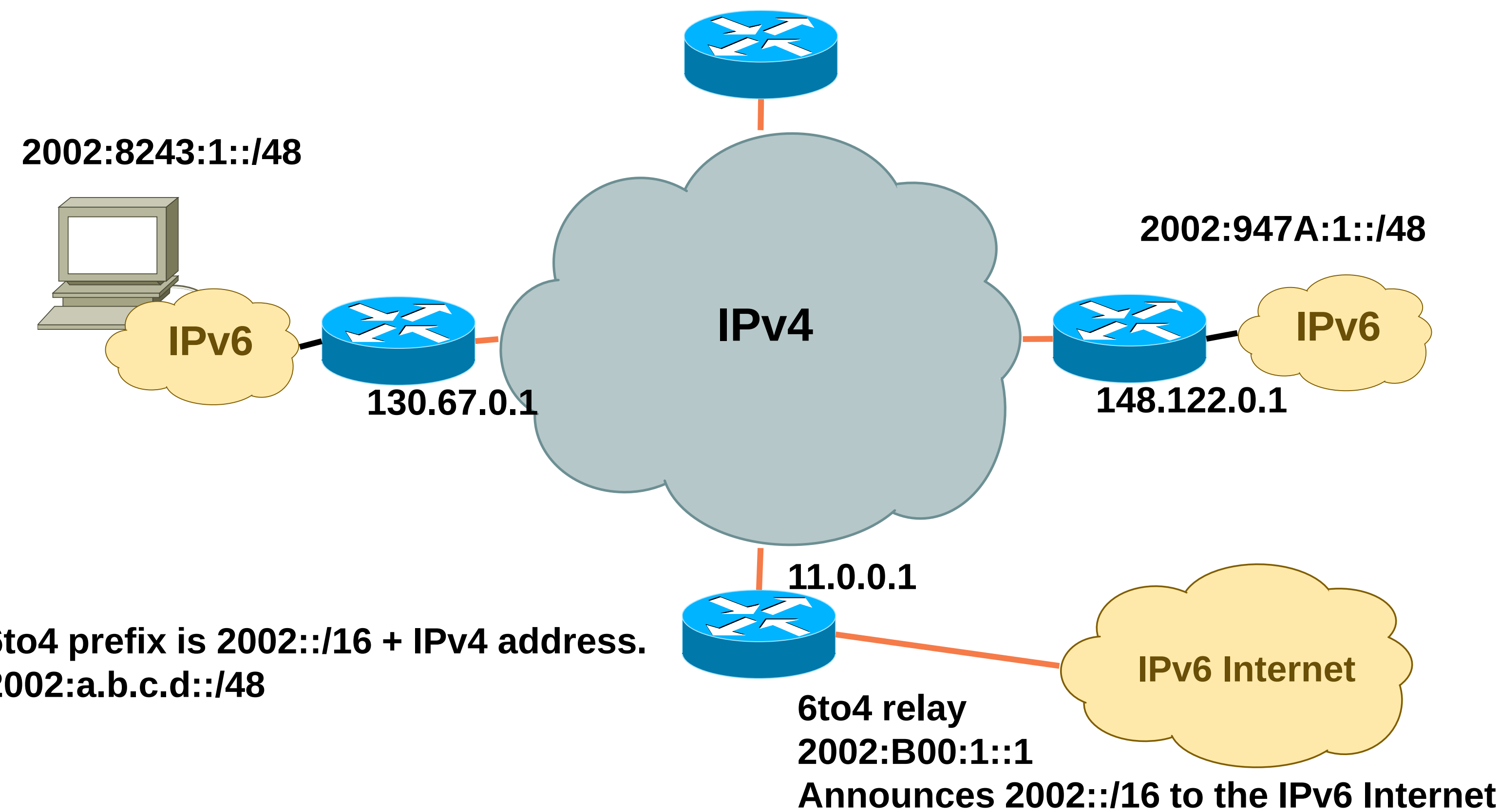
0	IPv4 Address (32bits)
Defined	ISP assigned



Connects dual stacked nodes
Quite obsolete

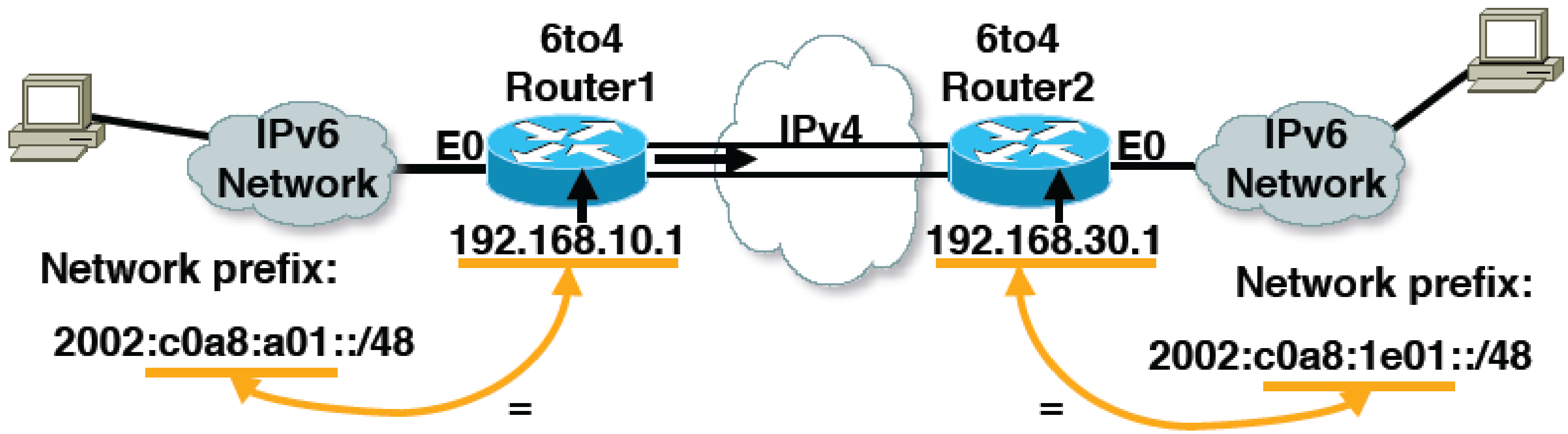
6to4 tunnels

FP (3bits)	TLA (13bits)	IPv4 Address (32bits)	SLA ID (16bits)	Interface ID (64bits)
001	0x0002	ISP assigned	Locally administered	Auto configured



6to4 prefix is 2002::/16 + IPv4 address.
2002:a.b.c.d::/48

6to4 Tunnel (RFC 3056)

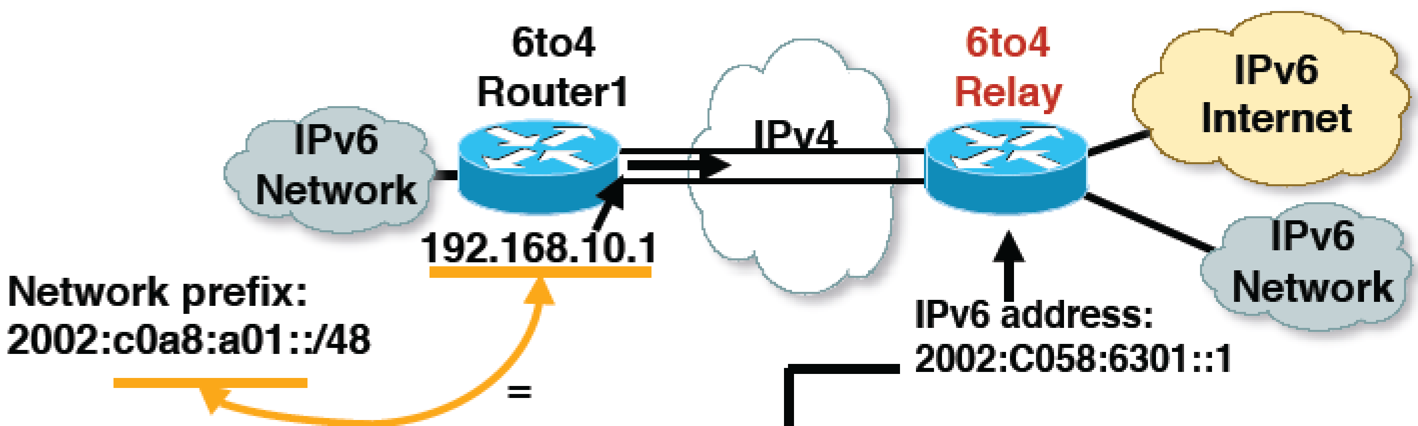


- 6to4 Tunnel:
  - Is an automatic tunnel method
  - Gives a prefix to the attached IPv6 network
  - 2002::/16 assigned to 6to4
  - Requires one global IPv4 address on each Ingress/Egress site

```
router2#

interface Loopback0
 ip address 192.168.30.1 255.255.255.0
 ipv6 address 2002:c0a8:1e01::1/128

interface Tunnel0
 no ip address
 ipv6 unnumbered Ethernet0
 tunnel source Loopback0
 tunnel mode ipv6ip 6to4

ipv6 route 2002::/16 Tunnel0
```


© 2006 Cisco Systems, Inc. All rights reserved.

6to4 Relay




```
router1#

interface Loopback0
 ip address 192.168.10.1 255.255.255.0
 ipv6 address 2002:c0a8:a01::1/128

interface Tunnel0
 no ip address
 ipv6 unnumbered Ethernet0
 tunnel source Loopback0
 tunnel mode ipv6ip 6to4

ipv6 route 2002::/16 Tunnel0
ipv6 route ::/0 2002:c058:6301::1
```

- 6to4 relay:
  - Is a gateway to the rest of the IPv6 Internet
  - Carries 2002:c058:6301::1 IPv6 address
  - Carries 192.88.99.1 IPv4 address
  - Anycast address (RFC 3068) for multiple 6to4 Relay

© 2008 Cisco Systems, Inc. All rights reserved.

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 address of 192.88.99.1

III. NAT-PT for IPv6

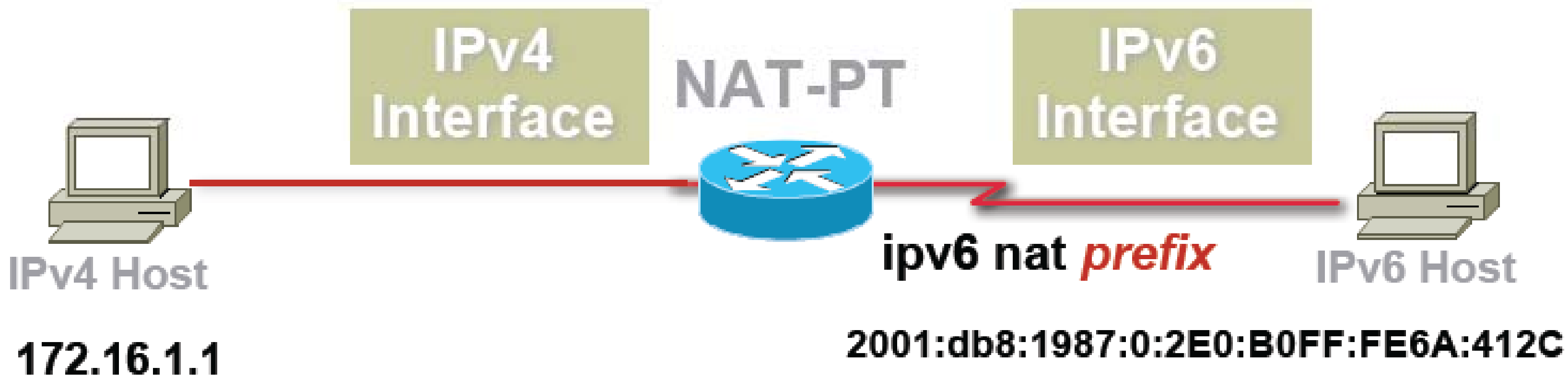
- NAT-PT

  (Network Address Translation – Protocol Translation)

  RFC 2766 & RFC 3596

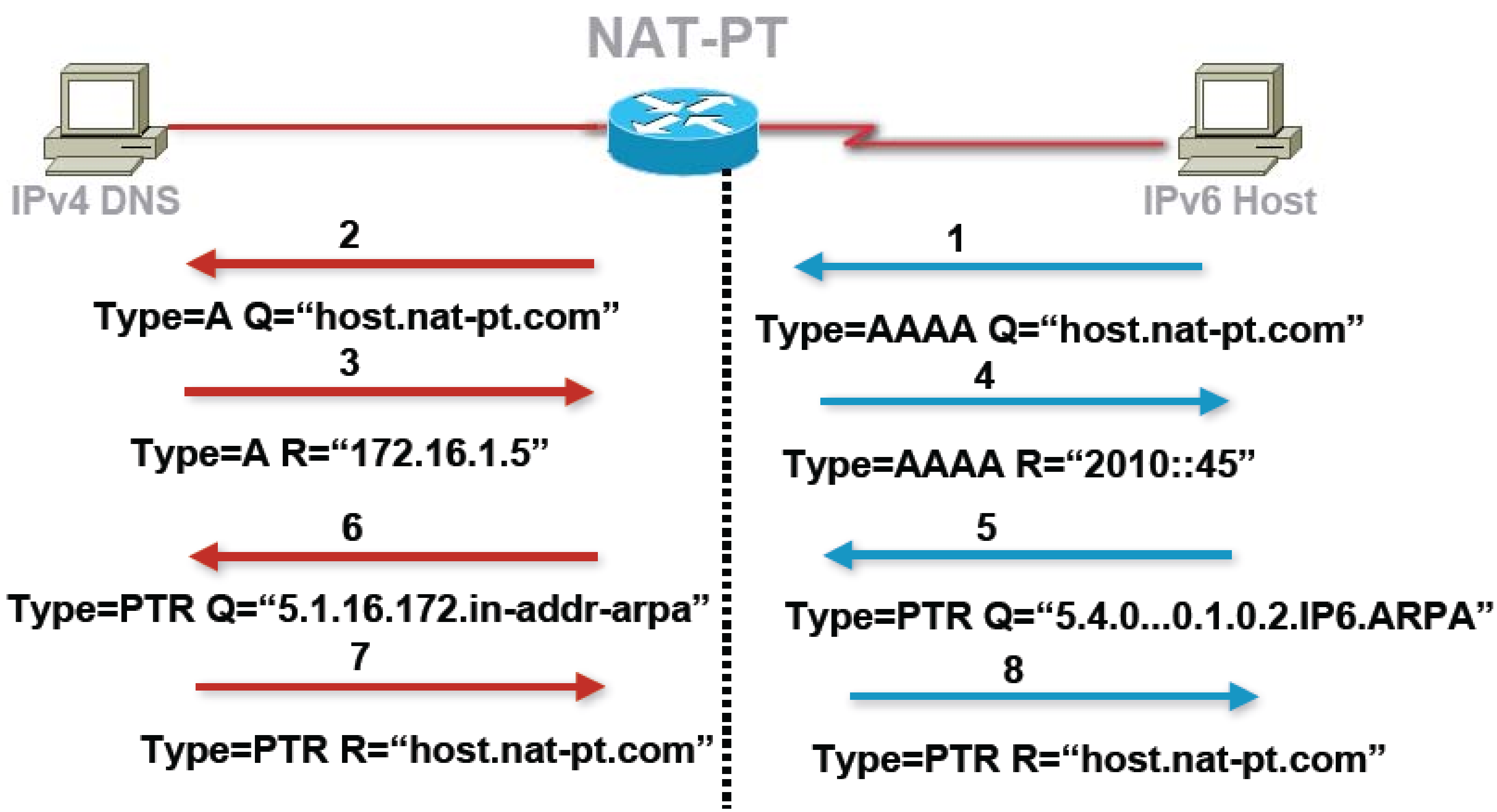
- Allows native IPv6 hosts and applications to communicate with native IPv4 hosts and applications, and vice versa
- Easy-to-use transition and co-existence solution

NAT-PT Concept



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

DNS Application Layer Gateway



Linux 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

Linux Nameserver

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

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

Tells bind to listen on IPv6 ports

Forward zone contains v4 and v6 information

Sets up reverse zone for IPv6 hosts

IPv6 and DNS

	IPv4	IPv6
Hostname to IP address	A record: www.abc.test. A 192.168.30.1	AAAA record: www.abc.test AAAA 2001:db8:c18:1::2
IP address to hostname	PTR record: 1.30.168.192.in-addr.arpa. PTR www.abc.test.	PTR record: 2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.8.b.d.0.1.0.0.2.ip6.arpa PTR www.abc.test.