Issues in IPv6 Deployment

Jeff Doyle
Professional Services
jeff@juniper.net

Objective

A "wide but shallow" overview of the issues, proposed mechanisms, and protocols involved in successfully deploying IPv6

Assumption

- **You attended the morning tutorial on IPv6 basics, or**
- **You already understand IPv6 basics**
  - **Addressing**
  - **Header format**
  - **Extension headers**
  - **ICMPv6 and neighbor discovery**
  - **Address autoconfiguration**

Agenda

- Drivers for IPv6 Deployment
- Routing IPv6
- Multihoming IPv6
- Transition Mechanisms
- Transition Issues

Agenda

- **Drivers for IPv6 Deployment**
- Routing IPv6
- Multihoming IPv6
- Transition Mechanisms
- Transition Issues

Juniper NETWORKS

IPv6 Features

- **Increased address space**
  - **128 bits = 340 trillion trillion trillion addresses**
  - **(2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456)**
  - **= 67 billion billion addresses per cm^2 of the planet surface**
- **Hierarchical address architecture**
  - **Improved address aggregation**
- **More efficient header architecture**
  - **Improved routing efficiency, in some cases**
- **Neighbor discovery and autoconfiguration**
  - **Improved operational efficiency**
  - **Easier network changes and renumbering**
  - **Simpler network applications (Mobile IP)**
- **Integrated security features**

IPv6 Drivers:
IPv4 Address Exhaustion

- **IPv4 addresses particularly scarce in Asia**
  - **Some U.S. universities and corporations have more IPv4 address space than some countries**
- **Imminent demise of IPv4 address space predicted since mid 1990's**
- **NAT + RFC 1918 has slowed that demise**
- **70% of Fortune 1000 companies use NAT***

BUT...

***Source: Center for Next Generation Internet NGI.ORG**

NAT Causes Problems

- Breaks globally unique address model
- Breaks address stability
- Breaks always-on model
- Breaks peer-to-peer model
- Breaks some applications
- Breaks some security protocols
- Breaks some QoS functions
- Introduces a false sense of security
- Introduces hidden costs

IPv6 = plentiful, global addresses = no NAT

IPv6 Drivers: Mobile IP

- **Mobile nodes must be able to move from router to router without losing end-to-end connection**
  - **Home address: Maintains connectivity**
  - **Care-of address: Maintains route-ability**
- **Mobile IP will require millions or billions of care-of addresses**

IPv6 Drivers:
Mobile IP

Current Wireless Subscribers

Region	Number	Regional Percentage
North America	156.6 Million	50.1%
Europe	366.8 Million	57.7%
Japan	72.8 Million	57.3%
Asia Pacific	332.2 Million	10.9%

Sources: U.S. Census Bureau, International Data Corp.

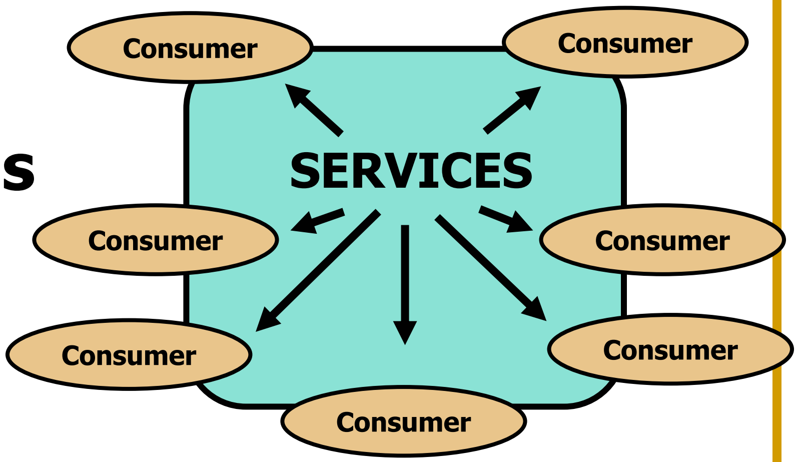
IPv6 Drivers:
Peer-to-Peer Networking

- **"The network is the computer" –Sun Microsystems**
- **Every host is a client and a server**
  - **That is, a consumer and a producer**

P2P:
A group of nodes actively participating in the computing process

IPv6 Drivers:
Peer-to-Peer Networking

- The Internet has evolved into a "Services in the Middle" model
- Information and services flow primarily toward the user
- Contributing factors:
  - Commercial interests
  - Legacy of low-powered PCs
  - NAT breaks network transparency



IPv6 Drivers:
Peer-to-Peer Networking

- **Content sharing**
  - **Napster was a wake-up call**
  - **Kazaa**
  - **Morpheus, FreeNet, Grokster, Gnutella, many more…**
- **Distributed data processing**
  - **SETI@home**
  - **Folding@home**
  - **Popular Power**
  - **United Devices**
- **Distributed applications**
  - **Black-hat hackers already appreciate this (DDoS)**

IPv6 Drivers:
Peer-to-Peer Networking

- **Online gaming will be an early driver**
- **Current gaming market in U.S. $210M**
  - **$1.8B by 2005* (>100% PA growth)**
  - **Gamers account for 10% of U.S. broadband market****
  - **¥271B ($2.2B) industry in Japan by 2006*****
  - **114 million gamers online by 2006******
- **Millions of on-line gamers in Japan and Korea**
- **Microsoft investing $2B in XBox Live**
- **Present online gaming mostly client/server**
  - **Forced by insufficient IPv4 addresses**
  - **Creates bandwidth bottlenecks**

* Source: NCSoft
**Source: ISP-Planet.com
***Source: Nomura Research Institute
****Source: DFC Intelligence

IPv6 Drivers:
Internet-Enabled Devices

- **Internet-enabled appliances**
  - **Electrolux Screenfridge**
  - **Samsung Digital Network Refrigerator**

- **Internet-enabled automobiles**
  - **Already available in many luxury cars**
  - **Interesting research being conducted in Japan**

IPv6 Drivers:
Internet-Enabled Devices

- Internet-enabled ATMs
  - Fujitsu Series 8000
  - Infonox, Western Union conducting pilot program

- Smart sensors
- Bioelectronics

IPv6 Drivers:
Conclusion

- The common factor in all cases is:

MORE IP ADDRESSES

- For billions of new users
- For billions of new devices
- For always-on access
- For transparent Internet connectivity the way it was meant to be

Agenda

- Drivers for IPv6 Deployment
- **Routing IPv6**
- Multihoming IPv6
- Transition Mechanisms
- Transition Issues

MTU Path Discovery

- IPv6 routers do not fragment packets
- IPv6 MTU must be at least 1280 bytes
  - Recommended MTU: 1500 bytes
- Nodes should implement MTU PD
  - Otherwise they must not exceed 1280 bytes
- MTU path discovery uses ICMP "packet too big" error messages

Configuration Example:
Static Route

```
[edit routing-options]
ps@R1# show
rib inet6.0 {
    static {
        route 3ffe::/16 next-hop 2001:468:1100:1::2;
    }
}
```

RIPng

- RFC 2080 describes RIPngv1, not to be confused with RIPv1
- Based on RIP Version 2 (RIPv2)
- Uses UDP port 521
- Operational procedures, timers and stability functions remain unchanged
- RIPng is not backward compatible to RIPv2
- Message format changed to carry larger IPv6 addresses

Configuration Example:
RIPng

```
[edit protocols]
lab@Juniper5# show
ripng {
    group external_neighbors {
        export default_route;
        neighbor ge-0/0/0.0;
        neighbor ge-0/0/1.0;
        neighbor ge-0/0/2.0;
    }
    group internal_neighbors {
        export external_routes;
        neighbor ge-1/0/0.0;
    }
}
```

IS-IS

- **draft-ietf-isis-ipv6-02.txt, Routing IPv6 with IS-IS**
- **2 new TLVs are defined:**
  - **IPv6 Reachability (TLV type 236)**
  - **IPv6 Interface Address (TLV type 232)**
- **IPv6 NLPID = 142**

Configuration Example:
IS-IS for IPv6 Only

- **By default, IS-IS routes both IPv4 and IPv6**

```
lab@Juniper5# show
isis {
    no-ipv4-routing;
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
}
```

OSPFv3

- **Unlike IS-IS, entirely new version required**
- **RFC 2740**
- **Fundamental OSPF mechanisms and algorithms unchanged**
- **Packet and LSA formats are different**

OSPFv3 Differences from OSPFv2

- Runs per-link rather than per-subnet
  - Multiple instances on a single link
- More flexible handling of unknown LSA types
- Link-local flooding scope added
  - Similar to flooding scope of type 9 Opaque LSAs
  - Area and AS flooding remain unchanged
- Authentication removed
- Neighboring routers always identified by RID
- Removal of addressing semantics
  - IPv6 addresses not present in most OSPF packets
  - RIDs, AIDs, and LSA IDs remain 32 bits

OSPFv3 LSAs

Type	Description
0x2001	Router-LSA
0x2002	Network-LSA
0x2003	Inter-Area-Prefix-LSA
0x2004	Inter-Area-Router-LSA
0x2005	AS-External-LSA
0x2006	Group-Membership-LSA
0x2007	Type-7-LSA (NSSA)
0x2008	Link-LSA
0x2009	Intra-Area-Prefix-LSA

Configuration Example:
OSPFv3

```
[edit protocols]
lab@Juniper5# show
ospf3 {
    area 0.0.0.0 {
        interface ge-1/1/0.0;
    }
    area 192.168.1.2 {
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
    }
}
```

Multiprocotol BGP-4

- MBGP defined in RFC 2283
- Two BGP attributes defined:
  - Multiprotocol Reachable NLRI advertises arbitrary Network Layer Routing Information
  - Multiprotocol Unreachable NLRI withdraws arbitrary Network Layer Routing Information
  - Address Family Identfier (AFI) specifies what NLRI is being carried (IPv6, IP Multicast, L2VPN, L3VPN, IPX…)
- Use of MBGP extensions for IPv6 defined in RFC 2545
  - IPv6 AFI = 2
- BGP TCP session can be over IPv4 or IPv6
- Advertised Next-Hop address must be global or site-local IPv6 address
  - And can be followed by a link-local IPv6 address
  - Resolves conflicts between IPv6 rules and BGP rules

Example Configuration:
BGP

```
[edit protocols]
lab@Juniper5# show
bgp {
    group IPv6_external {
        type external;
        import v6_externals;
        family inet6 {
            unicast;
        }
        export v6_routes;
        peer-as 65502;
        neighbor 3ffe:1100:1::b5;
    }
    group IPv6_internal {
        type internal;
        local-interface lo0.0;
        family inet6 {
            unicast;
        }
        neighbor 2001:88:ac3::51;
        neighbor 2001:88:ac3::75;
    }
}
```

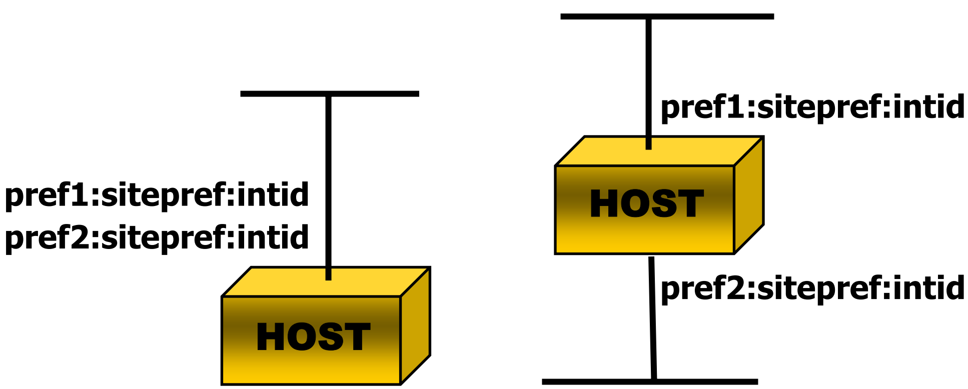
Agenda

- Drivers for IPv6 Deployment
- Routing IPv6
- **Multihoming IPv6**
- Transition Mechanisms
- Transition Issues

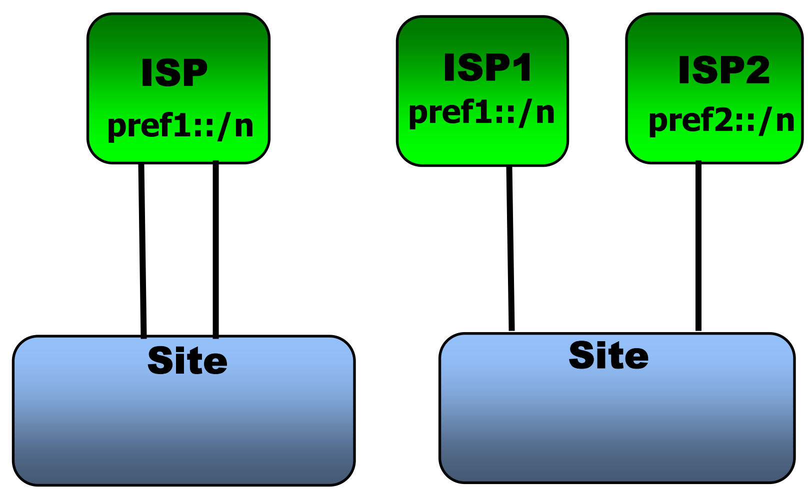
What is Multihoming?

- **Host multihoming**
  - **More than one unicast address on an interface**
  - **Interfaces to more than one network**
- **Site multihoming**
  - **Multiple connections to the same ISP**
  - **Connections to multiple ISPs**

Host Multihoming



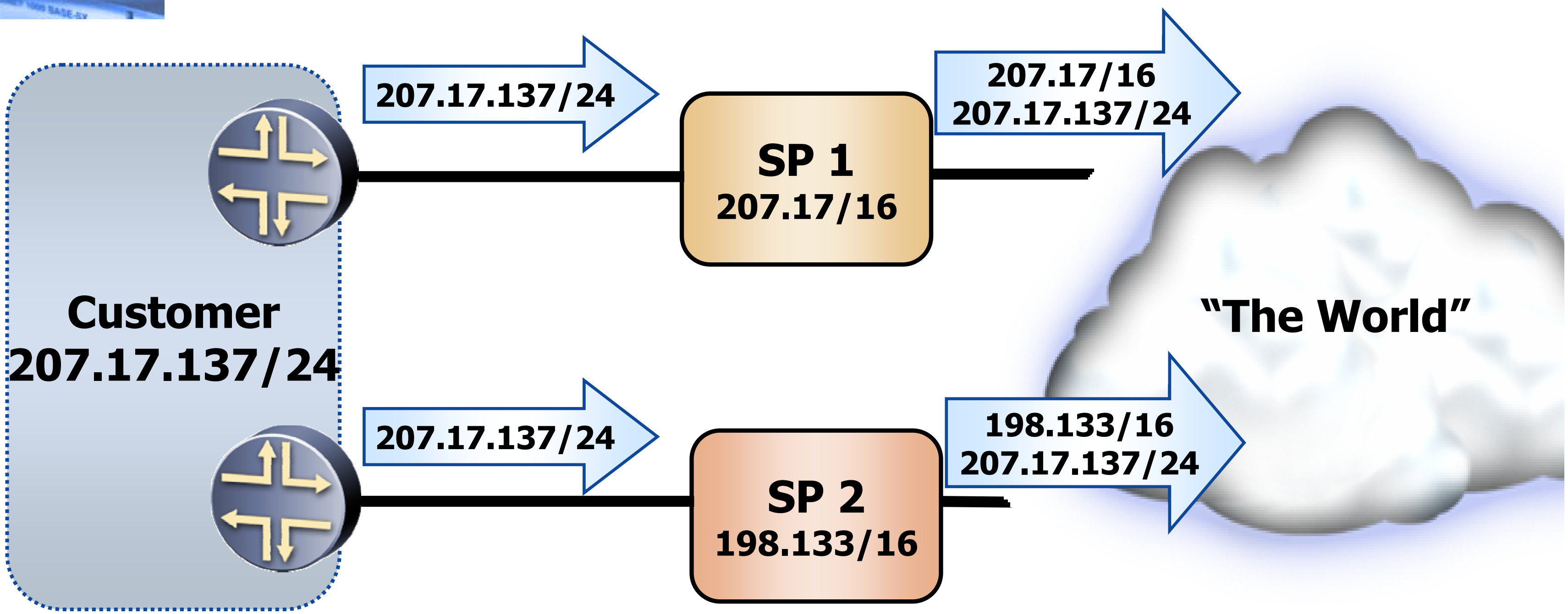
Site Multihoming



Why Multihome?

- Redundancy
  - Against router failure
  - Against link failure
  - Against ISP failure
- Load sharing
- Local connectivity across large geography
- Corporate or external policies
  - Acceptable use policies
  - Economics

The Multihoming Problem



- ISP2 must advertise additional prefix
- ISP1 must "punch a hole" in its CIDR block
- Contributes to routing table explosion
- Contributes to Internet instability
  - Due to visibility of customer route flaps
  - Due to increased convergence time
- Same problem can apply to provider-independent (PI) addresses

IPv6 and The Multihoming Problem

- IPv6 does not have a set solution to the problem
- Currently, 6Bone disallows IPv4-style multihoming (RFC 2772)
  - ISPs cannot advertise prefixes of other ISPs
  - Sites cannot advertise to upstream providers prefixes longer than their assigned prefix
- However, IPv6 offers the possibility of one or more solutions
  - Router-based solutions
  - Host-based solutions
  - Mobile-based solutions
  - Geographic or Exchange-based solutions

Multihoming Requirements

Requirements for IPv6 Site-Multihoming Architectures
(draft-ietf-multi6-multihoming-requirements-03)

- **Must support redundancy**
- **Must support load sharing**
- **Protection from performance difficulties**
- **Support for multihoming for external policy reasons**
- **Must not be more complex than current IPv4 solutions**
- **Re-homing transparency for transport-layer sessions (TCP, UDP, SCTP)**
- **No impact on DNS**
- **Must not preclude packet filtering**
- **Must scale better than IPv4 solutions**
- **Minor impact on routers**
- **No impact on host connectivity**
- **May involve interaction between hosts and routers**
- **Must be manageable**
- **Must not require cooperation between transit providers**

Possible Solution #1: Do Nothing

- **Allow Internet default free zone (DFZ) to continue to grow**
- **Put responsibility on router vendors to keep increasing memory, performance to compensate**

Pros:
- As simple as it gets
- No special designs, policies, or mechanisms needed

Cons:
- Does nothing to increase Internet stability
- Large routing tables = Large convergence times
- No guarantee vendors can continue to stay ahead of the curve

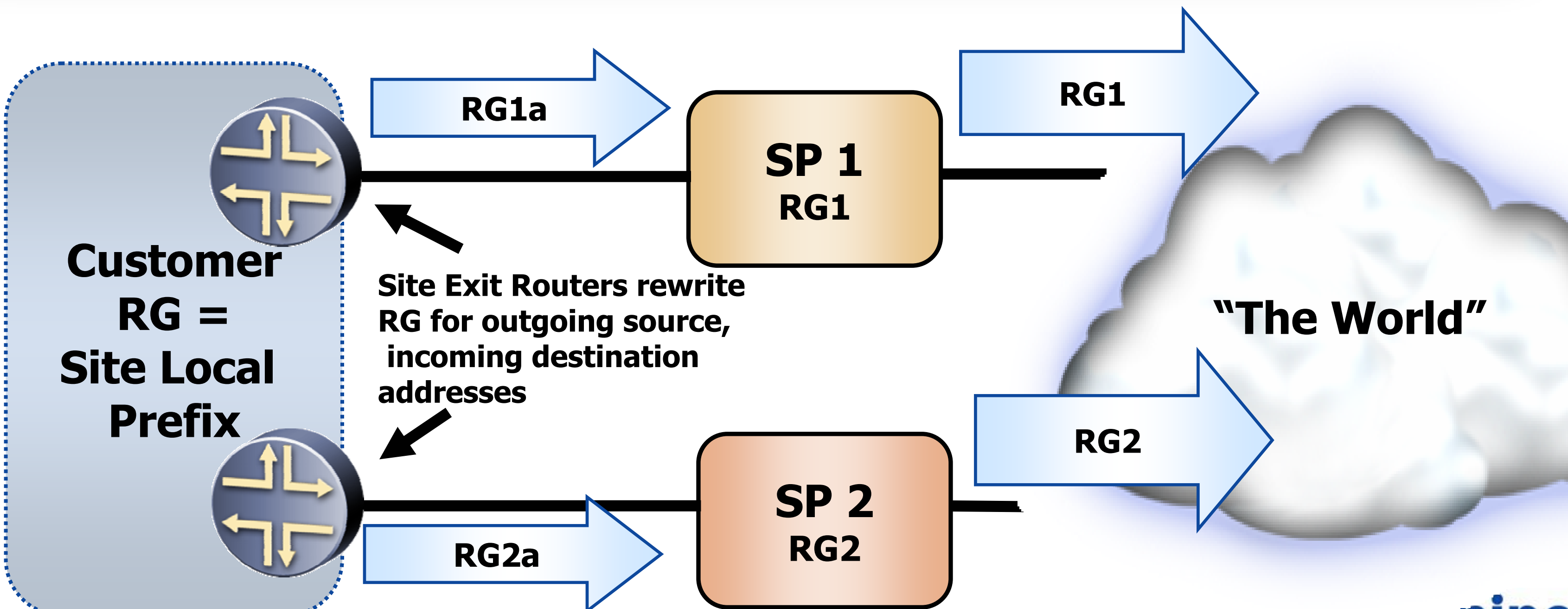
Possible Solution #2: GSE/8+8

GSE: Global, Site, and End System Address Elements
(draft-ipng-gseaddr-00.txt)
(draft-ietf-ipngwg-esd-analysis-05.txt)

- **Router-based solution**
- **Key concepts:**
  - **Distinct separation of Locator and Identifier entities in IPv6 addresses**
  - **Rewriting of locator (Routing Goop) at Site Exit Router**
  - **Identifier (End System Designator) is globally unique**
  - **DNS AAA records and RG records**

Possible Solution #2: GSE/8+8

6+ Bytes	~2 Bytes	8 Bytes
Global Routing Goop (RG)	Site Topology Partition (STP)	End System Designator (ESD)
Locator		Identifier



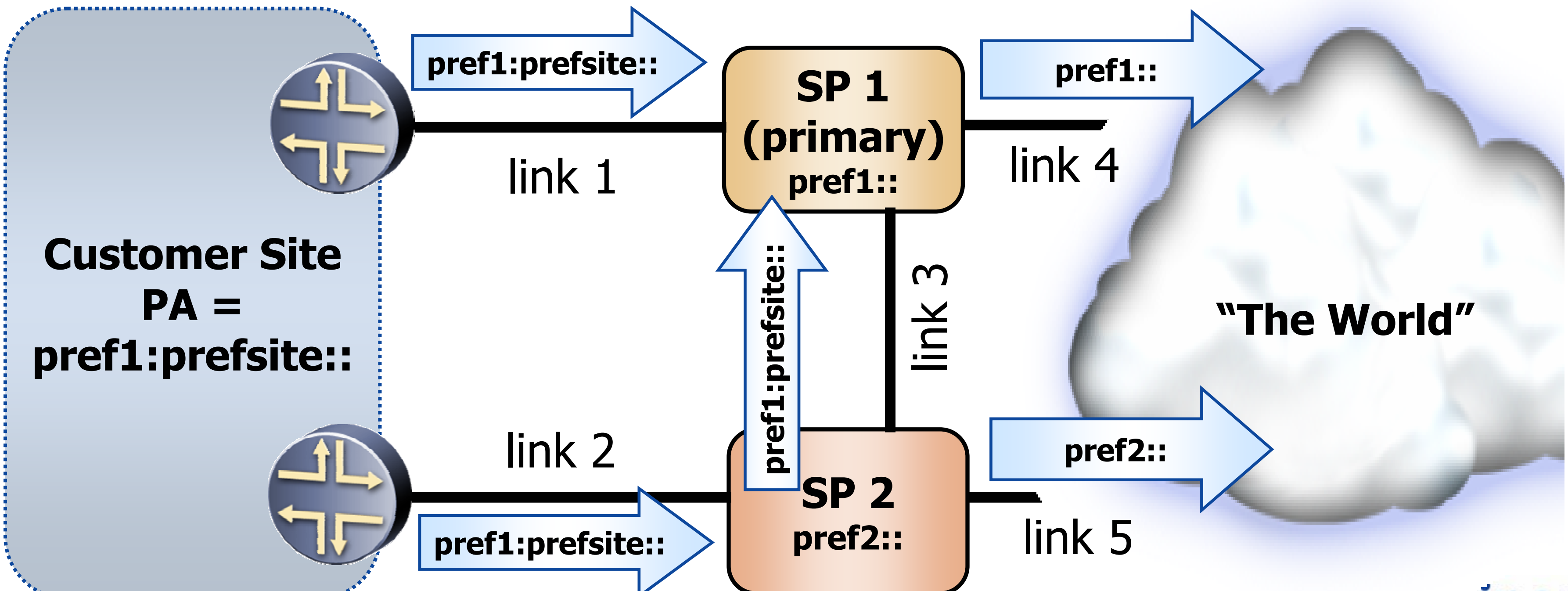
Possible Solution #2: GSE/8+8

- **GSE as proposed rejected by IPng WG in 1997**
  - **Thought to introduce more problems than it solved**
    - **"Separating Identifiers and Locators in Addresses: An Analysis of the GSE Proposal for IPv6" (draft-ietf0ipngwg-esd-analysis-04.txt)**
  - **But, concept is still being discussed**

Possible Solution #3: Multihoming with Route Aggregation

(draft-ietf-ipngwg-ipv6multihome-with-aggr-01.txt)

- Router-based solution
- Customer site gets PA from primary ISP
- PA advertised to both ISPs, but not upstream
- PA advertised from ISP2 to ISP1



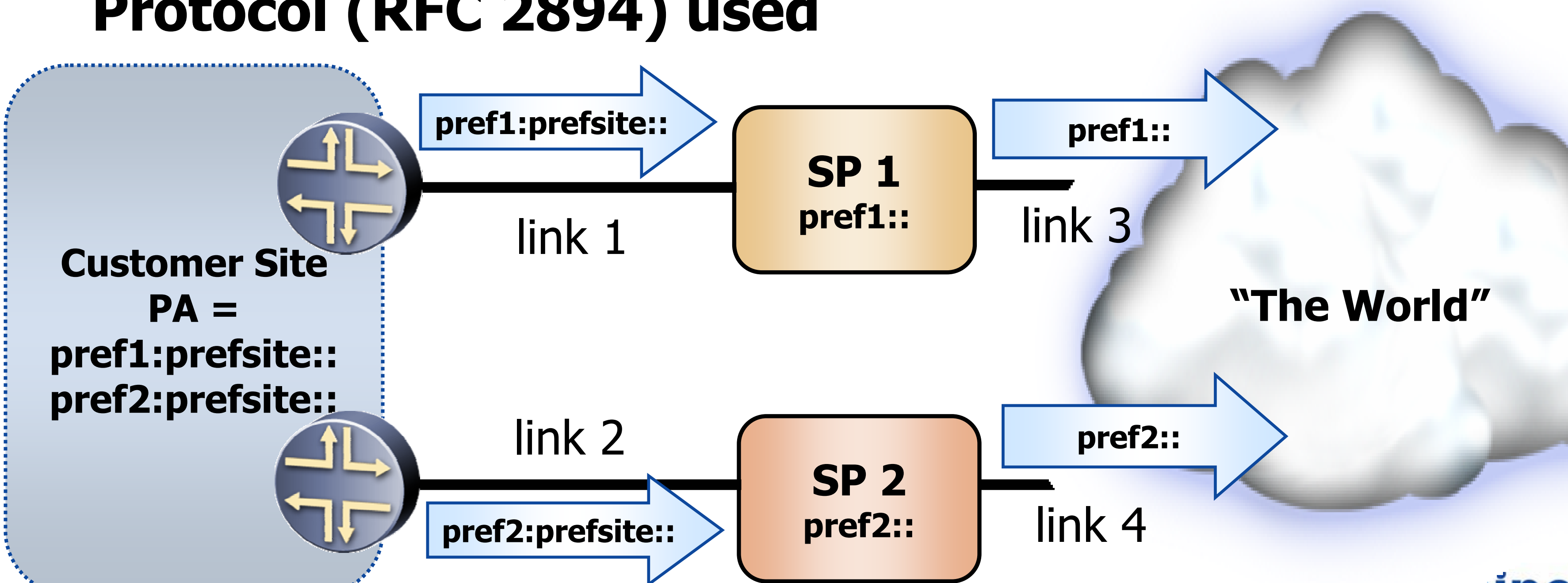
Possible Solution #3: Multihoming with Route Aggregation

- **Pros:**
  - No new protocols or modifications needed
  - Fault tolerance for links 1 and 2
  - Load sharing with ISPs 1 and 2
  - Link failure does not break established TCP sessions
- **Cons:**
  - No fault tolerance if ISP1 or link 4 fails
  - No load sharing if link 3 fails
  - Problematic if link 3 must pass through intermediate ISP
  - Assumes ISP1 and ISP2 are willing to provide link 3 and appropriate route advertisements

Possible Solution #4: Multihoming Using Router Renumbering

(draft-ietf-ipngwg-multi-isp-00.txt)

- Router-based solution
- All customer device interfaces carry addresses from each ISP
- Router Advertisements and Router Renumbering Protocol (RFC 2894) used



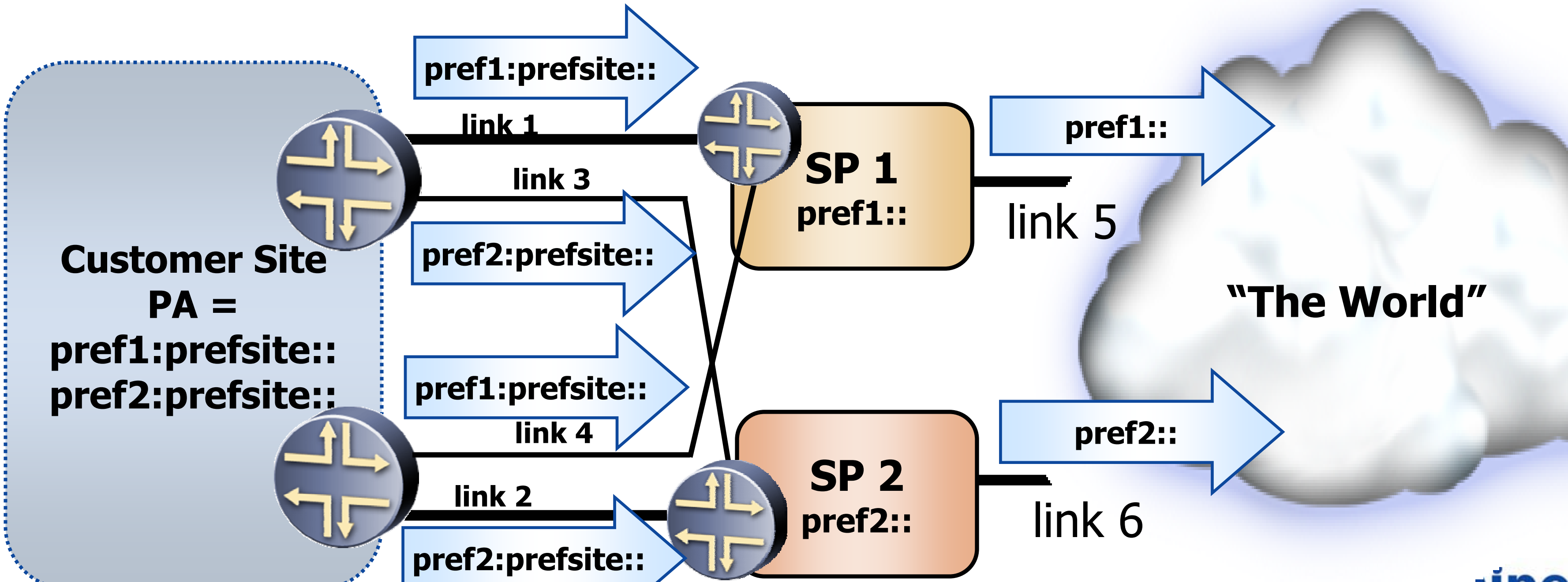
Possible Solution #4: Multihoming Using Router Renumbering

- **If an ISP fails:**
  - Site border router detecting failure sends RAs to deprecate ISP's delegated addresses
  - Router Renumbering Protocol propagates information about deprecation to internal routers
- **Pros:**
  - No new protocols or modifications needed
  - Fault tolerance for both links and ISPs
- **Cons:**
  - No clear criteria for selecting among multiple interface addresses
  - No clear criteria for load sharing among ISPs
  - Link or ISP failure breaks established TCP sessions

Possible Solution #4: Multihoming Support at Site Exit Routers

(RFC 3178)

- Router-based solution
- Links 3 and 4 (IP in IP tunnels) configured as secondary links
- Primary and secondary links on separate physical media for link redundancy
- Prefixes advertised over secondary links have weak preference relative to prefixes advertised over primary links



Possible Solution #4: Multihoming Support at Site Exit Routers

- **Pros:**
  - **No new protocols or modifications needed**
  - **Link fault tolerance**
  - **Link failure does not break established TCP sessions**
- **Cons:**
  - **No fault tolerance if ISP fails**
  - **No clear criteria for selecting among multiple interface addresses**
  - **No clear criteria for load sharing among ISPs**

Possible Solution #5: Host-Centric IPv6 Multihoming

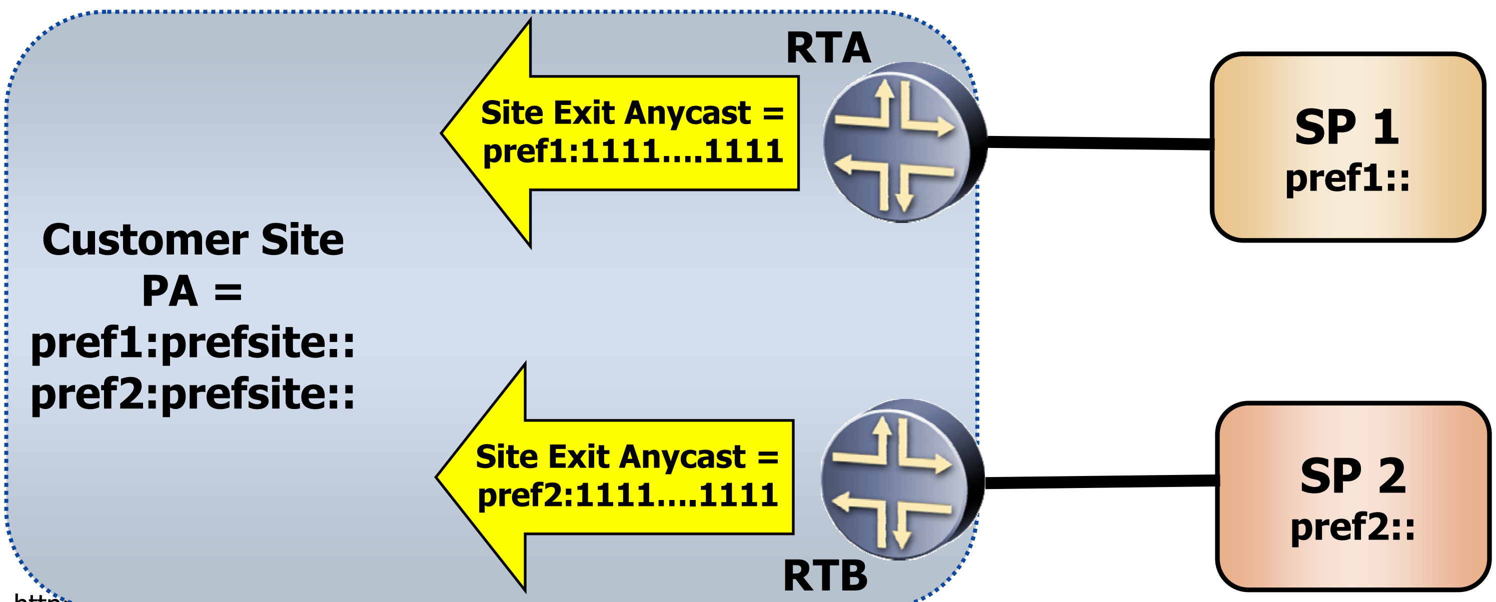
(draft-huitema-multi6-hosts-01.txt)

- Host- *and* router-based solution
- Key Concepts:
  - Multiple addresses per host interface
  - Site exit router discovery
  - Site exit anycast address
  - Site exit redirection
    - New Site Exit Redirection ICMP message defined

Possible Solution #5: Host-Centric IPv6 Multihoming

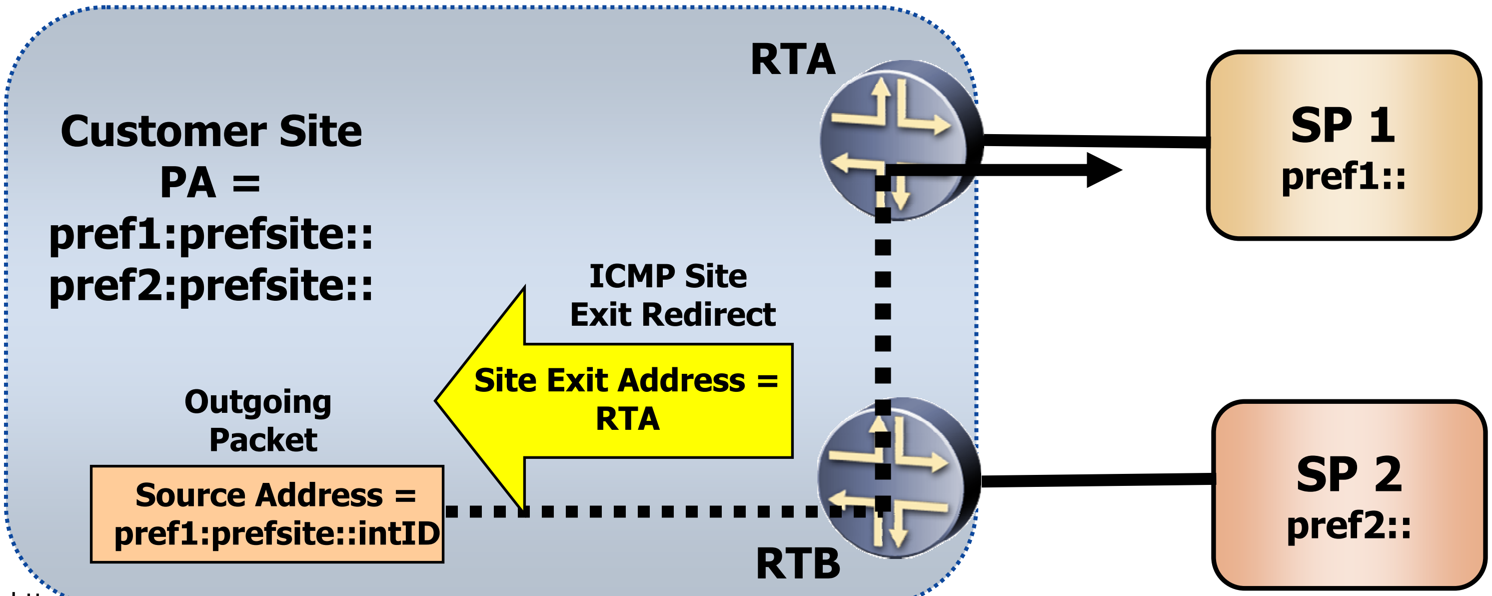
- Site anycast address indicates site exit address
- Site anycast address advertised via IGP
- Hosts tunnel packets to selected site exit router

L bits	128 – L bits
Site Prefix	All Ones (1111…………………………………………1111)



Possible Solution #5: Host-Centric IPv6 Multihoming

- **Site redirection:**
  1. Tunnels created between all site exit routers
  2. Source address of outgoing packets examined
  3. Packet tunneled to correct site exit router
  4. Site exit redirect sent to host



Possible Solution #5: Host-Centric IPv6 Multihoming

- **Pros:**
  - **Fault tolerant of link, router, and ISP failure**
  - **Overcomes problem of ingress source address filtering at ISPs**
- **Cons:**
  - **Requires new ICMP message**
  - **Requires modification to both routers and hosts**
  - **Tunneling can become complex**
    - **Between site exit routers**
    - **Hosts to all site exit routers**

And Many Other Proposed Solutions…

- Extension Header for Site Multihoming Support
  - (draft-bagnulo-multi6-mhExtHdr-00.txt)
- Host Identity Payload Protocol (HIP)
- Exchange-Based Aggregation
- Multihoming Aliasing Protocol (MHAP)
  - (draft-py-mhap-01a.txt)
- Provider-Internal Aggregation Based on Geography to Support Multihoming in IPv6
  - (draft-van-beijnum-multi6-isp-int-aggr-00.txt)
- GAPI: A Geographically Aggregatable Provider Independent Address Space to Support Multihoming in IPv6
  - (draft-py-multi6-gapi-00.txt)
- An IPv6 Provider-Independent Global Unicast Address Format
  - (draft-hain-ipv6-pi-addr-03.txt)

Other IPv6 Multihoming Issues

- How does a host choose between multiple source and destination addresses?
  - See draft-ietf-ipv6-default-addr-select-09
- How are DNS issues resolved?
  - See RFC 2874, "DNS Extensions to Support IPv6 Address Aggregation and Renumbering," section 5.1, for DNS proposals for multihoming

Agenda

- Drivers for IPv6 Deployment
- Routing IPv6
- Multihoming IPv6
- **Transition Mechanisms**
- Transition Issues

Transition Assumptions

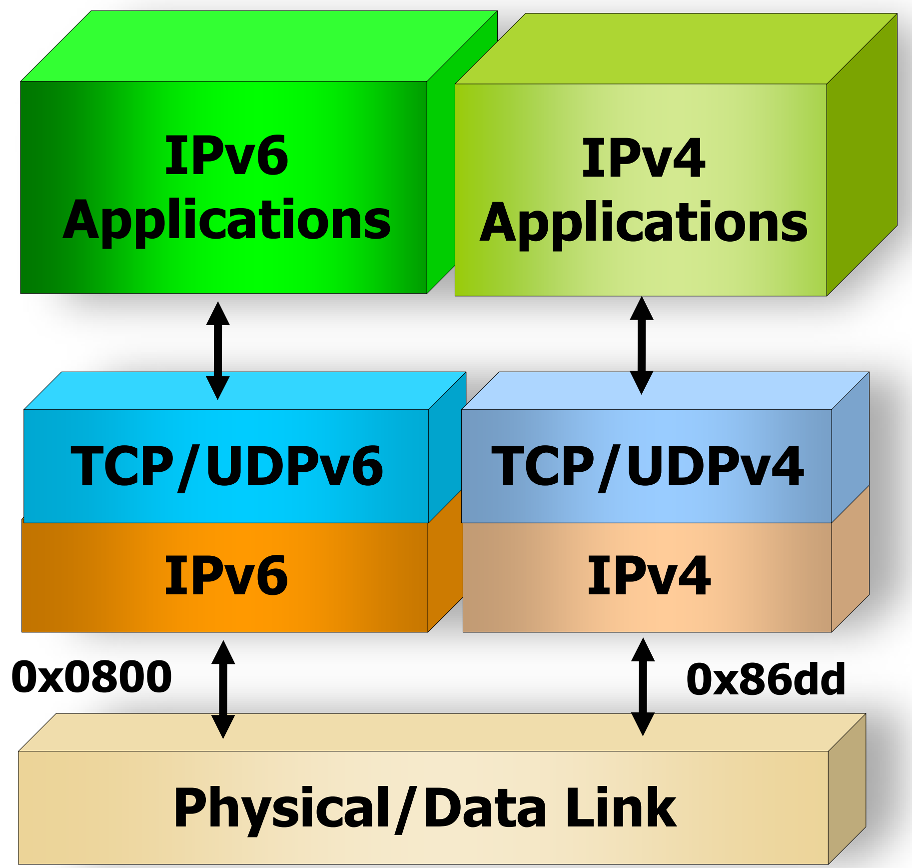
- No "Flag Day"
  - Last Internet transition was 1983 (NCP → TCP)
- Transition will be incremental
  - Possibly over several years
- No IPv4/IPv6 barriers at any time
- No transition dependencies
  - No requirement of node X before node Y
- Must be easy for end user
  - Transition from IPv4 to dual stack must not break anything
- IPv6 is designed with transition in mind
  - Assumption of IPv4/IPv6 coexistence
- Many different transition technologies are A Good Thing™
  - "Transition toolbox" to apply to myriad unique situations

Types of Transition Mechanisms

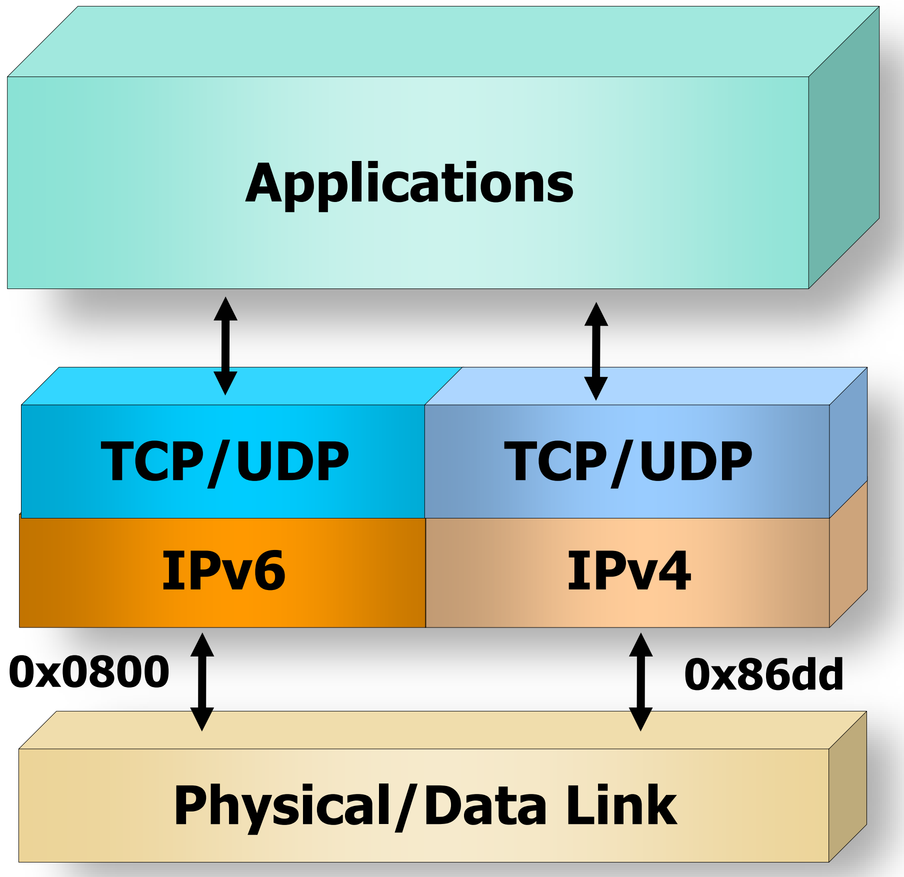
- **Dual Stacks**
  - **IPv4/IPv6 coexistence on one device**
- **Tunnels**
  - **For tunneling IPv6 across IPv4 clouds**
  - **Later, for tunneling IPv4 across IPv6 clouds**
  - **IPv6 <-> IPv6 and IPv4 <-> IPv4**
- **Translators**
  - **IPv6 <-> IPv4**

Dual Stacks

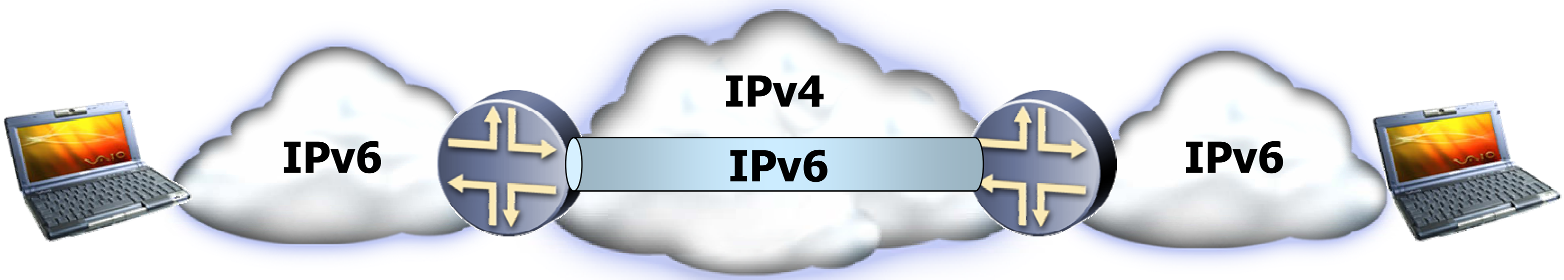
- **Network, Transport, and Application layers do not necessarily interact without further modification or translation**



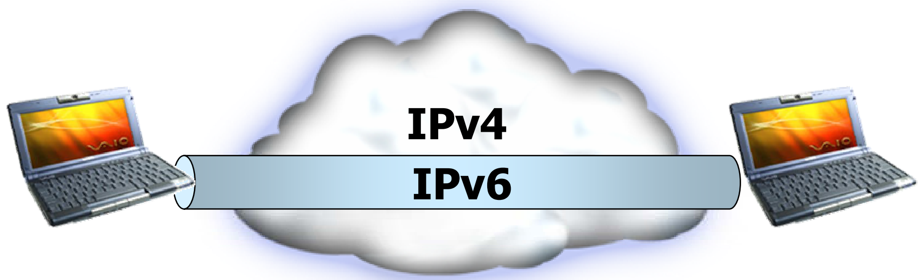
"Dual Layers"


Applications
TCP/UDP
TCP/UDP
IPv6
IPv4
0x0800
0x86dd
Physical/Data Link

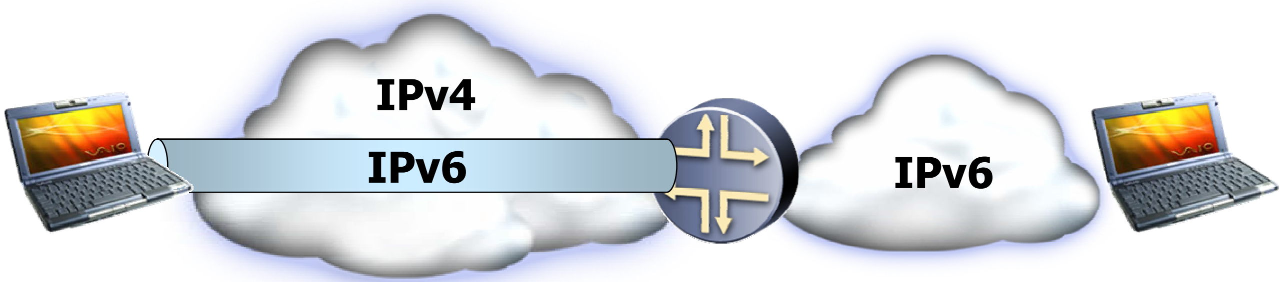
Tunnel Applications



Router to Router



Host to Host

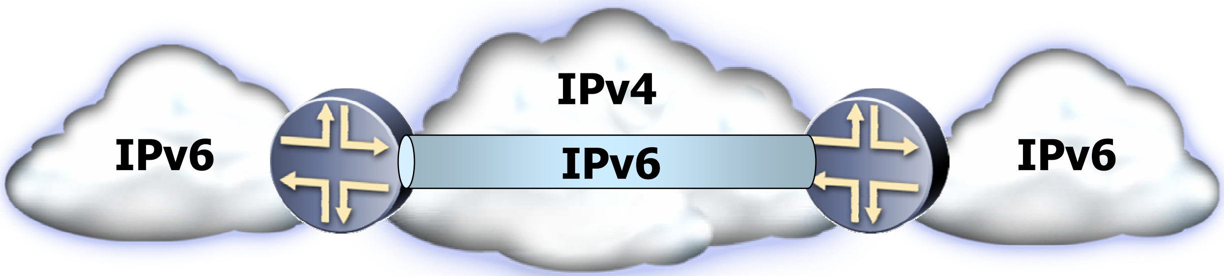


Router / Router to Host

Tunnel Types

- **Configured tunnels**
  - **Router to router**
- **Automatic tunnels**
  - **Tunnel Brokers (RFC 3053)**
    - **Server-based automatic tunneling**
  - **6to4 (RFC 3056)**
    - **Router to router**
  - **ISATAP (Intra-Site Automatic Tunnel Addressing Protocol)**
    - **Host to router, router to host**
    - **Maybe host to host**
  - **6over4 (RFC 2529)**
    - **Host to router, router to host**
  - **Teredo**
    - **For tunneling through IPv4 NAT**
  - **IPv64**
    - **For mixed IPv4/IPv6 environments**
  - **DSTM (Dual Stack Transition Mechanism)**
    - **IPv4 in IPv6 tunnels**

Configuration Example: Configured GRE Tunnel



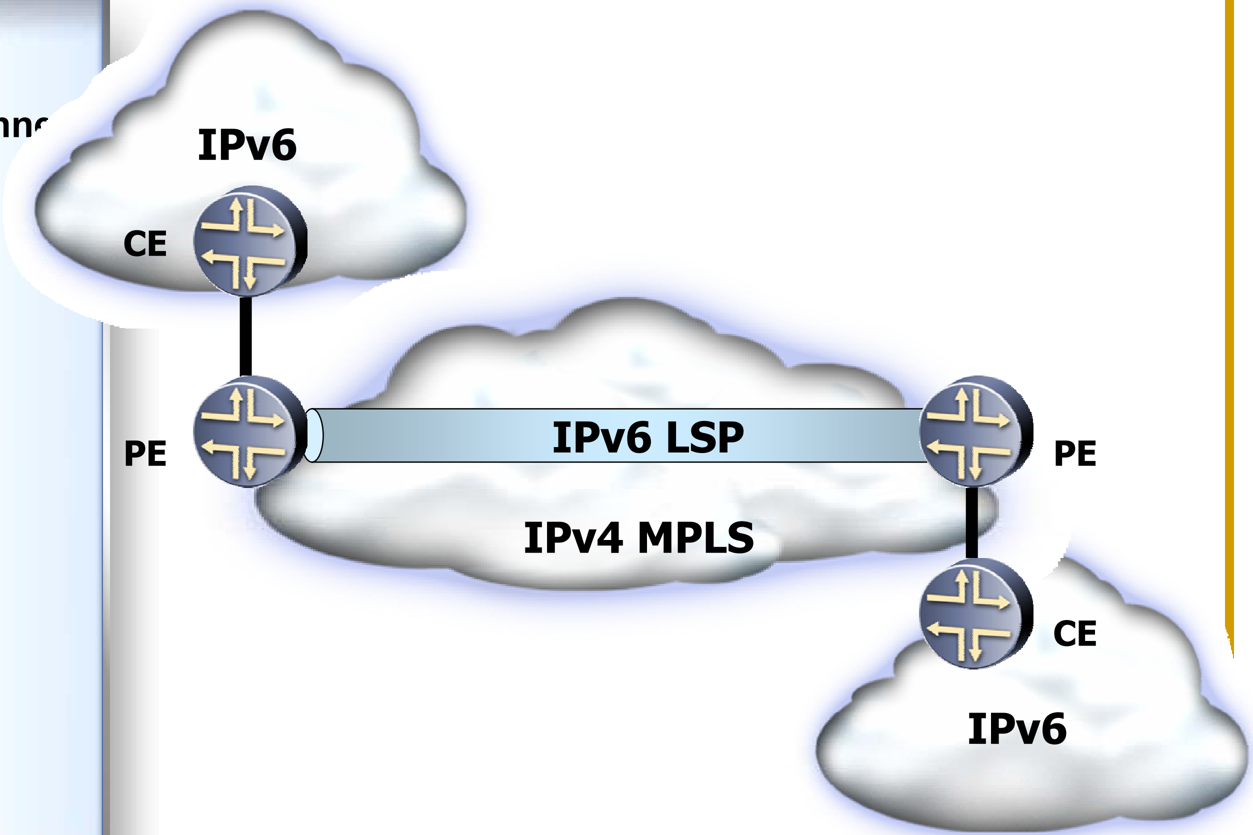

```
gr-0/0/0 {
    unit 0 {
        tunnel {
            source 172.16.1.1;
            destination 192.168.2.3;
        }
        family inet6 {
            address 2001:240:13::1/126;
        }
    }
}
```

```
gr-1/0/0 {
    unit 0 {
        tunnel {
            source 192.168.2.3;
            destination 172.16.1.1;
        }
        family inet6 {
            address 2001:240:13::2/126;
        }
    }
}
```

Configuration Example:
Configured MPLS Tunnel

PE Router:

```
mpls {
    ipv6-tunneling;
    label-switched-path v6-tunn
{
        to 192.168.2.3;
        no-cspf;
    }
}
bgp {
    group IPv6-neighbors {
        type internal;
        family inet6 {
            labeled-unicast {
                explicit-null;
            }
        }
        neighbor 192.168.2.3;
    }
}
```



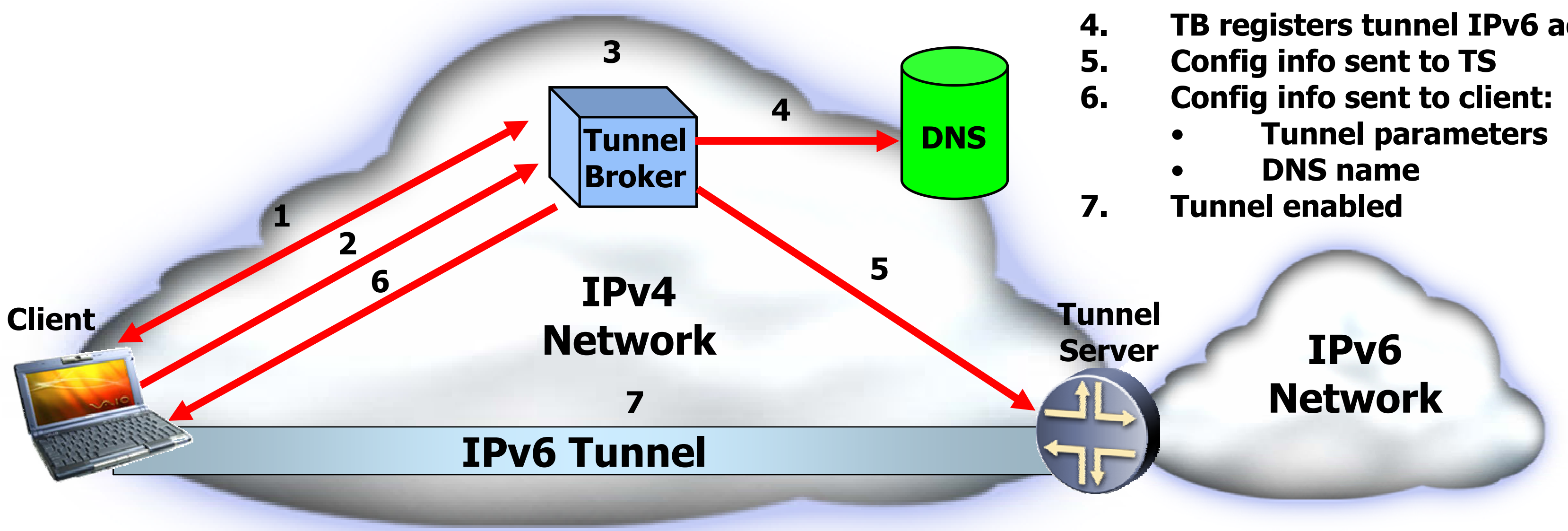
Tunnel Setup Protocol (TSP)

- **Proposed control protocol for negotiating tunnel parameters**
  - **Applicable to several IPv6 tunneling schemes**
  - **Can negotiate either IPv6 or IPv4 tunnels**
  - **Uses XML messages over TCP session**
- **Example tunnel parameters:**
  - **IP addresses**
  - **Prefix information**
  - **Tunnel endpoints**
  - **DNS delegation**
  - **Routing information**
  - **Server redirects**
- **Three TSP phases:**
  1. **Authentication Phase**
  2. **Command Phase (client to server)**
  3. **Response Phase (server to client)**

Tunnel Broker

- RFC 3053 describes general architecture, not a specific protocol
- Designed for small sites and isolated IPv6 hosts to connect to an existing IPv6 network
- Three basic components:
  - Client: Dual-stacked host or router, tunnel end-point
  - Tunnel Broker: Dedicated server for automatically managing tunnel requests from users, sends requests to Tunnel Server
  - Tunnel Server: Dual-stacked Internet-connected router, other tunnel end point
- A few tunnel brokers:
  - Freenet6 [Canada] (www.freenet6.net)
  - CERNET/Nokia [China] (www.tb.6test.edu.cn)
  - Internet Initiative Japan (www.iij.ad.jp)
  - Hurricane Electric [USA] (www.tunnelbroker.com)
  - BTexacT [UK] (www.tb.ipv6.btexact.com)
  - Many others…

Tunnel Broker



1. AAA Authorization
2. Configuration request
3. TB chooses:
   - TS
   - IPv6 addresses
   - Tunnel lifetime
4. TB registers tunnel IPv6 addresses
5. Config info sent to TS
6. Config info sent to client:
   - Tunnel parameters
   - DNS name
7. Tunnel enabled

6to4

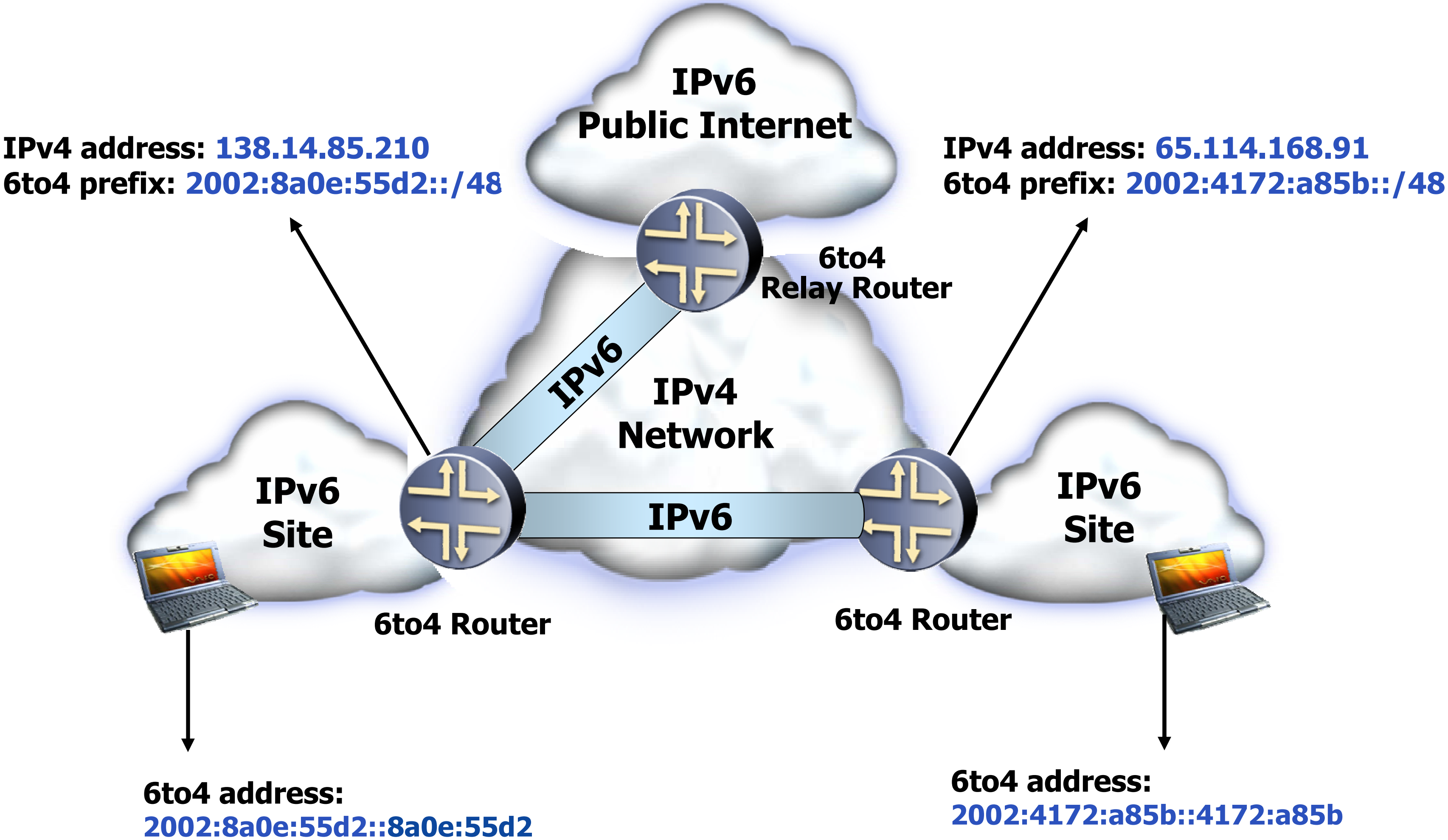
- Designed for site-to-site and site to existing IPv6 network connectivity
- Site border router must have at least one globally-unique IPv4 address
- Uses IPv4 embedded address

Example:

Reserved 6to4 TLA-ID:	2002::/16
IPv4 address:	138.14.85.210 = 8a0e:55d2
Resulting 6to4 prefix:	2002:8a0e:55d2::/48

- Router advertises 6to4 prefix to hosts via RAs
- Embedded IPv4 address allows discovery of tunnel endpoints

6to4


IPv4 address: 138.14.85.210
6to4 prefix: 2002:8a0e:55d2::/48
IPv6
Public Internet
IPv4 address: 65.114.168.91
6to4 prefix: 2002:4172:a85b::/48
6to4
Relay Router
IPv6
IPv4
Network
IPv6
Site
IPv6
IPv6
Site
6to4 Router
6to4 Router
6to4 address:
2002:8a0e:55d2::8a0e:55d2
6to4 address:
2002:4172:a85b::4172:a85b

Configuration Example: Windows XP 6to4 Interface

```
C:\Documents and Settings\Jeff Doyle>ipv6 if 3
Interface 3: 6to4 Tunneling Pseudo-Interface
  does not use Neighbor Discovery
  does not use Router Discovery
    preferred global 2002:4172:a85b::4172:a85b, life infinite
  link MTU 1280 (true link MTU 65515)
  current hop limit 128
  reachable time 23000ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
```

6to4 Prefix

= 65.114.168.91

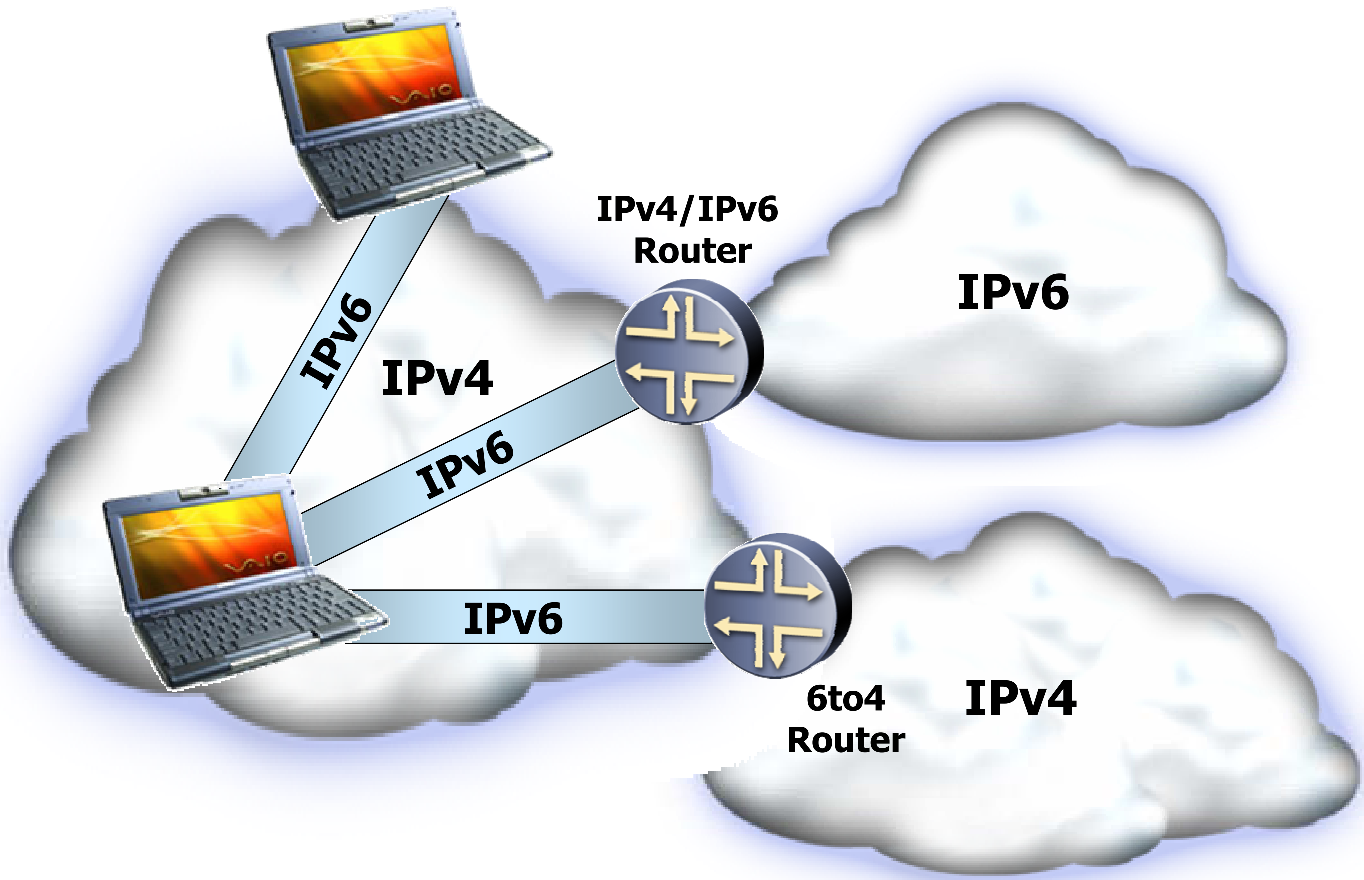
ISATAP

- **Forms 64-bit Interface ID from IPv4 address + special reserved identifier**
  - **Format: ::0:5efe:W.X.Y.Z**
  - **0:5efe = 32-bit IANA-reserved identifier**
  - **W.X.Y.Z = IPv4 address mapped to last 32 bits**

Example:

IPv4 address:	**65.114.168.91**
Global IPv6 prefix:	**2001:468:1100:1::/64**
Link-local address:	**fe80::5efe:65.114.168.91**
Global IPv6 address:	**2001:468:1100:1::5efe:65.114.168.91**

ISATAP



Configuration Example:
Windows XP ISATAP Interface

```
C:\Documents and Settings\Jeff Doyle>ipv6 if 2
Interface 2: Automatic Tunneling Pseudo-Interface
  does not use Neighbor Discovery
  does not use Router Discovery
  router link-layer address: 0.0.0.0
  EUI-64 embedded IPv4 address: 0.0.0.0
    preferred link-local fe80::5efe:169.254.113.126, life infinite
    preferred link-local fe80::5efe:65.114.168.91, life infinite
    preferred global ::65.114.168.91, life infinite
  link MTU 1280 (true link MTU 65515)
  current hop limit 128
  reachable time 24000ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
```

- **Link-Local IPv6 Address**
- **ISATAP Identifier**
- **IPv4 Address**

6over4

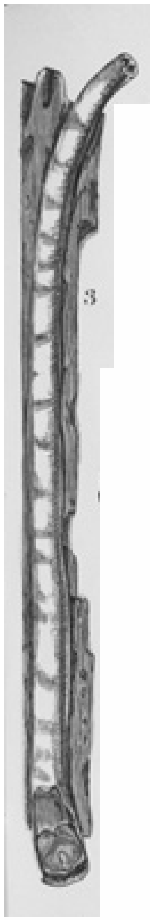
- aka "Virtual Ethernet"
- Early proposed tunnel solution
- Isolated IPv6 hosts create their own tunnels
- Encapsulates IPv6 packets in IPv4 (protocol type 41)
- Assumes IPv4 multicast domain
  - Multicast for neighbor/router discovery, autoconfiguration

Example IPv4 Multicast Address:

239.192.A.B

A, B = Last 2 Bytes of IPv6 Address

Teredo



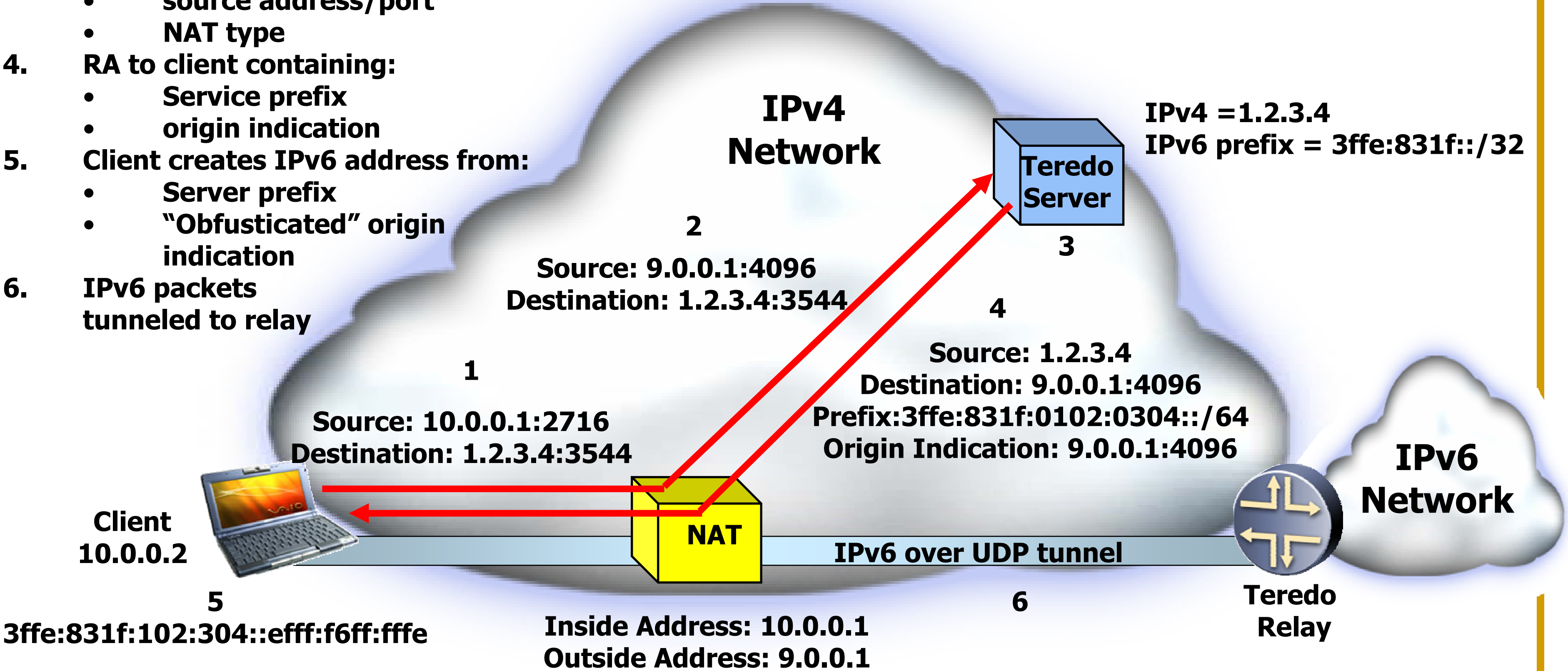
Teredo navalis

- aka “Shipworm”
- For tunneling IPv6 through one or several NATs
  - Other tunneling solutions require global IPv4 address, and so do not work from behind NAT
  - Can be stateless or stateful (using TSP)
- Tunnels over UDP (port 3544) rather than IP protocol #41
- Basic components:
  - Teredo Client: Dual-stacked node
  - Teredo Server: Node with globally routable IPv4 Internet access, provides IPv6 connectivity to client
  - Teredo Relay: Dual-stacked router providing connectivity to client
  - Teredo Bubble: IPv6 packet with no payload (NH #59) for creating mapping in NAT
  - Teredo Service Prefix: Prefix originated by TS for creating client IPv6 address

Teredo

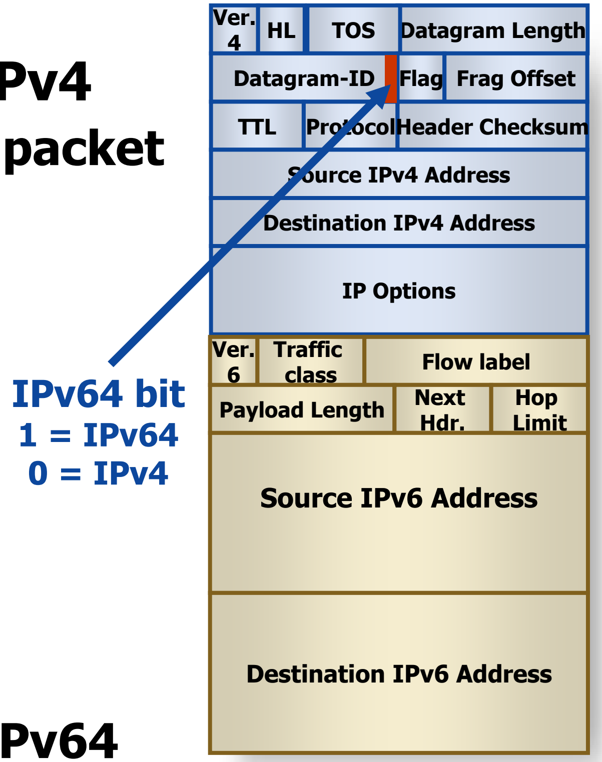
1. RS to server
2. NAT maps inside address/port to outsde address/port
3. TS notes:
   - source address/port
   - NAT type
4. RA to client containing:
   - Service prefix
   - origin indication
5. Client creates IPv6 address from:
   - Server prefix
   - “Obfusticated” origin indication
6. IPv6 packets tunneled to relay

- TSP can be used in place of RS/RA for:
  - Stateful tunnel
  - Authentication



IPv64

- Proposed for highly interconnected IPv4 and IPv6 networks (mid-transition)
- IPv64 packets: IPv6 encapsulated in IPv4
  - 48th bit of IPv4 header indicates IPv64 packet
- IPv64 routers:
  - Process IPv64 packets as IPv6
  - Process IPv4 packets as IPv4
  - Process IPv6 packets as IPv6
- IPv4 routers:
  - Process IPv64 packets as IPv4
- IPv6 routers:
  - Cannot process IPv64 packets
  - IPv64-to-IPv4 translation required at IPv64 routers
  - Proposed IPv6 Extension Header carries necessary IPv4 information for re-translating back to IPv64, if necessary

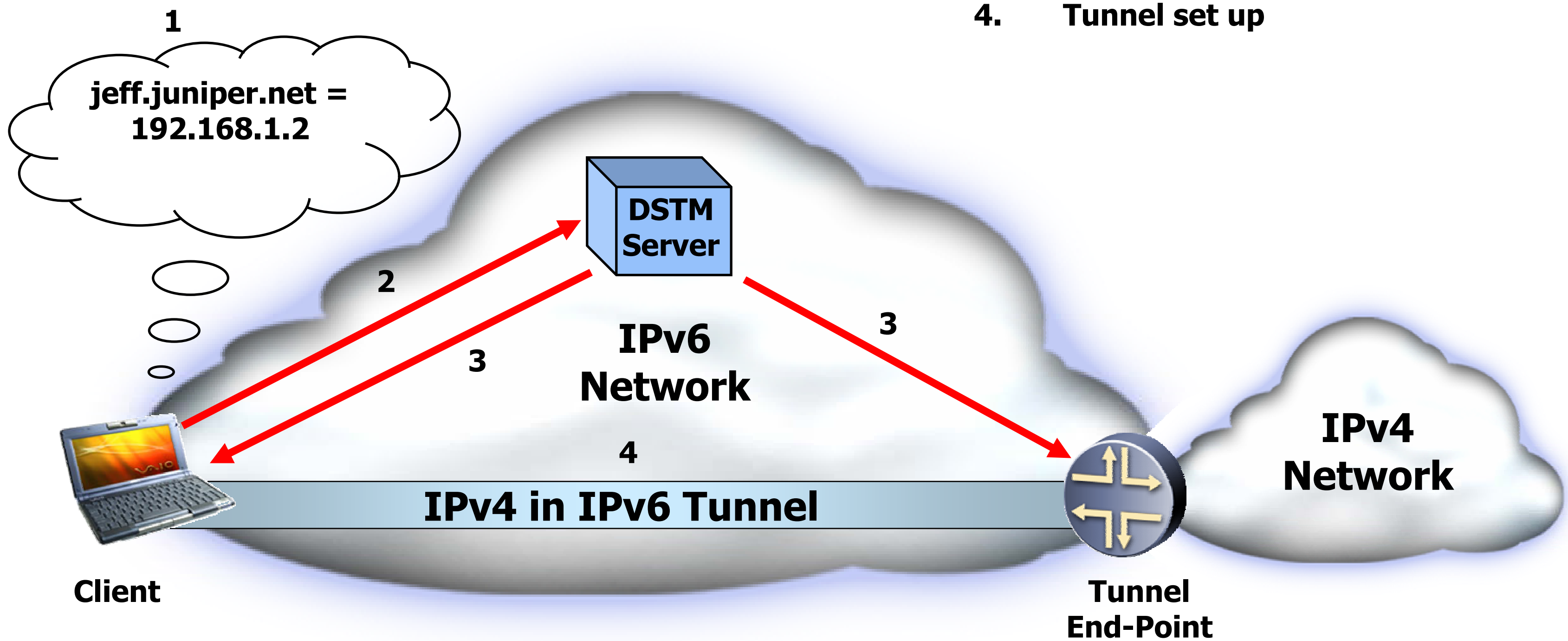


Dual-Stack Transition Mechanism (DSTM)

- aka 4over6
  - Tunnels IPv4 over IPv6 networks
  - Next-Header Number for IPv4 = 4
- Three basic components:
  - Tunnel End Point: Border router between IPv6-only network and IPv4 Internet or intranet
  - DSTM Clients: Dual-stacked nodes, create tunnels to Tunnel End Pont (TEP)
  - DSTM Address Server: Allocates IPv4 addresses to clients
- Uses existing protocols
  - DSTM Server can communicate with Client or TEP via DHCPv6 or TSP
- Server can optionally assign port range for IPv4 address conservation
  - Multiple clients have same IPv4 address, different port ranges

DSTM

1. Client needs IPv4 connectivity
2. Client requests tunnel info
3. Server sends IPv4 tunnel endpoint addresses
4. Tunnel set up



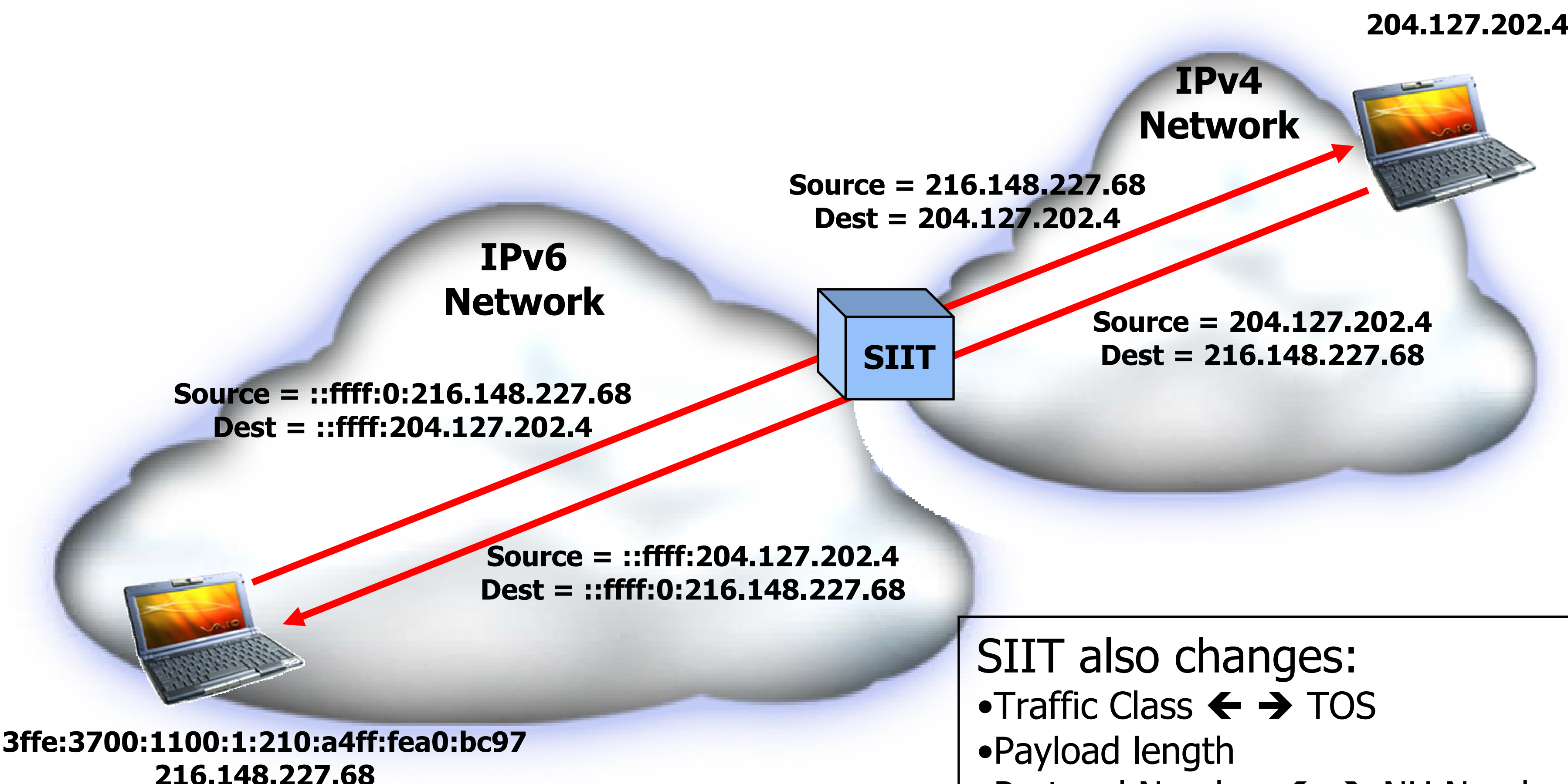
Translators

- **Network level translators**
  - Stateless IP/ICMP Translation Algorithm (SIIT)(RFC 2765)
  - NAT-PT (RFC 2766)
  - Bump in the Stack (BIS) (RFC 2767)
- **Transport level translators**
  - Transport Relay Translator (TRT) (RFC 3142)
- **Application level translators**
  - Bump in the API (BIA)(RFC 3338)
  - SOCKS64 (RFC 3089)
  - Application Level Gateways (ALG)

Stateless IP/ICMP Translation (SIIT)

- Translator replaces headers IPv4 ⬄IPv6
- Translates ICMP messages
  - Contents of message translated
  - ICMP pseudo-header checksum added
- Fragments IPv4 messages to fit IPv6 MTU when necessary
- Uses IPv4-translated addresses to refer to IPv6-enabled nodes
  - 0:0:ffff:0:0:0/96 + 32-bit IPv4 address
- Uses IPv4-mapped addresses to refer to IPv4-only nodes
  - 0:0:0:0:0:ffff/96 + 32-bit IPv4 address
- Requires IPv6 hosts to acquire an IPv4 address
  - SIIT must know these addresses

Stateless IP/ICMP Translation (SIIT)



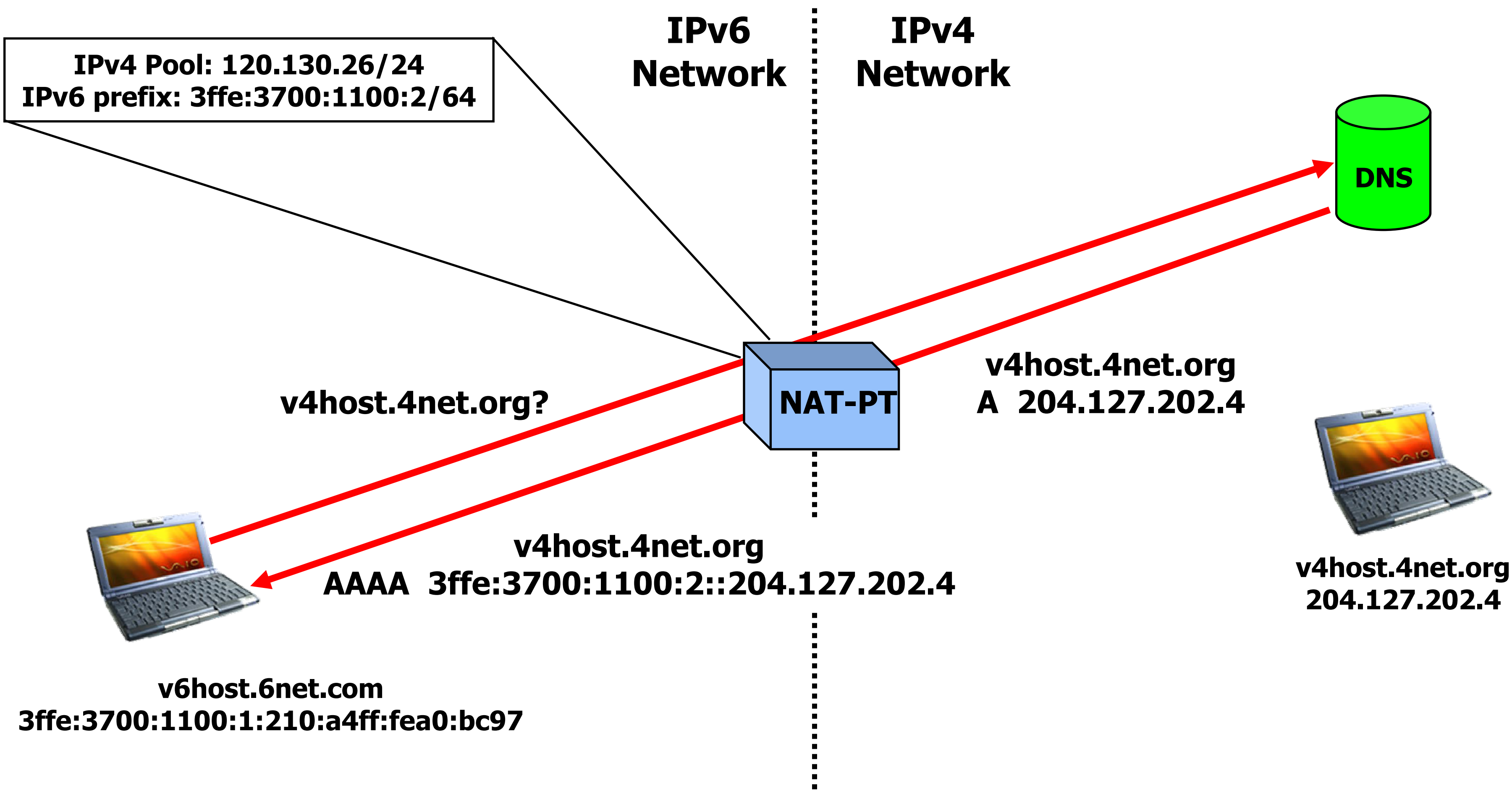
SIIT also changes:

- Traffic Class ← → TOS
- Payload length
- Protocol Number ← → NH Number
- TTL ← → Hop Limit

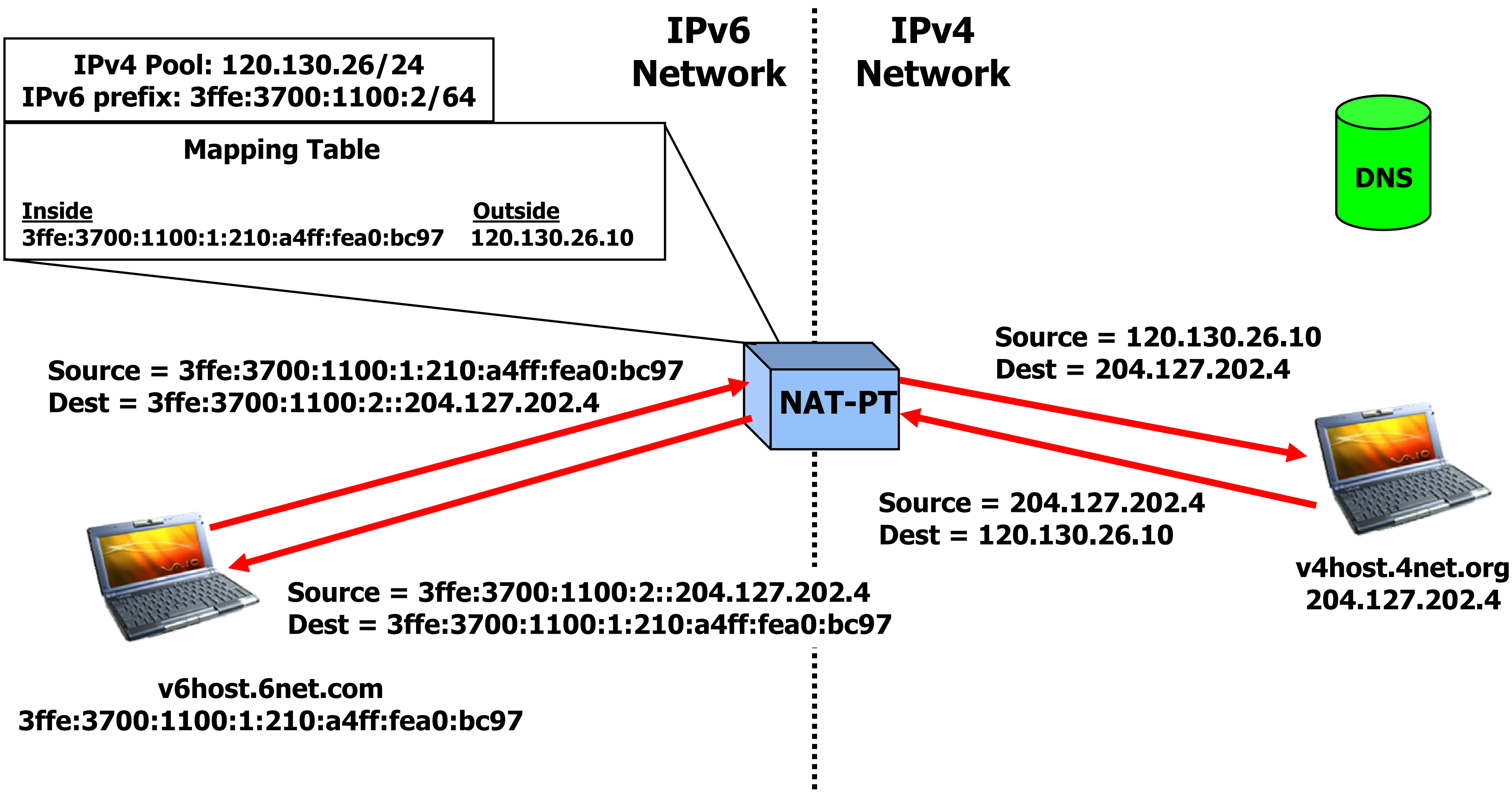
Network Address Translation - Protocol Translation (NAT-PT)

- **Stateful address translation**
  - **Tracks supported sessions**
  - **Inbound and outbound session packets must traverse the same NAT**
- **Uses SIIT for protocol translation**
- **Two variations:**
  - **Basic NAT-PT provides translation of IPv6 addresses to a pool of IPv4 addresses**
  - **NAPT-PT manipulates IPv6 port numbers so that multiple IPv6 sources can share a single IPv4 address**
- **DNS Application Level Gateway (DNS-ALG) is also specified, but has some problems**
  - **Internal A queries might return AAAA record**
  - **Possible problems for internal zone transfers, mixed v4/v6 networks, etc.**
  - **Possible problems resolving to external dual-stacked hosts**
  - **Assumes DNS traffic traverses NAT-PT box (topology limitation)**
  - **No DNS-sec**
  - **Vulnerable to DoS attacks by depletion of address pools**
  - **See:**
    - **draft-durand-natpt-dns-alg-issues-00 for more information**
    - **draft-hallin-natpt-dns-alg-solutions-01 for some proposed solutions**

Network Address Translation - Protocol Translation (NAT-PT)

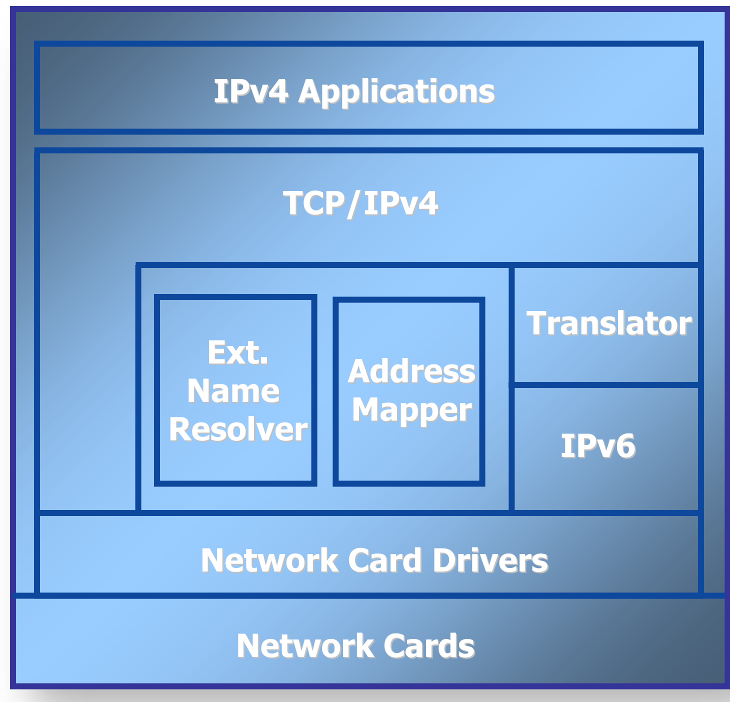


Network Address Translation - Protocol Translation (NAT-PT)



Bump in the Stack (BIS)

- Translator resides in host
- Allows IPv4 applications to run on IPv6 host
- Three components:
  - Translator
    - IPv4 ← → IPv6
    - Uses SIIT
  - Address mapper
    - Maintains IPv4 address pool
    - Maps IPv6 addresses to IPv4 addresses
  - Extension Name Resolver
    - Manages DNS queries
    - Converts AAAA records to A records
    - Similar to NAT-PT DNS ALG



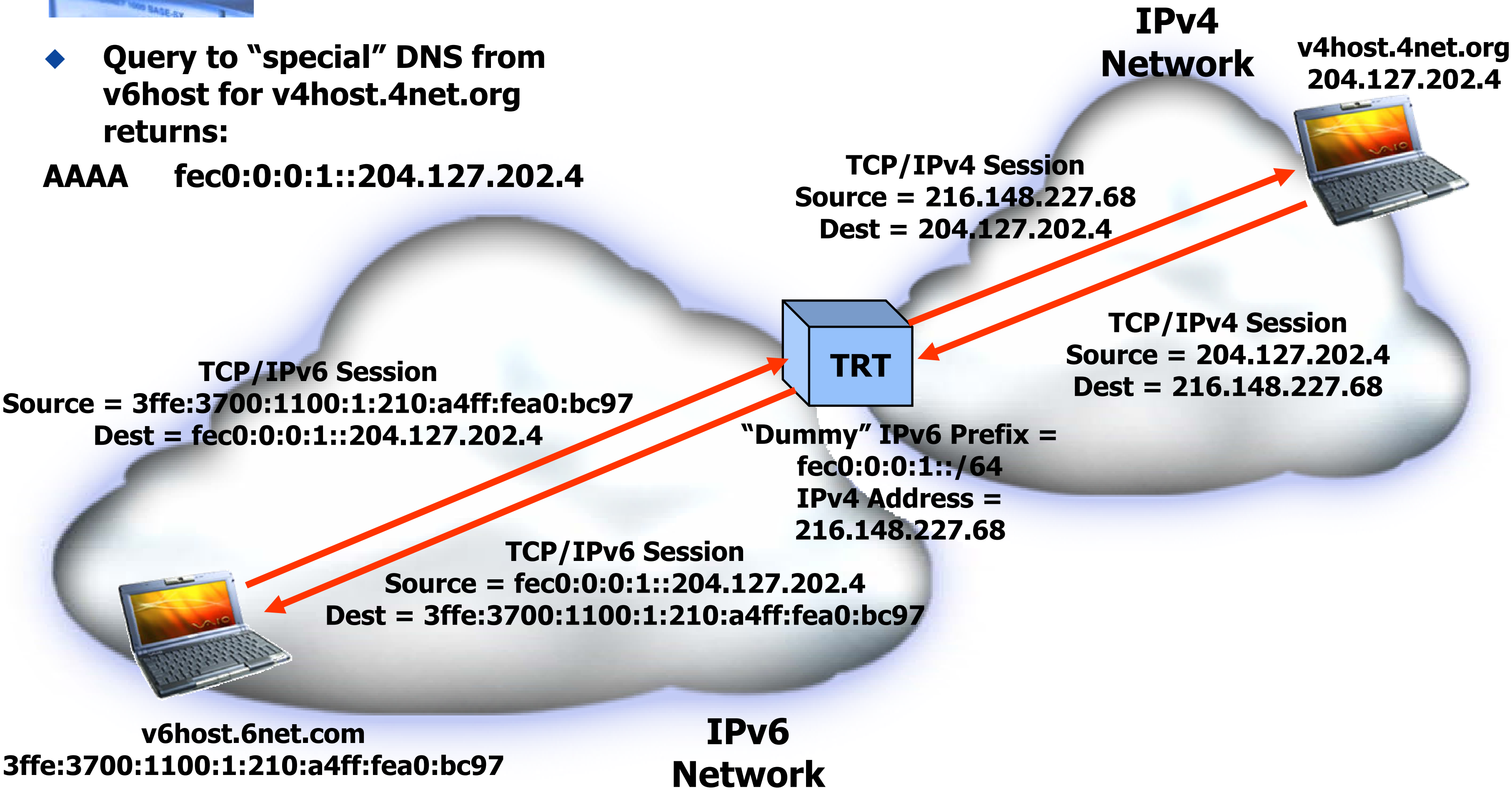
Transport Relay Translator (TRT)

- aka TCP/UDP Relay
- Based on proxy firewall concept
- No IP packets transit the TRT
- Two connections established:
  - Initiator to TRT
  - TRT to target node
- Requires "special" DNS to translate IPv4 addresses into IPv6 and vice versa
  - TRT does not translate DNS queries/records
- Only works with TCP and UDP

Transport Relay Translator (TRT)

- **Query to "special" DNS from v6host for v4host.4net.org returns:**

AAAA fec0:0:0:1::204.127.202.4

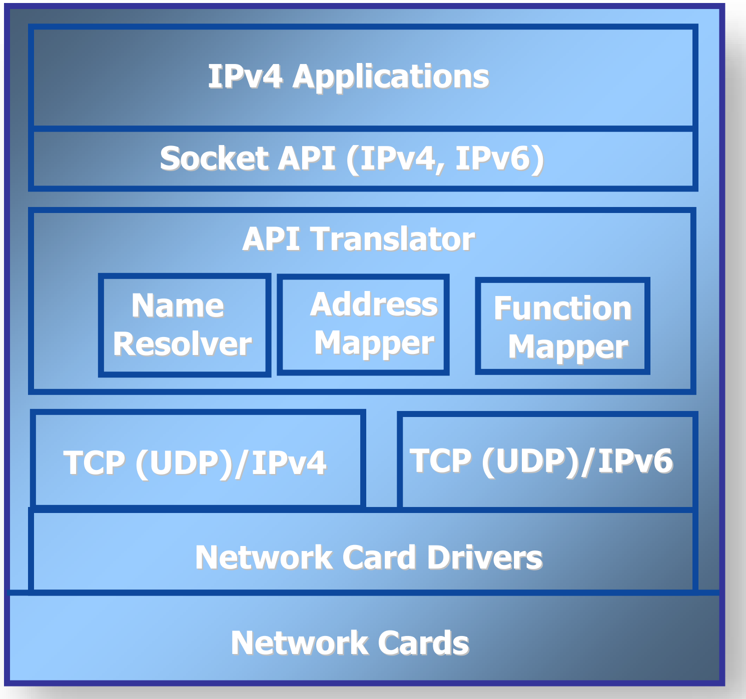


Bump in the API (BIA)

- Allows dual-stacked IPv6 hosts to use IPv4 applications
  - Same goal as BIS, but translation is between IPv4 and IPv6 APIs
  - API Translator resides between socket API module and IPv4/IPv6 TCP/IP modules
  - No header translation required
  - Uses SIIT for conversion mechanism

Bump in the API (BIA)

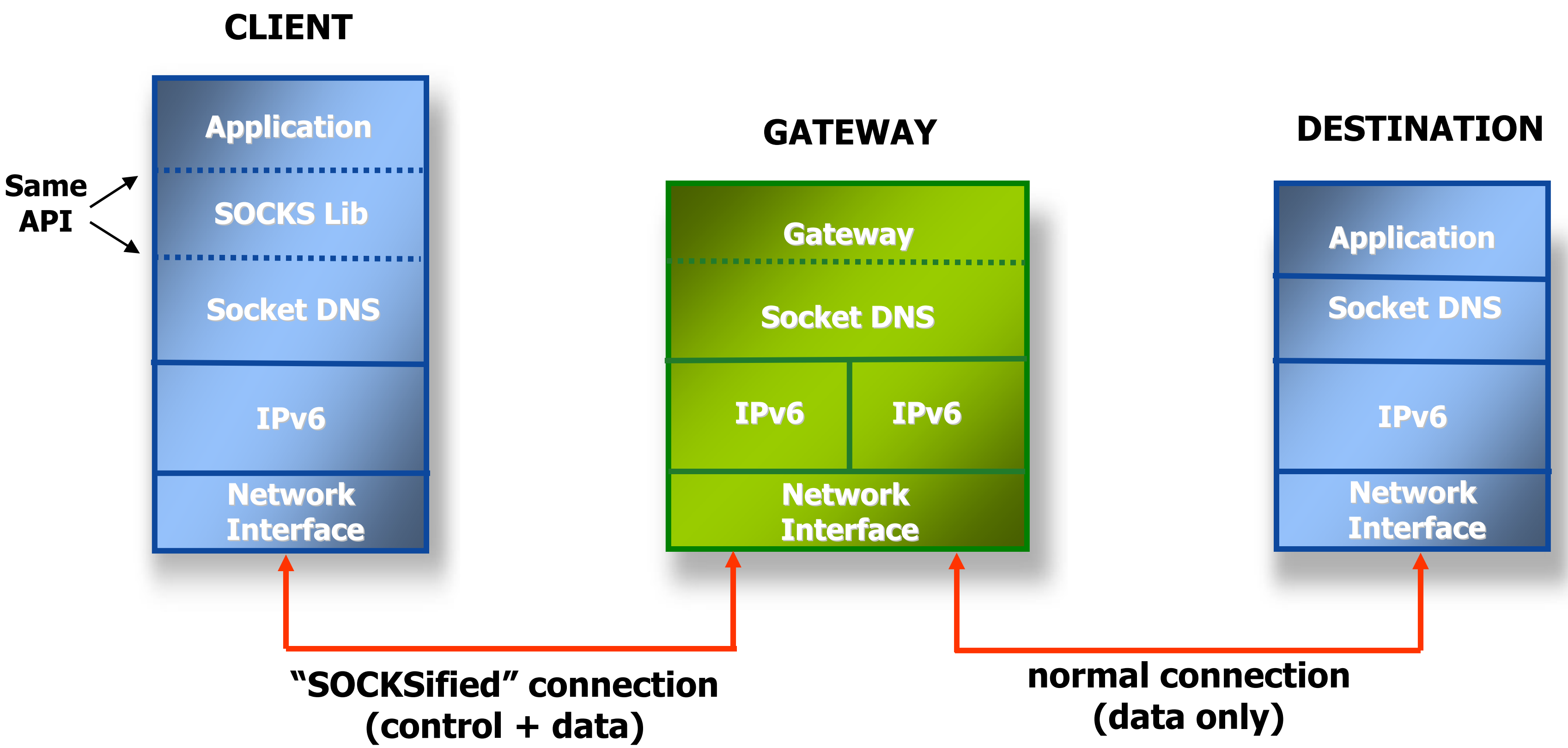
- **API Translator consists of three modules:**
  - **Name Resolver intercepts IPv4 DNS calls, uses IPv6 calls instead**
  - **Address Mapper maintains mappings of internal pool unassigned of IPv4 addresses (0.0.0.1 ~ 0.0.0.255) to IPv6 addresses**
  - **Function Mapper translates IPV4 socket API functions to IPv6 socket API functions and vice versa**



SOCKS64

- Uses existing SOCKSv5 protocol
  - RFC 1928
  - Designed for firewall systems
- Two basic components:
  - Gateway
    - SOCKS server
    - IPv4 and IPv6 connections terminate at gateway
    - Gateway relays connections at application layer
  - SOCKS Lib
    - Installs on client between application layer and socket layer
    - Can replace:
      - Applications' socket APIs
      - DNS name resolving APIs
    - Maintains mapping table between "fake" IPv4 addresses (0.0.0.1 ~ 0.0.0.255) and logical host names (FQDNs)

SOCKS64


CLIENT
Same API
Application
SOCKS Lib
Socket DNS
IPv6
Network Interface
GATEWAY
Gateway
Socket DNS
IPv6
IPv6
Network Interface
DESTINATION
Application
Socket DNS
IPv6
Network Interface
"SOCKSified" connection (control + data)
normal connection (data only)

Application Layer Gateways

- Application-specific translator
- Needed when application layer contains IP address
- Similar to ALGs used in firewalls, some NATs

Agenda

- Drivers for IPv6 Deployment
- Routing IPv6
- Multihoming IPv6
- Transition Mechanisms
- **Transition Issues**

Transition Issues:
DNS

- Namespace fragmentation
  - Some names on IPv4 DNS, others on IPv6 DNS
  - How does an IPv4-only host resolve a name in the IPv6 namespace, and vice versa?
  - How does a dual-stack host know which server to query?
  - How do root servers share records?
- MX records
  - How does an IPv4 user send mail to an IPv6 user and vice versa?
- Solutions:
  - Dual stacked resolvers
  - Every zone must be served by at least one IPv4 DNS server
  - Use translators
    - NAT-PT does not work for this
    - totd: proxy DNS translator
- Some DNS transition issues discussed in RFC 1933, Section 3.2

DNS AAAA Records

- RFC 1886
- BIND 4.9.4 and up; BIND 8 is recommended
- Simple extension of A records
  - Resource Record type = 28
  - Query types performing additional section processing (NS, MX, MB) redefined to perform both A and AAAA additional section processing
- ip6.int, ipv6.arpa analogous to in-addr.arpa for reverse mapping
  - IPv6 address represented in reverse, dotted hex nibbles

AAAA record:

homer	IN	AAAA	2001:4210:3:ce7:8:0:abcd:1234

PTR record:

4.3.2.1.d.c.b.a.0.0.0.0.8.0.0.0.7.e.c.0.3.0.0.0.0.1.2.4.1.0.0.2.ip6.int.	IN	PTR	homer.simpson.net

- RFC 3152 deprecates ip6.int in favor of ip6.arpa

DNS A6 Records

- Proposed alternative to AAAA records
  - RFC 2874
  - Resource Record type = 38
- A6 RR can contain:
  - Complete IPv6 address, or
  - Portion of address and information leading to one or more prefixes
- Supported in BIND 9
- More complicated records , but easier renumbering
  - Segments of IPv6 address specified in chain of records
  - Only relevant records must be changed when renumbering
  - Separate records can reflect addressing topology

A6 Record Chain

Queried Name: homer.simpson.net

$ORIGIN simpson.net					
homer	IN	A6	64	::8:0:abcd.1234	sla5.subnets.simpson.net.
$ORIGIN subnets.simpson.net					
sla5	IN	A6	48	0:0:0:ce7::	site3.sites.net.
$ORIGIN sites.net					
site3	IN	A6	32	0:0:3::	area10.areas.net.
$ORIGIN areas.net					
area10	IN	A6	24	0:10::	tla1.tlas.net.
$ORIGIN tlas.net					
tla1	IN	A6	0	2001:4200::	

Returned Address: 2001:4210:3:ce7:8:0:abcd:1234

Bitstring Labels

- New scheme for reverse lookups
- Bitstring Labels: RFC 2874
- Bitstring Labels for IPv6: RFC 2673

Examples:

Address: `2001:4210:3:ce7:8:0:abcd:1234`

Bitstring labels:

`\[x200142100003 0ce700080000abcd1234/128].ip6.arpa.`

`\[x00080000abcd1234/64].\[x0ce7/16].\[x20014210/48].ip6.arpa.`

- Pro:
  - More compact than textual (ip6.int) representation
- Con:
  - All resolvers and authoritative servers must be upgraded before new label type can be used
- RFC 3152 deprecates ip6.int in favor of ip6.arpa

DNAME

- DNAME: RFC 2672
- DNAME for IPv6: RFC 2874
- Provides alternate naming to an entire subtree of domain name space
  - Rather than to a single node
- Chaining complementary to A6 records
- DNAME not much more complex than CNAME
- DNAME changed from Proposed Standard to Experimental status in RFC 3363

DNAME Reverse Lookup

Queried Address: 2001:4210:3:ce7:8:0:abcd:1234

$ORIGIN ip6.arpa. \[x200142/24]	IN	DNAME	ip6.tla.net
$ORIGIN ip6.tla.net \[x10/8]	IN	DNAME	ip6.isp1.net
$ORIGIN ip6.isp1.net \[x0003/16]	IN	DNAME	ip6.isp2.net
$ORIGIN ip6.isp2.net \[x0ce7/16]	IN	DNAME	ip6.simpson.net
$ORIGIN ip6.simpson.net \[x00080000abcd1234/64]	IN	PTR	homer.simpson.net

Returned Name: homer.simpson.net

AAAA or A6?

- Good discussion of tradeoffs in RFC 3364
- AAAA Pros:
  - Essentially identical to A RRs, which are backed by extensive experience
  - "Optimized for read"
- AAAA Cons:
  - Difficult to inject new data
- A6 Pros:
  - "Optimized for write"
  - Possibly superior for rapid renumbering, some multihoming approaches (GSE-like routing)
- A6 Cons:
  - Long chains can reduce performance
  - Very little operational experience
- A6 RRs changed from Proposed Standard to Experimental status in RFC 3363
  - AAAA preferred for production deployment

Transition Issues: Security

- **Many transition technologies open security risks such as DoS attacks**
- **Examples:**
  - **Abuse of IPv4 compatible addresses**
  - **Abuse of 6to4 addresses**
  - **Abuse of IPv4 mapped addresses**
  - **Attacks by combining different address formats**
  - **Attacks that deplete NAT-PT address pools**

Transition Planning

- Assumption: Existing IPv4 network
- Easy Does It
  - Deploy IPv6 incrementally, carefully
- Have a master plan
- Think IPv4/IPv6 interoperability, not migration
- Evaluate hardware support
- Evaluate application porting
- Monitor IETF v6ops WG
  - ngtrans wg has been closed

Transition Strategies

- **Edge-to-core**
  - **The edge is the killer app!**
  - **When services are important**
  - **When addresses are scarce**
  - **User (customer) driven**
- **Core-to-edge**
  - **Good ISP strategy**
- **By routing protocol area**
  - **When areas are small enough**
- **By subnet**
  - **Probably <u>too</u> incremental**

Transition Lessons from the Past

- **KEEP TRANSITION SIMPLE**
- **Limit scope and interaction of mechanisms**
- **Beware of semantic interdependence**
- **Make sure normal humans can fully understand the interactions and implications of all mechanisms**
- **Transition/Migration is <u>THE</u> hard part**
  - **Ensuring existing products do IPv6 well**
  - **Keeping transition mechanisms under control**

Thank You!

http://www.juniper.net
jeff@juniper.net

