IPv6 (IP version 6) Essentials
Ch10: Interoperability

Louis Chuang
Fu Jen Catholic University
EE ENCL

Introductoion

- IPv6 and IPv4 will coexist for many years. There are three main transition categories:
  - Dual-stack techniques:
    - They allow IPv4 and IPv6 to coexist in the same devices and networks.
  - Tunneling techniques:
    - They allow the transport of IPv6 traffic over the existing IPv4 infrastructure.
  - Translation techniques:
    - They allow IPv6-only nodes to communicate with IPv4-only nodes.
- RFC 2893 "Transition Mechanisms for IPv6 Hosts and Routers“.
- RFC 4213 “Basic Transition Mechanisms for IPv6 Hosts and Routers”.

Dual-stack Techniques (1)

- A dual-stack node has complete support for both protocol versions:
  - This type of node is often referred to as an IPv6/IPv4 node.
- In communication with an IPv6 node: it behaves like an IPv6-only node.
- In communication with an IPv4 node: it behaves like an IPv4-only node.
- An IPv6/IPv4 node has at least one address for each protocol version.
  - It uses IPv4 mechanisms: to be configured for an IPv4 address (static configuration or DHCP).
  - It uses IPv6 mechanisms: to be configured for an IPv6 address (static configuration or autoconfiguration).
- DNS is used with both protocol versions to resolve names and IP addresses.
  - An IPv6/IPv4 node needs a DNS resolver, which is capable of resolving both types of DNS address records.
    - The DNS A record: is used to resolve IPv4 addresses.
    - The DNS AAAA or A6 record: is used to resolve IPv6 addresses.

Dual-stack Techniques (1)

- General, DNS returns only an IPv4 or an IPv6 address.
- However, if it is a dual-stack host, DNS might return both types of addresses.
- Note that the DNS resolver may run over an IPv4 or IPv6 network, but the worldwide DNS tree is mainly reachable through an IPv4 network layer.
- The disadvantage of this dual-stack technique is:
  - We must perform a full network software upgrade to run the two separate protocol stacks.
  - That is, all tables (e.g., routing tables) are kept simultaneously, routing protocols being configured for both protocols. It takes more memory and CPU power.
  - Ex) ping.exe for IPv4, and ping6.exe for IPv6.

Tunneling Techniques (1)

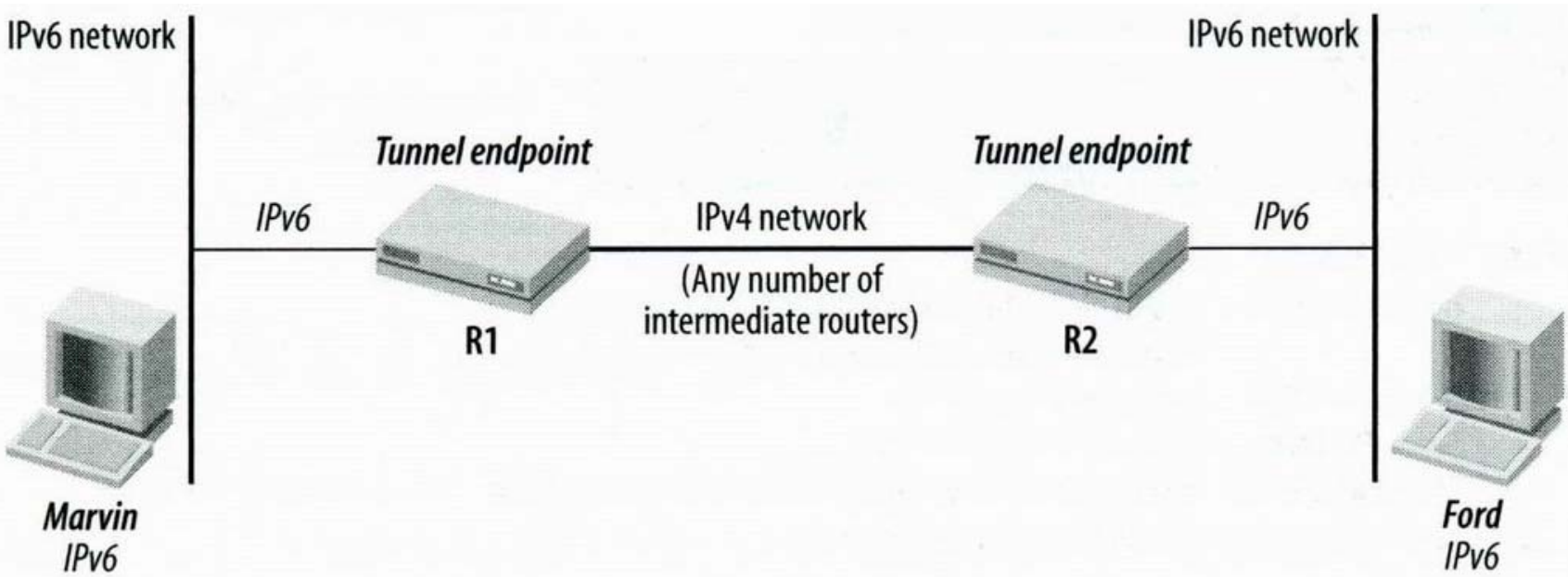
- In the IPv4 infrastructure, tunneling mechanisms can be used for IPv6 communication.
- Tunneling is also called encapsulation.
- The process of encapsulation has 3 components:
  - Encapsulation at the tunnel entry point.
  - Decapsulation at the tunnel exit point.
  - Tunnel management.
- The encapsulation of IPv6 packets in IPv4 packets are defined in several RFCs, such as RFC 2473, 2893, 3056, and 4213.
- There are two types of tunneling:
  - Manually configured tunneling of IPv6 over IPv4:
    - IPv6 packets are encapsulated in IPv4 packets to be carried over IPv4 routing infrastructures. These are point-to-point tunnels that need to be configured manually (configured tunnel).

Tunneling Techniques (2)

- Automatic tunneling of IPv6 over IPv4:
  - IPv6 nodes can use different types of addresses:
    - Such as IPv4-compatible IPv6 addresses or 6to4, ISATAP, or Teredo addresses, to dynamically tunnel IPv6 packets over an IPv4 routing infrastructure. These special IPv6 unicast addresses carry an IPv4 address in some of the IPv6 address fields.
- How tunneling works:
  - A tunnel has two endpoints:
    - One is the tunnel entry point.
    - The other is the tunnel exit point.
  - The tunnel can be implemented in different ways:
    - Router-to-router.
    - Host-to-router.
    - Host-to-host.
    - Router-to-host.

Tunneling Techniques (3)

- Encapsulation and tunneling.



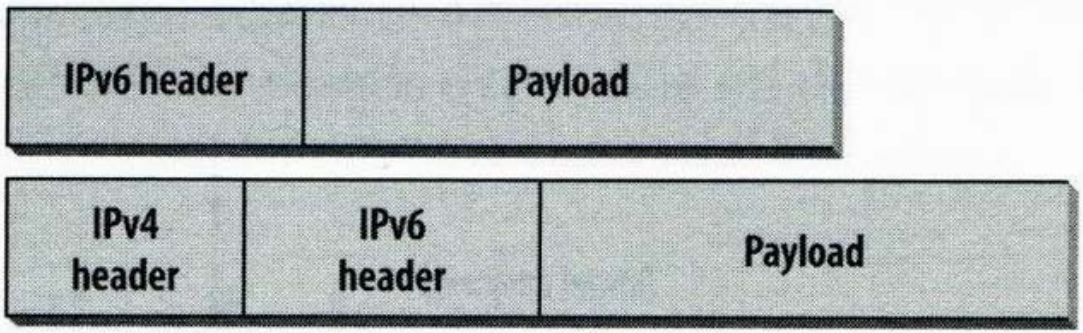
1. Marvin sends an IPv6 packet to Router 1.
2. Router 1—in this case, the tunnel entry point—encapsulates the IPv6 packet in an IPv4 header and sends it to Router 2.
3. Router 2—in this case, the tunnel exit point—strips off the IPv4 header and forwards the packet to Ford.

Tunneling Techniques (4)

- The steps for encapsulation of the IPv6 packet are the following:
  - Step 1: The entry point of the tunnel decrements the IPv6 hop limit by one.
  - Step 2: Encapsulates the packet in an IPv4 header, and transmits the encapsulated packet through the tunnel. If necessary, the IPv4 packet is fragmented.
  - Step 3: The exit point of the tunnel receives the encapsulated packet. If the packet was fragmented, the exit point reassembles it.
  - Step 4: Then the exit point removes the IPv4 header and forwards the IPv6 packet to its original IPv6 destination.
- When the tunnel exit point receives an IPv4 datagram with a protocol value of 41, it knows that this packet has been encapsulated.

Tunneling Techniques (5)

- Encapsulation.



Original IPv6 packet sent from source host to tunnel entry point.

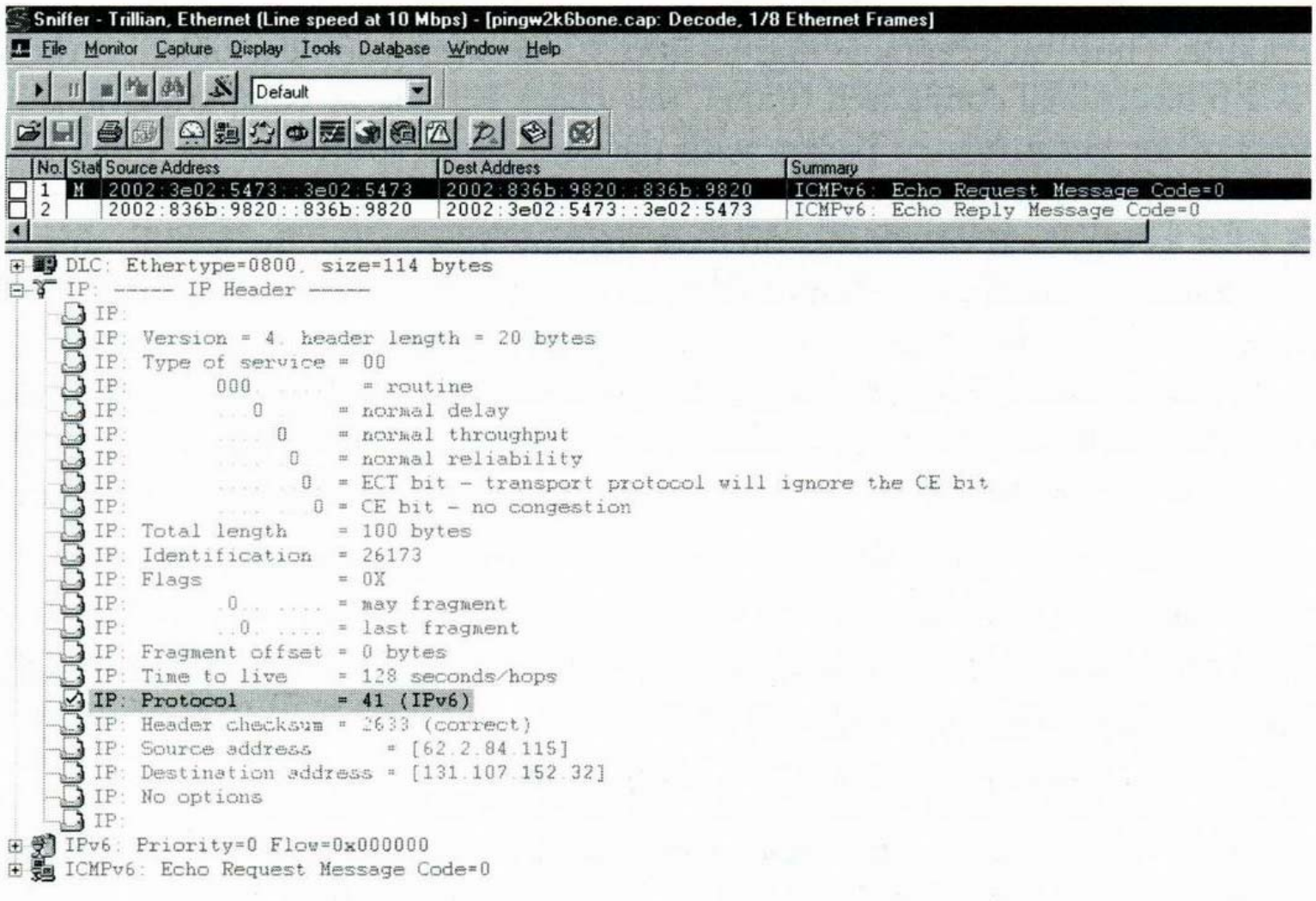
Encapsulated packet sent to tunnel exit point.

Fields in IPv4 Header:

Header Length:	Length of IPv4 header plus IPv6 header plus any extension headers and IPv6 payload. ← (packet total length)
Time to Live (TTL):	Implementation-specific.
Protocol:	Value 41 (assigned for IPv6).
Source Address:	IPv4 address of outgoing interface of tunnel entry point.
Destination Address:	IPv4 address of tunnel exit point.

Tunneling Techniques (6)

- Encapsulation in the trace file.



Tunneling Techniques (7)

- Before the tunnel exit point delivering the IPv6 packet to the final IPv6 destination, it checks to see if the IPv6 source address is valid.
- Following source addresses are invalid:
  - All multicast address (FF00::/8).
  - The loopback address (::1).
  - All IPv4-compatible IPv6 addresses (::/96).
    - Excluding the unspecified address for DAD (::/128).
  - All IPv4-mapped IPv6 addresses (::ffff:0:0/96).
- If an IPv4 router in the tunnel generates an ICMPv4 error message, this router sends the message back to the tunnel entry point (because that host is the source of that packet).
  - If the ICMPv4 packet contains enough information about the originally encapsulated IPv6 packet, the tunnel entry point may send an ICMPv6 error message back to the original IPv6 source node.

Tunneling Techniques (8)

- Both tunnel end points form their link-local IPv6 address:
  - Using their IPv4 addresses to be the interface identifiers (interface IDs) for their link-local IPv6 addresses.
  - Ex) A end point has an IPv4 address of 192.168.0.2, then its link-local IPv6 address is fe80::192.168.0.2/64.
- Automatic tunneling (RFC 2893):
  - Automatic tunneling allows IPv6/IPv4 nodes to communicate over an IPv4 infrastructure without the need for tunnel destination pre-configuration.
  - Ex) The tunnel endpoint address is the IPv4-compatible destination address.
    - The IPv4 address is 62.2.84.115. The IPv4-compatible address is :: 62.2.84.115.
    - If the IPv4 address is a global address (not private address), the IPv4-compatible address is globally unique.

Tunneling Techniques (9)

- The IPv4-compatible IPv6 address.

```
Interface 2 (site 0): Tunnel Pseudo-Interface
  does not use Neighbor Discovery
  link-level address: 0.0.0.0
    preferred address 2002:3e02:5473::3e02:5473, infinite/infinite
    preferred address ::62.2.84.115, infinite/infinite
  link MTU 1280 (true link MTU 65515)
  current hop limit 128
```

- Configured tunneling (RFC 2893):
  - RFC 4213 discusses the configuration.
  - Configured tunneling is IPv6-over-IPv4 (6over4) tunneling.
    - All tunnels are assumed to be bidirectional.
  - The IPv6 address of an IPv6/IPv4 router at the other end of the tunnel is added into the routing table as a default route.
  - All packets for all IPv6 destinations will first be sent to the route.
    - Packets are all first tunneled to the configured / default router.

Tunneling Techniques (10)

- The mask of this default route is zero.
  - That is, all different IPv6 destination addresses are first tunneled to the default IPv6 address → the configured IPv6/IPv4 router.
- Combination of automatic and configured tunneling:
  - Ex) Such a host has been configured with two routing entries for tunneling:
    - One entry point is the router ‘A’ with all-zeros 96-bit prefix (for IPv4-compatible IPv6 addresses).
      - All packets with IPv4-compatible IPv6 destination addresses will be sent via this route ‘A’.
    - The other entry point is an IPv6 router ‘B’, which is configured to perform general automatic tunneling.
      - All packets with native IPv6 destination addresses will be sent via the router ‘B’.

Tunneling Techniques (11)

- If a host sending a packet has an IPv4-compatible IPv6 address, then it will use the IPv4-compatible address as a source address for the packet and send it to the IPv4-compatible IPv6 destinations.
- If a host sending a packet has a global native IPv6 address, then it will use the native IPv6 address as a source address for the packet and send it to the native IPv6 destinations.

- Marvin's routing table:

```
C:\>ipv6 rt
::/0 -> 2/::131.107.152.32 pref 0 (lifetime 1800s, publish, no aging)
2002::/16 -> 2 pref 0 (lifetime 1800s, publish, no aging)
::/96 -> 2 pref 0 (lifetime infinite)

C:\>_
```

- The command for displaying the routing table: ipv6 rt.
- The first entry is ::/0.

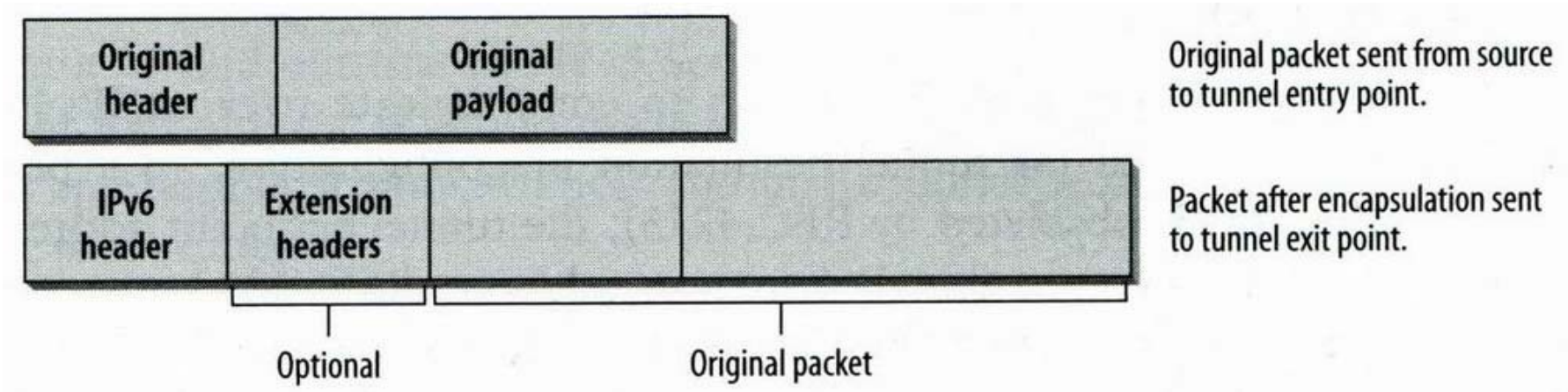
Tunneling Techniques (11)

- It is the zero-length prefix, which means it is the default route. It is on the interface 2, and its next hop IPv6 address is ::131.107.152.32 (an IPv4-compatible IPv6 address router).
- Any IPv6 traffic that does not match a more specific routing entry is encapsulated in an IPv4 header and routed through this default route.
- All traffic going out through interface 2 is encapsulated in an IPv4 header.

- The second entry is 2002::/16, it is for 6to4 router.
  - It is also going out via interface 2 and being encapsulated.
- Using these two routes, this host can reach the 6Bone and 6to4 hosts.
- The third routing entry is ::/96, it means the route is for the all-zeros prefix with a 96-bit mask (the IPv4-compatible IPv6 address).
  - IPv6 addresses matched this prefix are also going out via interface 2.

Tunneling Techniques (12)

- Encapsulation with IPv6 (RFC 2473):
  - In IPv6 network, the (IPv4) packets need to be encapsulated in an IPv6 header before transmission.
  - The tunnel entry point adds the IPv6 header.
  - If needed, one or a set of Extension headers in front of the original packet header.
  - The figure shows the tunnel IPv6 headers from the packet view.



Tunneling Techniques (13)

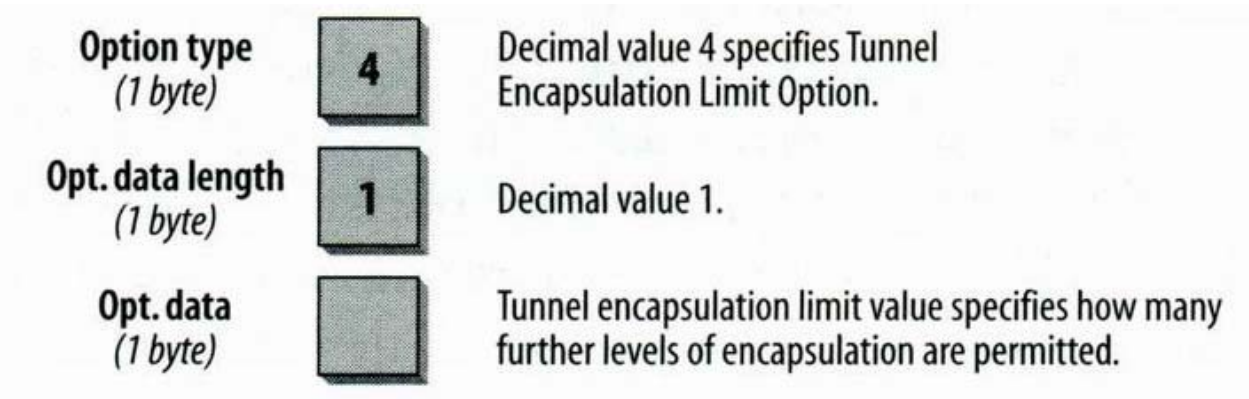
- In the IPv6 tunnel entry point:
  - The source address: is the address of the tunnel entry point node.
  - The destination address: is the address of the tunnel exit point node.
- The original packet, including its header, becomes the payload of the encapsulated packet.
- The Tunnel IPv6 header is processed according to the IPv6 protocol rules.
  - The extension headers, are also processed as though the packet was a standard IPv6 packet.
- Destination options header (extension header) may be used when tunnels are nested.
  - One hop (node) in the tunnel is the entry point of another tunnel.
  - In this case, we have nested tunnels.
    - The first tunnel is called the outer tunnel.
    - The second tunnel is called the inner tunnel.

Tunneling Techniques (14)

- The inner tunnel entry point treats the whole packet received from the outer tunnel as the original packet and encapsulates the whole received packet.
- Every encapsulation adds the size of the tunnel IPv6 headers, which should be limited.
- It allows around 1600 nested tunnels.
  - It needs a mechanism to limit the number of nested tunnels (is specified in RFC 2473).
  - It is called the tunnel encapsulation limit option.
  - This option is carried in a Destination Option header and has the format as follows.

Tunneling Techniques (15)

- Format of the tunnel encapsulation limit option.



- The Option Type field (1-byte):
  - The value is 4: means the Tunnel Encapsulation Limit Option.
- The Option Data Length field (1-byte):
  - The value is 1: the length of the following Option field is 1-byte.

Tunneling Techniques (16)

- The Option field (1-byte):
  - It means how many further levels of encapsulation are permitted.
  - If the value is zero: the packet is discarded and an ICMP Parameter Problem message is sent back to the source (the tunnel entry point of the previous tunnel).
  - If the value is non-zero: the packet is encapsulated and forwarded.
    - Update this value: Value = value – 1.
- If the received packet does not have a tunnel encapsulation limit, but this tunnel entry point has one configured:
  - Using the configured value of the tunnel entry point in the Option field.

Tunneling Techniques (17)

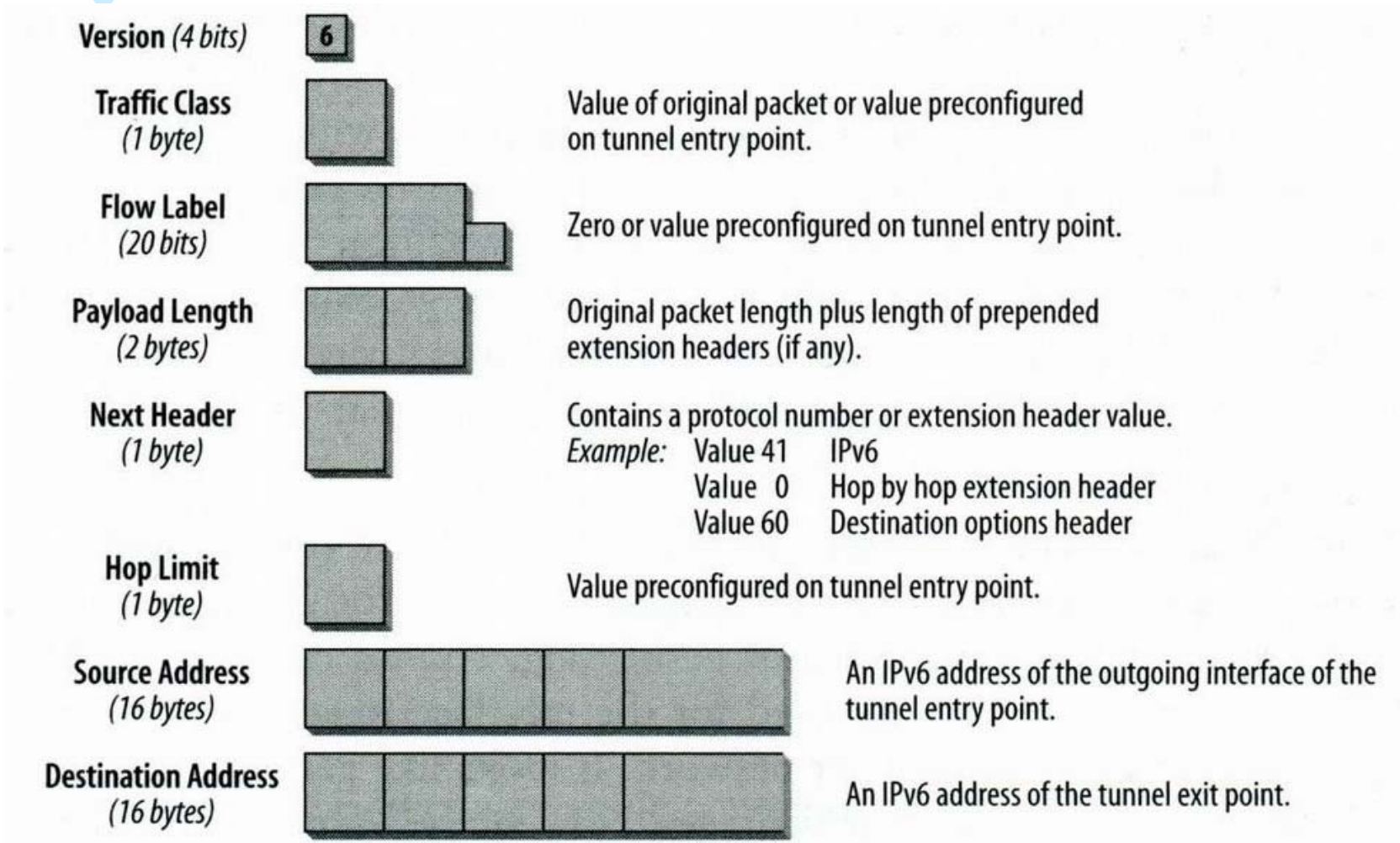
- Question1: Loopback encapsulation should be avoided.
  - Loopback encapsulation happens: When a node encapsulates a packet originating from itself and destined to itself.
  - Solve: By checking and rejecting configurations of tunnels where both the entry point and exit point belong to the same host.
- Question2: Routing-loop nested encapsulation.
  - Routing loop happens: If a packet from an inner tunnel reenters an outer tunnel (the packet has not yet exited).
  - Solve: By checking the original packet's hop limit & the tunnel encapsulation limits.

Tunneling Techniques (18)

- A tunnel entry point node must support fragmentation of packets, which it encapsulates.
  - Because: Packets that are encapsulated may exceed the Path MTU of the tunnel.
    - The tunnel entry point is considered the source of the encapsulated packet, it should fragment it if needed.
- The tunnel exit point node will reassemble the packet.
- If the original packet can not be fragmented:
  - The tunnel entry point discards the packet, and sends an ICMP Destination Unreachable message with the code "fragmentation needed and DF (Don’t Fragment) set" back to the source of the packet.

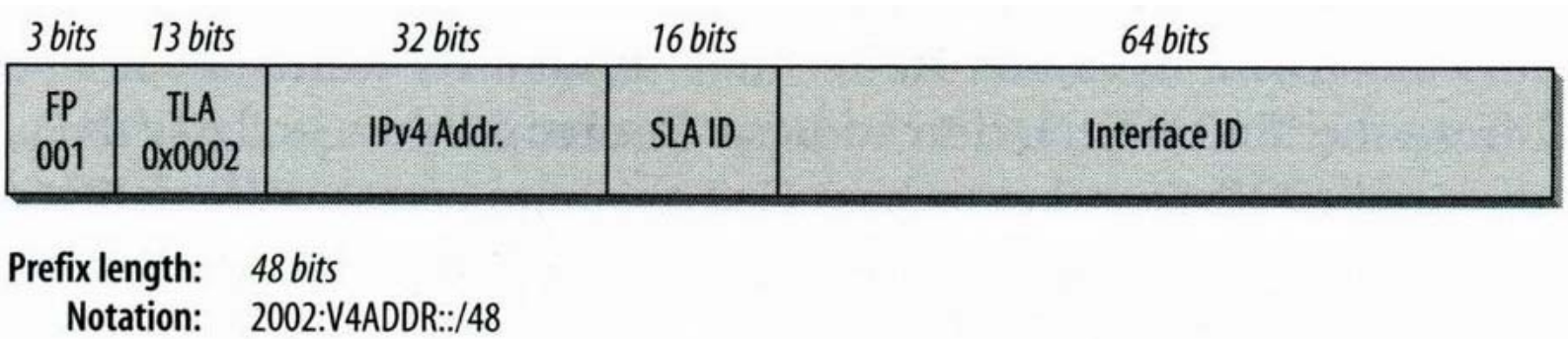
Tunneling Techniques (19)

- The tunnel IPv6 header.



6to4 Tunnel (RFC 3056) (1)

- In 6to4, the IPv4 network is a infrastructure.
- The native (pure) IPv6 domains will communicate via 6to4 routers (6to4 gateways).
- The IPv6 packets are encapsulated in IPv4 header at the 6to4 gateway.
- Globally unique IPv4 unicast address is required on the 6to4 gateway.
- The IANA has assigned a special TLA for the 6to4 scheme:
  - The address prefix is 2002::/16.
  - Format of the 6to4 prefix.



6to4 Tunnel (RFC 3056) (2)

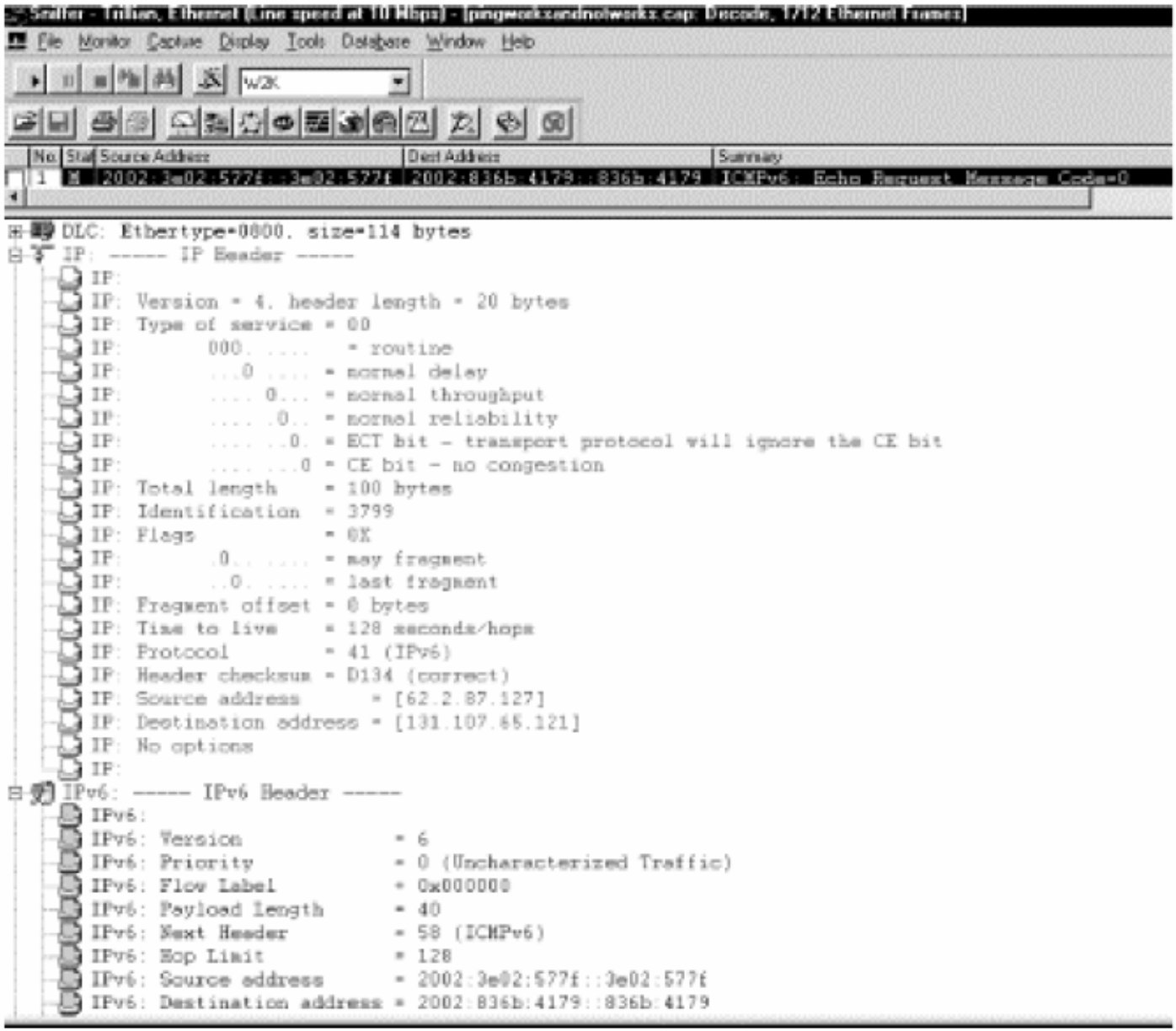
- The 32 bits IPv4 address of the 6to4 gateway are appended after the prefix 2002::/16 (in hex representation).
  - Ex) The IPv4 address 62.2.84.115, its 6to4 prefix is 2002:3e02:5473::/48.
- The remainder 80-bit is for internal network.
- If we want to speak to other IPv6 hosts on remote (pure) IPv6 networks, we need a 6to4 relay router on the other side.
  - The relay router is a router, which is configured for 6to4 and IPv6. It connects the 6to4 network to the native IPv6 network.
  - To simplify the configuration for 6to4 gateways, which need a default route to find a 6to4 relay router on the Internet.
    - RFC 3068 defines a 6to4 relay router anycast address.
    - IANA assigned an IPv4 6to4 relay anycast prefix of 192.88.99/24.
      - Ex) The assigned anycast address for the first 6to4 relay router may be 192.88.99.1.

6to4 Tunnel (RFC 3056) (3)

- In IPv4 header:
  - The TTL field = the Hop Limit in the IPv6 header = 128 (it is a common default value for Microsoft).
  - The Protocol Type field: 41 (is for IPv6).
  - The Source Address: is the globally unique IPv4 address used for the 6to4 router.
  - The Destination Address: is the globally unique IPv4 address used for the 6to4 relay router.
- When two IPv6 hosts communicate:
  - If one only has a 6to4 address, and the other has both 6to4 address & native IPv6 address:
    - The two hosts should both use the 6to4 address for communication.
  - If both hosts have 6to4 & native IPv6 address:
    - They can both use either the 6to4 address or the IPv6 address.

6to4 Tunnel (RFC 3056) (4)

- The IPv4 & IPv6 headers in the trace file.

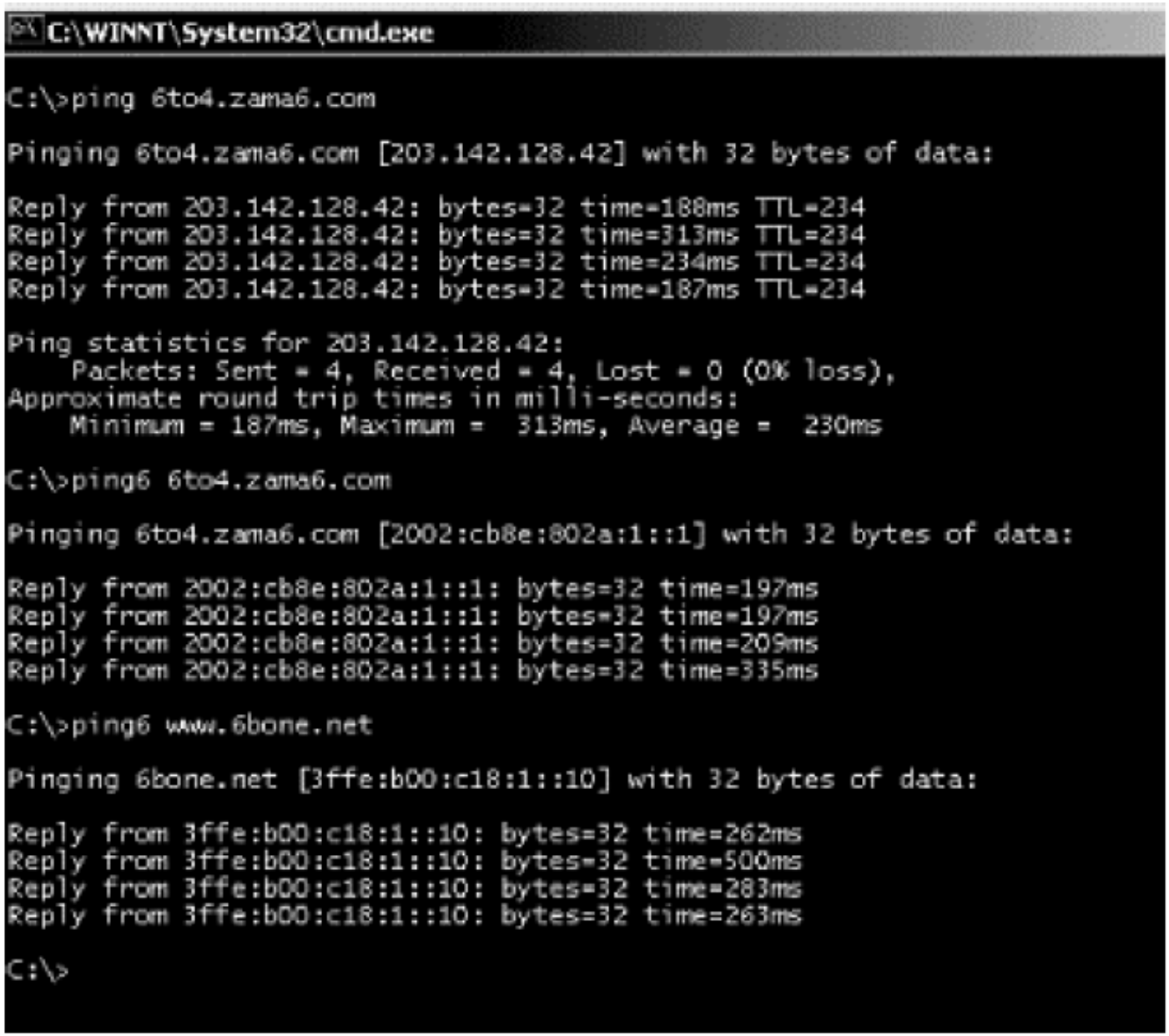


6to4 Tunnel (RFC 3056) (5)

- Ex) A ping IPv4 & IPv6 test.
  - In the public 6to4 relay router list, we find the Zama Networks, which offers a public 6to4 relay router (6to4.zama6.com).
  - The first ping is an IPv4 ping (ping.exe).
    - The 6to4 relay router's IPv4 address is 203.142.128.42.
  - The second ping is an IPv6 ping (ping6.exe).
    - The 6to4 relay router's IPv6 address is 2002:cb8e:802a:1::1.
    - To convert the IPv4 address to hexadecimal:
      - cb8e:802a = 203.142.128.42.
  - The third ping is an IPv6 ping (to a 6Bone host www.6bone.net).
    - It has the IPv6 address 3ffe:b00:c18:1::10.
    - In early, the prefix 3ffe was assigned for 6Bone testing (it is not now).

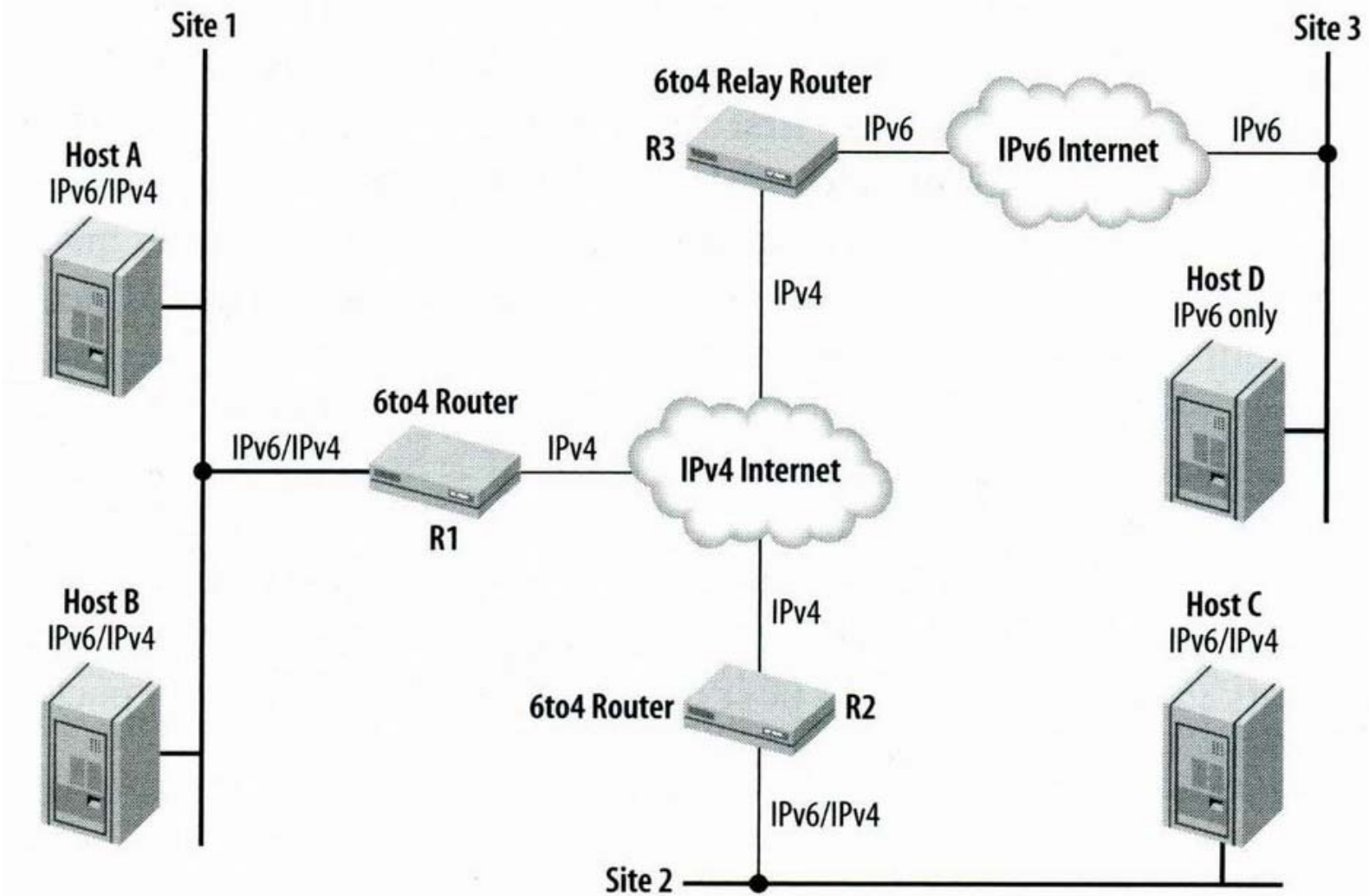
6to4 Tunnel (RFC 3056) (6)

- Pinging the 6to4 relay & a 6bone node.



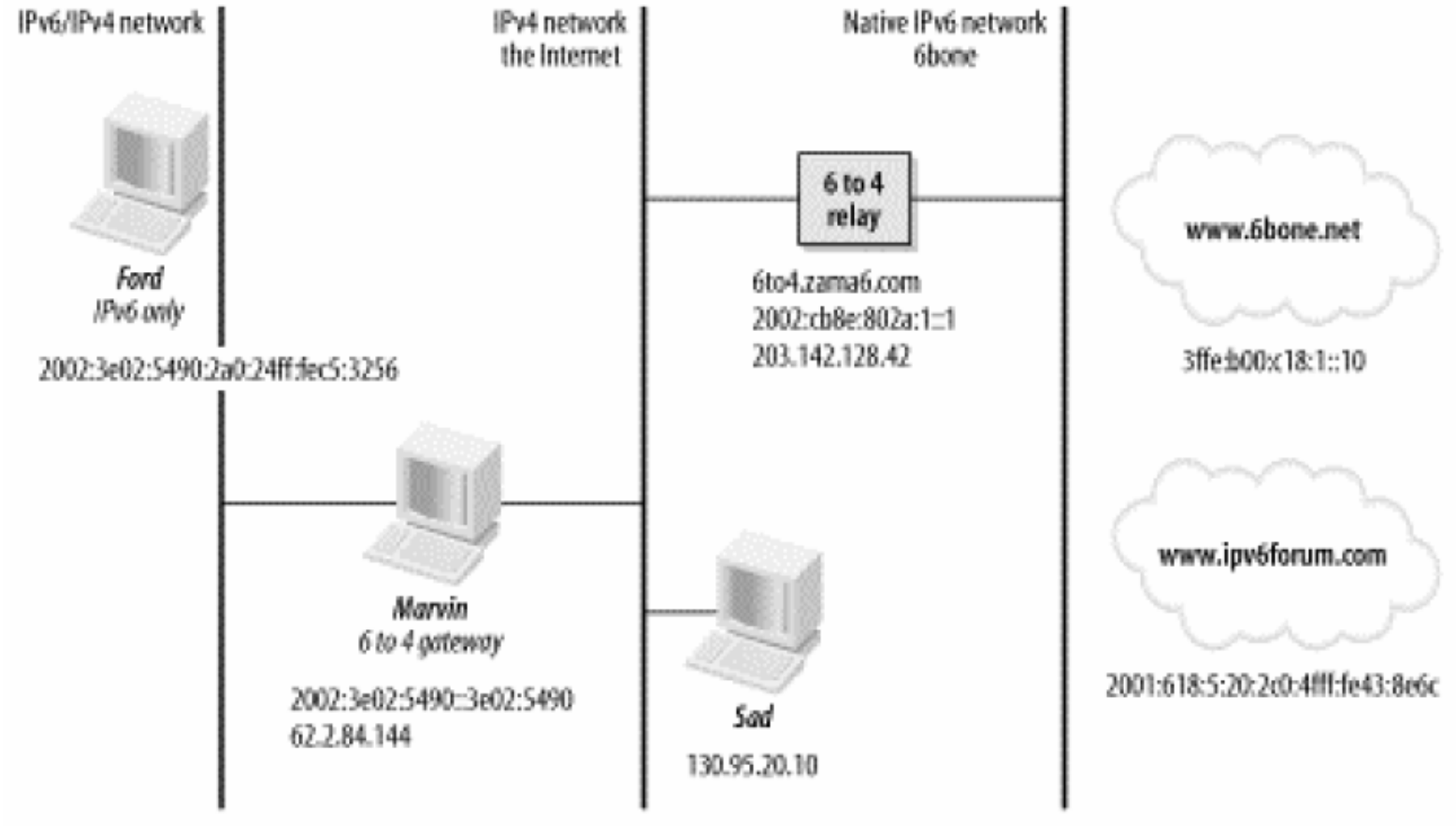
6to4 Tunnel (RFC 3056) (7)

- Shows the 6to4 components and how they play together.



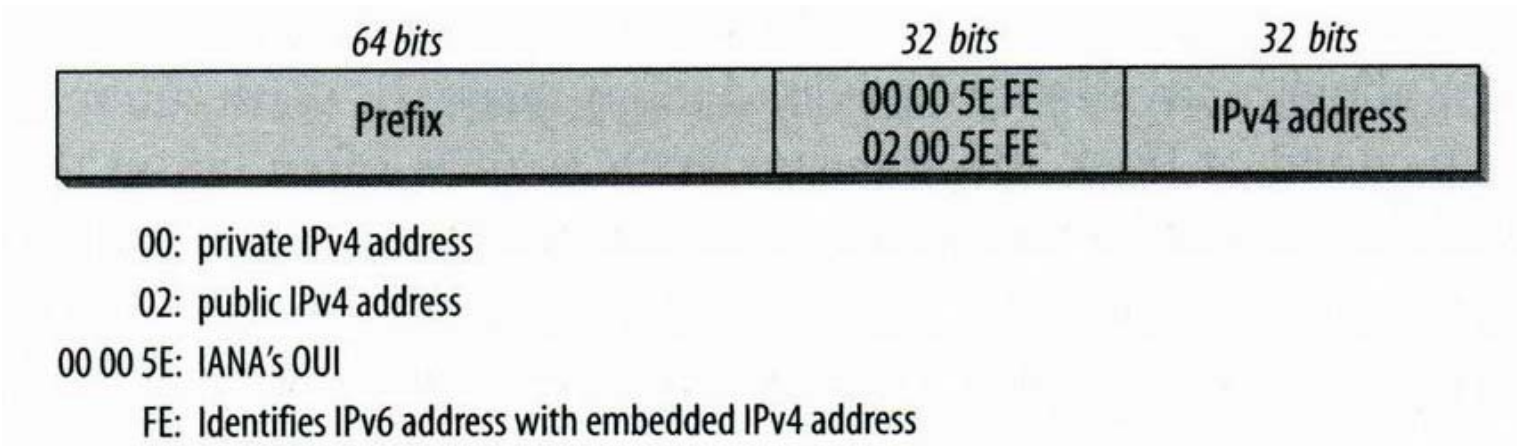
6to4 Tunnel (RFC 3056) (8)

- A 6to4 design.
  - If Ford (IPv6-only) wants to talk to Sad (IPv4-only): it can do by translation mechanisms.



ISATAP Automatic Tunnel (RFC 4214) (1)

- The Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) is designed to provide IPv6 connectivity between IPv6 nodes within a mainly IPv4-based intra-network, which does not have an IPv6 router in the site.
- ISATAP allows hosts to use an automatic tunneling mechanism, if hosts are using private IPv4 addresses & NAT.
- ISATAP addresses embed an IPv4 address in the EUI-64 interface identifier.
- The format of the ISATAP address.



ISATAP Automatic Tunnel (RFC 4214) (2)

- The ISATAP address has a standard 64-bit prefix, which can be:
  - Link-local, site-local, a 6to4 prefix, or belong to the global aggregatable unicast prefix.
- The 64-bit Interface identifier includes:
  - The IANA OUI: 00 00 5E.
    - The first byte 00: is for private IPv4 address.
    - The first byte 02: is for public IPv4 address.
  - A 1-byte Type field: the value FE indicates that this address contains an embedded IPv4 address.
  - The last four bytes are IPv4 address: it can be written in dotted decimal notation.
- The format of the address can be summarized as:
  - 64bitPrefix:5EFE:IPv4address.
  - Ex) An assigned prefix is 2001:620:600:200::/64.
    - An IPv4 address is 62.2.84.115.

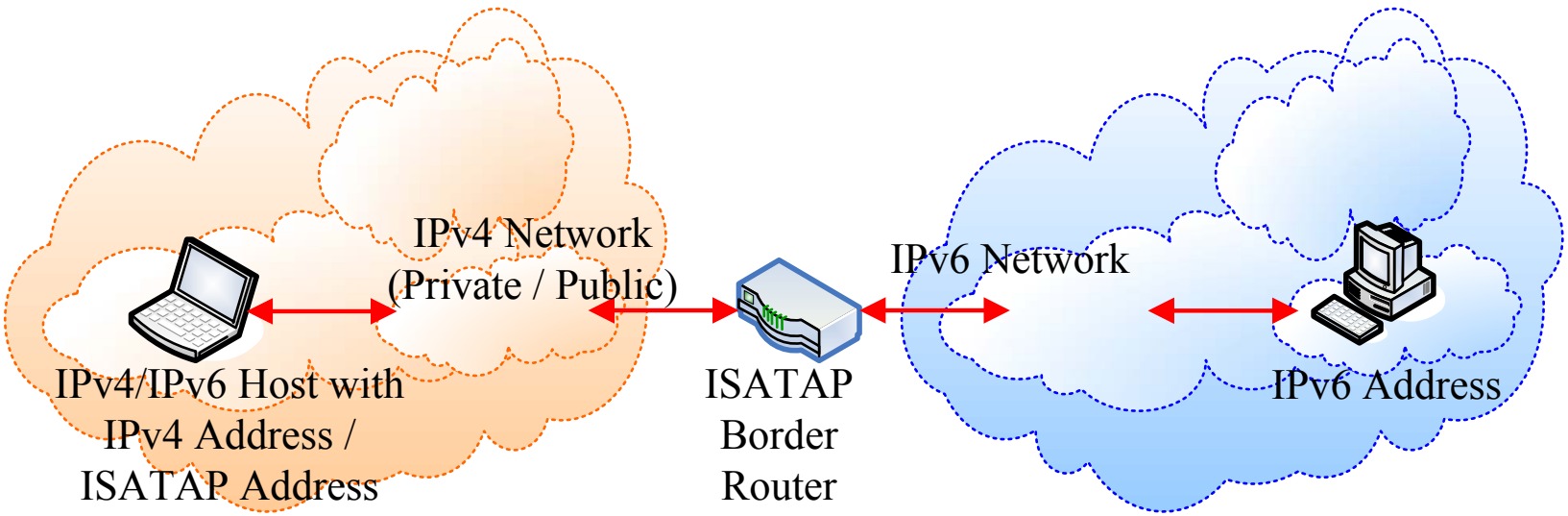
ISATAP Automatic Tunnel (RFC 4214) (3)

- The ISATAP address is:
  - 2001:620:600:200:0:5EFE:3e02:5473.
  - Or 2001:620:600:200:0:5EFE:62.2.84.115.
- The link local address is:
  - FE80::5EFE:62.2.84.115.

- Using ISATAP, IPv6 hosts within an IPv4 intranet can communicate with each other.
  - It does not need the IPv6 routers.
- If these IPv6 hosts want to communicate with other IPv6 hosts in Internet (such as 6Bone hosts), a border router must be configured.
  - The border router can be:
    - An ISATAP router or a 6to4 gateway.
- The IPv4 addresses of these IPv6 hosts can be private IPv4 addresses.

ISATAP Automatic Tunnel (RFC 4214) (4)

- ISATAP tunnel scenario with border router.



ISATAP Automatic Tunnel (RFC 4214) (5)

- Several ISATAP hosts can be assigned to one ISATAP prefix.
  - They form an IPv6 subnet (domain).
- Because the ISATAP nodes on the IPv4 network do not have an IPv6 router, which can send the prefix information for auto-configuration:
  - These hosts need to be manually configured for their prefixes and their default routes.
  - The default route points to the border router (the ISATAP router or 6to4 gateway).
- The ISATAP is developing:
  - Automatic intrasite IPv6 router discovery & stateless address autoconfiguration.

Teredo Tunnel (RFC 4380) (1)

- 6to4 tunnel mechanism uses public IPv4 addresses.
- ISATAP enables IPv6 hosts within a site regardless of whether they use public or private IPv4 addresses (no IPv6 routers in the site).
- Teredo is designed to make IPv6 hosts behind a or more NAT (in IPv4 network) to communicate with IPv6 networks (RFC 4380).
  - It is by tunneling packets over UDP.
- NAT creates issues:
  - 1st: NAT users have a private IPv4 address.
  - 2nd: Many NATs are configured to perform ingress filter.
    - They do not allow many types of packets to go through.

Teredo Tunnel (RFC 4380) (2)

- The following terms are used with Teredo:
  - Teredo service.
  - Teredo Client.
    - It is a node.
  - Teredo Server.
    - It is a node, which helps to maintain the address/port mapping table in NAT, and helps to construct the tunnel between Teredo Client & Teredo Relay.
  - Teredo Relay.
    - An IPv6 relay router, which transports data packet between Teredo Client and IPv6 network.
  - Teredo IPv6 Service Prefix.
    - It is for the Teredo Client, the global Teredo prefix assigned by IANA is 2002:0000::/32
  - Teredo UDP Port.
    - The UDP ports of Teredo Server & Relay are both 3544.

Teredo Tunnel (RFC 4380) (3)

- Teredo Service Port.
  - Teredo Client sends Teredo packets from this UDP port (any value).
- Teredo Server Address.
  - Teredo Server IPv4 address.
- Teredo IPv6 Client Prefix.
  - A global IPv6 prefix composed of:
    - (1) Teredo IPv6 Service Prefix.
    - (2) Teredo Server Address.
- Teredo Node Identifier.
  - A 64-bit identifier composed of:
    - (1) The mapped UDP port of Teredo Client in NAT.
    - (2) The mapped IPv4 address of Teredo Client in NAT.
    - (3) A flag indicates the type of NAT.

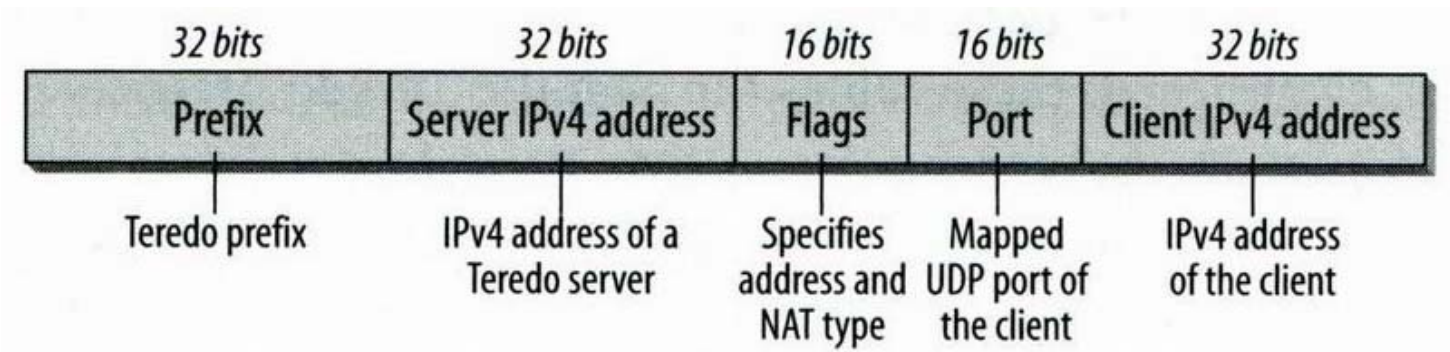
Teredo Tunnel (RFC 4380) (4)

- Teredo IPv6 Address.
  - It contains:
    - (1) Teredo IPv6 Client Prefix.
    - (2) Teredo Node Identifier.
- Teredo-mapped Address and Teredo-mapped Port.
  - In NAT, a global IPv4 address & UDP port are assigned for mapping Teredo Client IPv4 address & UDP port.
- Teredo Bubble.
  - A minimal IPv6 packet with a null payload (the payload type is 59: No Next Header).
  - Teredo clients & relays use it to create a mapping table in a NAT.
- Teredo Refresh Interval.
  - The time interval of a Teredo IPv6 address, which is expected to remain valid.
  - Default, clients assume an interval value of 30 seconds.

Teredo Tunnel (RFC 4380) (5)

- Teredo Secondary Port.
  - A UDP port used to send / receive packets to determined the appropriate value of the Teredo Refresh Interval, but not used to carry any Teredo traffic.
- Teredo IPv6 Discovery Address.
  - An IPv4 multicast address used to discover other Teredo clients.
  - The multicast address is: 224.0.0.253

- The format of the Teredo IPv6 address.



Teredo Tunnel (RFC 4380) (6)

- Research has shown that most NAT can be classified into (RFC 2663):
  - (1) Type Cone NAT, & (2) Restricted Cone NAT.
- Teredo can solve the scenarios of:
  - (1) Cone NATs, (2) Restricted Cone NATs, & (3) Port-restricted Cone NATs.
- Teredo cannot solve the scenarios is:
  - Symmetric NATs.

- In Teredo IPv6 address:
  - The mapped Client Port (16-bit) & the mapped Client IPv4 Address (32-bit) are all obfuscated.
    - Each bit in these two fields is reversed (0 → 1, 1 → 0).
- A Teredo client must be pre-configured with the IPv4 address of its Teredo server.

Teredo Tunnel (RFC 4380) (7)

- On Teredo client booting:
  - Teredo client sends a Router Solicitation (RS) to the All-routers multicast address using its link-local IPv6 address.
  - The RS (IPv6 packet) is sent to the IPv4 address of the Teredo server over UDP.
  - The Router Advertisement (RA) coming back from the Teredo server contains the Teredo IPv6 Service Prefix.
  - The client builds its Teredo IPv6 address.

Silkroad Tunnel

- Silkroad is a new tunnel mechanism, which allows nodes sitting behind a NAT to access the IPv6 Internet (like Teredo tunnel).
- It also use UDP tunnel to go through a NAT.
- It uses 2 devices:
  - Silkroad Navigator.
  - Silkroad Access router (SAR).
- The main different from Teredo is that:
  - Silkroad supports all types of NAT (including Symmetric NATs).
  - Silkroad does not need a special prefix.
    - Teredo prefix: 2001:0000::/32
- Silkroad uses many SARs to connect IPv6 network & IPv4 network.
- A node sitting behind a NAT has to choose a SAR to be a relay router.
  - Which is the SAR more efficient?
  - The information can be provided by Silkroad Navigator.

Proto 41 Forwarding (Tunnel)

- Protocol 41 forwarding:
  - Some NAT implementations allow the configuration of IPv6 tunnels from inside of the private LAN to routers or tunnel servers in the Internet.
  - This is a simple and helpful way to provide IPv6 nodes behind a NAT with access to the IPv6 Internet.
  - Many NAT boxes can be configured to forward packets based on the protocol value of 41 (for IPv6) in the IPv4 header.
  - Most of the client OSs already support IPv6.

Tunnel Broker (RFC 3053) (1)

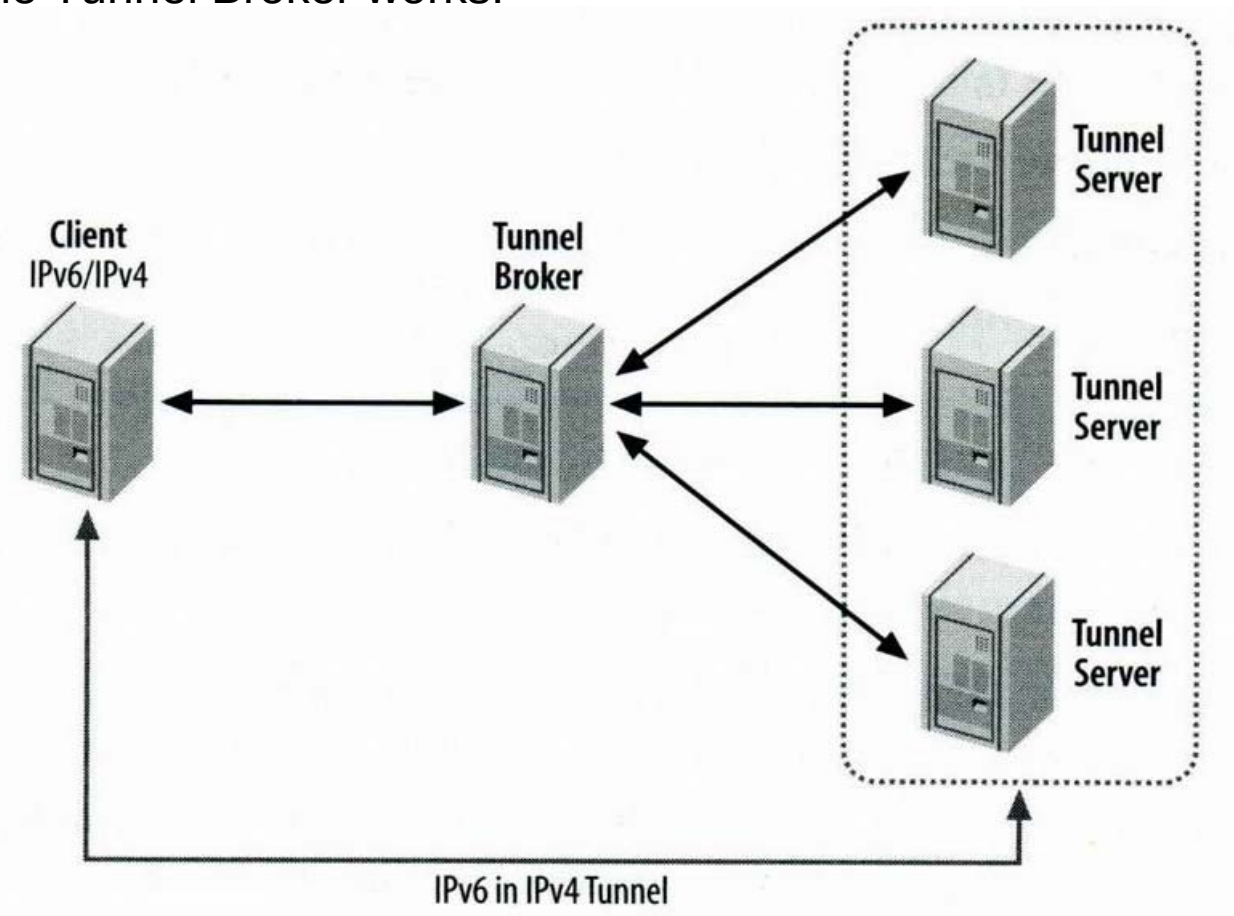
- Tunnel Brokers can be seen as virtual IPv6 providers providing IPv6 Internet connectivity to users, which already have an IPv4 connection to the Internet.
- A user (client) desiring an IPv6 connection registers with the Tunnel Broker.
- The Tunnel Broker sends the configuration information to a Tunnel Server, when it wants to establish, change, or delete a tunnel.
  - Tunnel Broker will choose a Tunnel Server as a tunnel exit point.
    - May be based on load-sharing criteria.
  - It also sends the configuration information back to the client.
    - Including the tunnel parameters & DNS names.
- The Tunnel Broker is a dual-stack node.
  - It must has an IPv4 address.
  - It can also have an IPv6 address, but it is not required.
  - It & Tunnel Server can run over either IPv4 or IPv6.

Tunnel Broker (RFC 3053) (2)

- Tunnel Server is a dual-stack router connected to the global IPv6 Internet.
- The client is also a dual-stack host or router.
  - Registering to the Tunnel Broker → authenticate with standard procedures (e.g., with RADIUS).
- The Tunnel Broker chooses a prefix for the registered client.
  - The prefix can be any length:
    - Site prefix: /48, subnet prefix: /64, single host: /128.
- Today, there are a number of ISPs that offer Tunnel Broker services.
- The Tunnel Broker model is designed for small & isolated IPv6 networks / hosts.
  - It should works only with public IPv4 addresses.

Tunnel Broker (RFC 3053) (3)

- How the Tunnel Broker works.



Dual-stack Transition Mechanism (DSTM) (1)

- Dual-stack IPv6 dominant transition mechanism (DSTM):
  - DSTM is still in draft.
  - DSTM allows the transport of IPv4 packets over an IPv6 network.
    - By encapsulating the IPv4 packets in IPv6 Packets.
  - The following terms for DSTM:
    - DSTM Domain.
      - In this network, IPv6/IPv4 nodes use DSTM to communicate with IPv4 nodes.
      - An IPv4 Address Allocation Server is implemented to manage an IPv4 address pool.
    - DSTM Client.
      - An IPv4/IPv6 node with DSTM client software.
    - DSTM Server.
      - An IPv4/IPv6 node with DSTM server software.
      - It maintains the IPv4 address pool.

Dual-stack Transition Mechanism (DSTM) (2)

- DSTM Border Router.
  - An IPv4/IPv6 node with DSTM border router software.
  - It connects the IPv6 network and the IPv4 network.
  - It manages the address mapping (IPv6 to IPv4 addresses).
- Dynamic Tunnel Interface.
  - An interface on a DSTM Client.

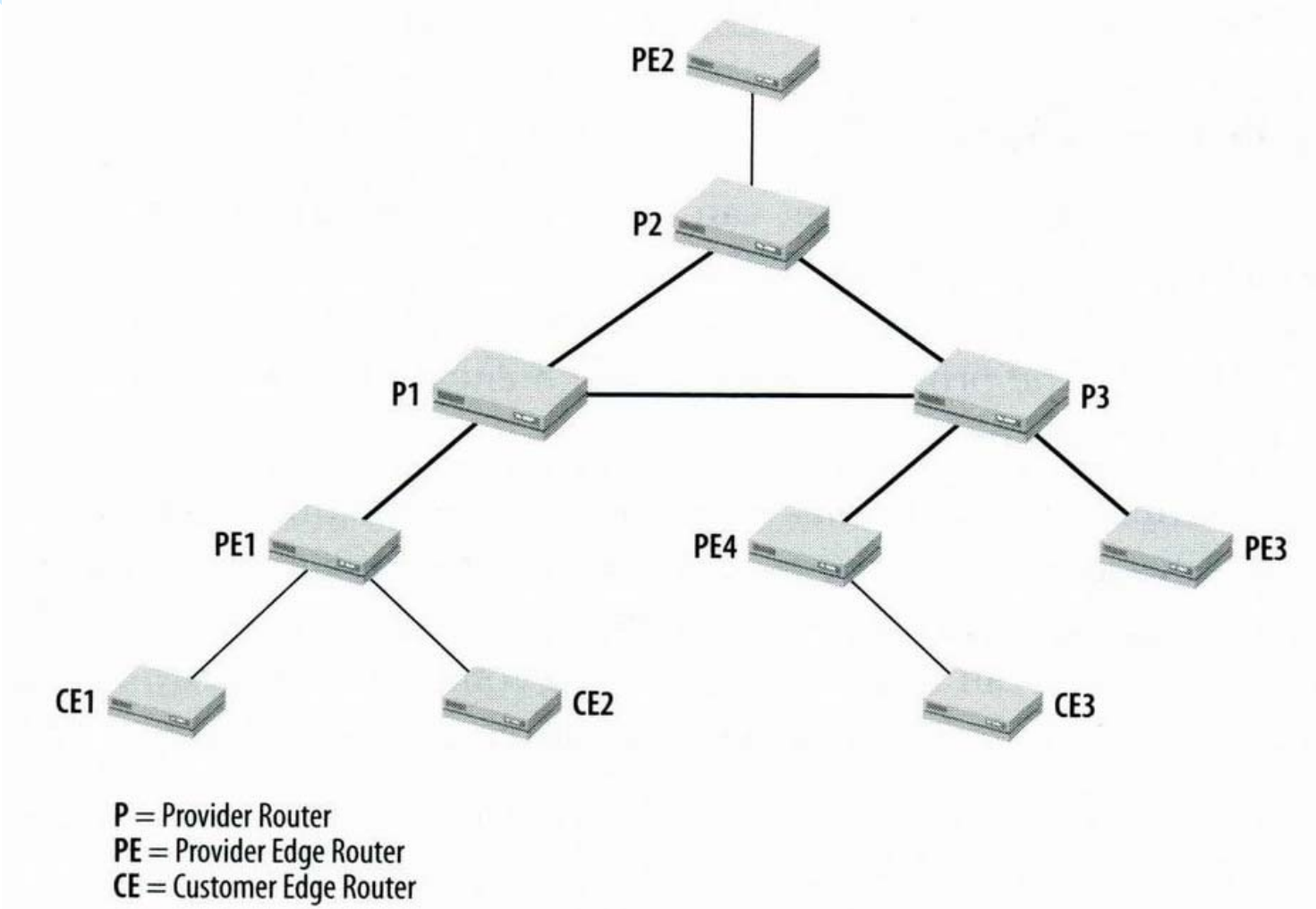
- A DSTM Client is a tunnel endpoint (TEP), which sends the encapsulated IPv6 packets to a DSTM Border Router.
- The DSTM Server & Border Router software can be installed on the same hardware.
- The DSTM client can receive its tunnel endpoint configuration from DHCPv6.

Other Tunnels (1)

- IPv4/IPv6 Coexistence by using VLANs.
- IPv6 in MPLS networks.
- Cisco’s 6PE.
- Generic routing encapsulation (GRE).
- SSH (Secure SHell) tunnels.

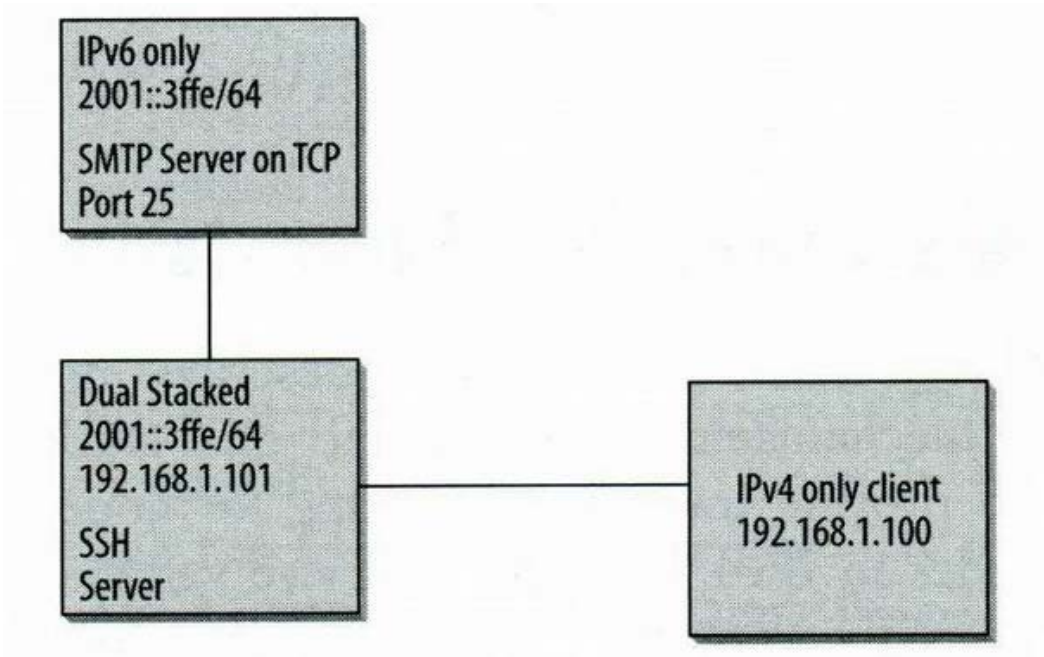
Other Tunnels (2)

- MPLS routing hierarchy.



Other Tunnels (3)

- IPv4 client connects to IPv6-only server through a dual-stack SSH host.

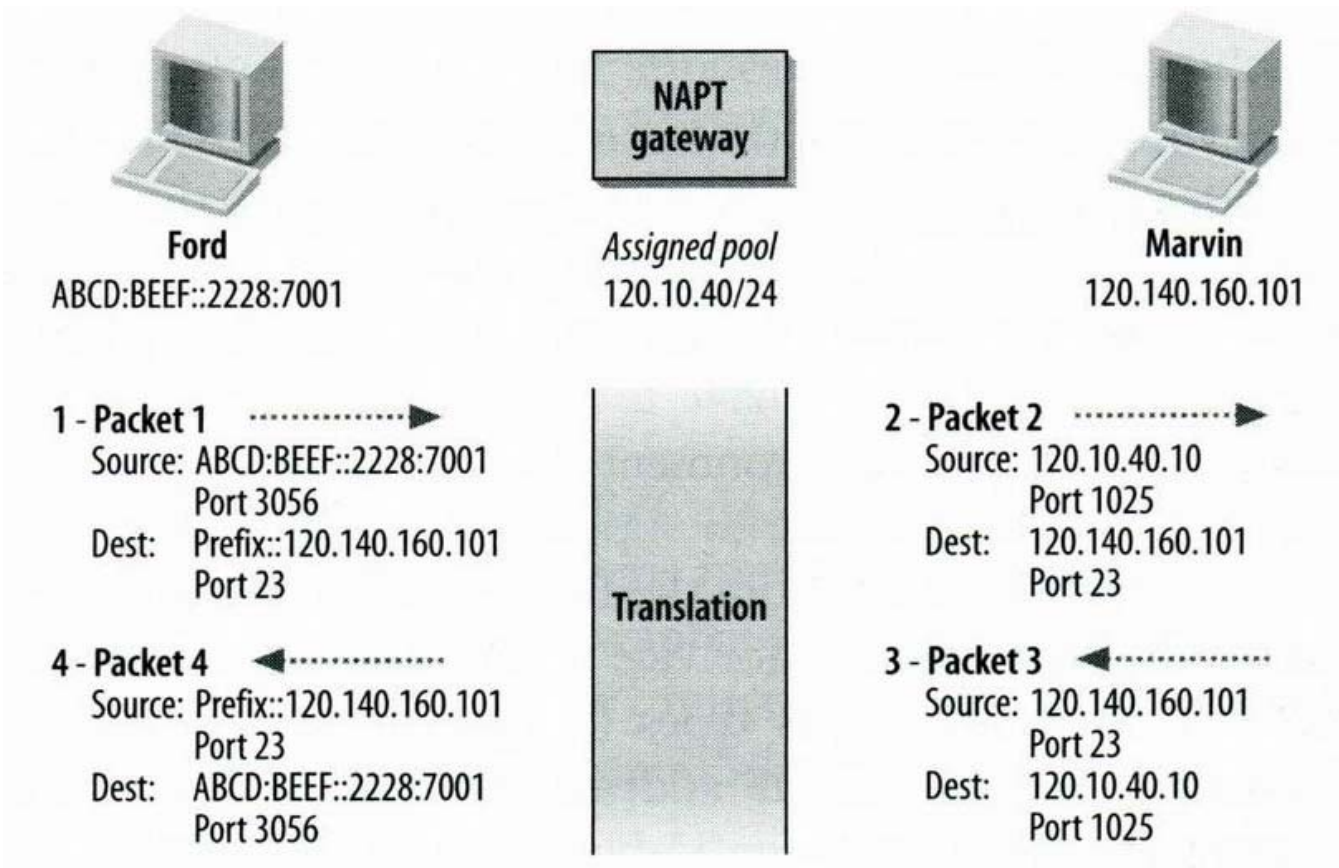


Network Address and Protocol Translation (1)

- Stateless IP/ICMP Translation.
  - Translating IPv4 to IPv6.
  - Translating ICMPv4 to ICMPv6 and vice versa.
  - Translating IPv6 to IPv4.
- NAT-PT.
- Limitations.

Network Address and Protocol Translation (2)

- Communication flow over NAT-PT.



Other Translation Techniques (1)

- Bump-in-the-stack.
- Bump-in-the-API.
- Transport relay translator.

Comparsion (1)

- Dual stack.
- Tunneling.
- NAT-PT.
- When to choose IPv6?

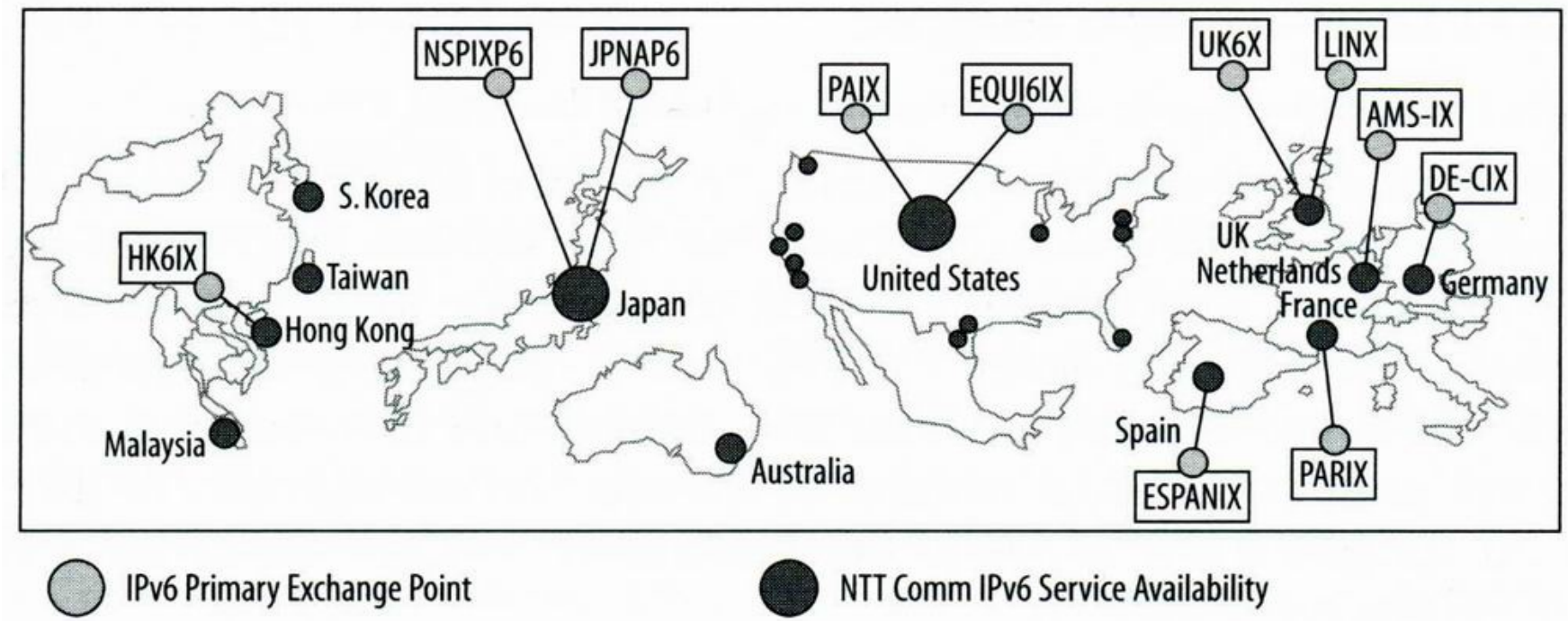
Integration Scenarios & Case Studies (1)

- Prganizations.
- ISPs.

- NTT communications: An ISP case study.
- University of Proto.
  - Access/perimeter technolopy.
  - Core and vertical distribution.
  - Network services.
  - Security.
  - Cost of introduction.
- University of Strasbourg.

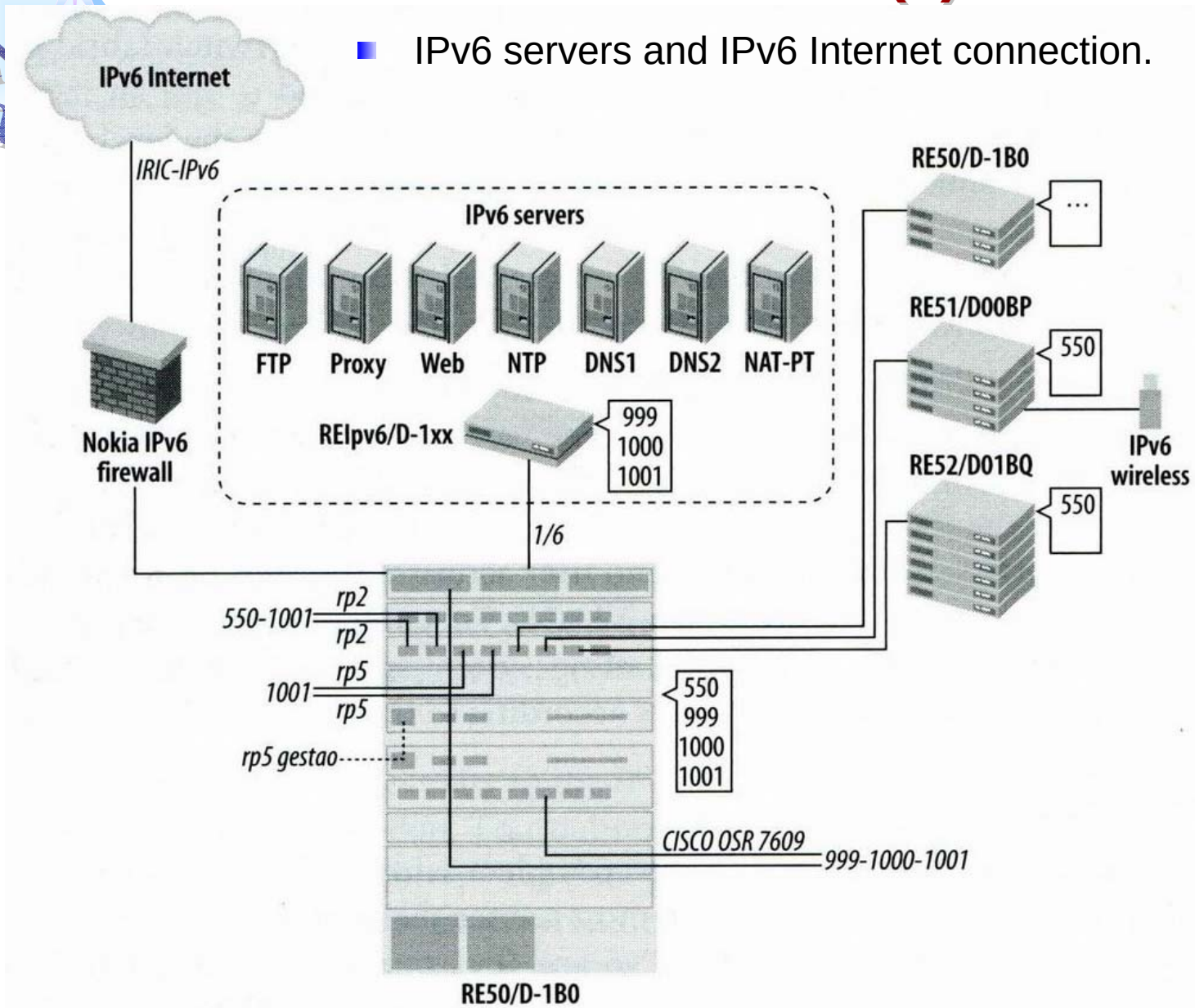
Integration Scenarios & Case Studies (2)

- NTT communication's IPv6 network map highlighting IPv6 peering points.



Integration Scenarios & Case Studies (3)

- IPv6 servers and IPv6 Internet connection.



Integration Scenarios & Case Studies (4)

- This book has been reviewed over IPv6.
- Moon6: The largest IPv6 test network.
  - Phase 1.
  - Phase 2.
  - Phase 3.
- What is missing?
  - IPv6 routing.
  - Protocol selection on dual-stack nodes.
  - Multihoming with IPv6.
  - DNS.
  - Network management.
  - IPv4 dependencies.

Integration Scenarios & Case Studies (5)

- Security Aspects:
- Applications.
- Cost of introduction.
  - Hardware and operating systems (OS).
  - Software.
  - Education.
  - Planning.
  - Other costs.

Vender Support (1)

- Operating systems (OS).
- Router support.
- IP address management.
- Firewalls.

Stateless IP/ICMP Translation (1)

Table 10-1. Translated IPv6 header fields

Header field	Information
Version	6
Traffic Class	All 8 bits from the Type of Service and Precedence Field are copied.
Flow Label	Zero
Payload Length	The Total Length from the IPv4 header field minus the size of the IPv4 header (including options, if present).
Next Header	Protocol Field copied from the IPv4 header.
Hop Limit	TTL value copied from the IPv4 header. Since the translator is a router, the value has to be decremented by one (either before or after translation) and checked on the value. If zero, an ICMP TTL exceeded message must be generated.
Source Address	Combination of IPv4-mapped address prefix and the IPv4 address in the 32 low-order bits, for example: `::ffff:0:0:192.168.0.1`.
Destination Address	Combination of IPv4-translatable address prefix and the IPv4 destination address, for example: `0::ffff:0:0:0:192.168.0.99`.
IPv4 Options	If any IPv4 options are present, they are ignored. If a source route option is present, the packet must be discarded and an ICMPv4 "destination unreachable/source route failed" (Type3, Code 5) error message should be returned to the sender.

Stateless IP/ICMP Translation (2)

Table 10-2. IPv6 header fields with fragmentation	
Header field	**Information**
Header fields	
Payload Length	The Total Length from the IPv4 header field minus the size of the IPv4 header (including options, if present) plus 8 bits for the size of the Fragment header.
Next Header	44 (Fragment Header)
Fragment Header fields	
Next Header	Protocol field copied from IPv4 header.
Fragment Offset	Fragment Offset field copied from IPv4 header.
M-Flag	More Fragments bit copied from the IPv4 header.
Identification	The high-order 16 bits are set to zero; the low-order 16 bits are copied from the Identification field in the IPv4 header.

Stateless IP/ICMP Translation (3)

Table 10-3. Translation of ICMPv4 query messages

Message type	Translation
Echo and Echo Reply (types 8 and 0)	Adjust the type to 128 or 129, respectively and adjust the ICMP checksum to take the type change into account and include the ICMPv6 pseudoheader.
Information Request/Reply (types 15 and 16)	Obsoleted in ICMPv6. Silently discard.
Timestamp and Timestamp Reply (types 13 and 14)	Obsoleted in ICMPv6. Silently discard.
Address Mask Request/Reply (types 17 and 18)	Obsoleted in ICMPv6. Silently discard.
Router Advertisement (type 9)	Single hop message. Silently discard.
Router Solicitation (type 10)	Single hop message. Silently discard.
Unknown ICMPv4 types	Silently discard.

Stateless IP/ICMP Translation (4)

Table 10-4. Translation of ICMPv4 error messages

Message type	Translation
Destination Unreachable (type 3)	For all codes not listed here, the type is set to one.
Code 0/1, Network/Host Unreachable	Type 1, Code 0 - No Route to Destination.
Code 2, Protocol Unreachable	Type 4, Code 1 - Port Unreachable. Make the pointer point to the IPv6 Next Header field.
Code 3, Port Unreachable	Type 1, Code 4 - Port Unreachable.
Code 4, Fragmentation needed but DF set	Type 2, Code 0 - Packet Too Big. The MTU field needs to be recalculated to reflect the difference between the IPv4 and the IPv6 header sizes.
Code 5, Source Route Failed	Type 1, Code 0 - No Route to Destination (note that source routes are not translated).
Code 6, 7, Destination Network/Host	Type 1, Code 0 - No Route to Destination.

Stateless IP/ICMP Translation (5)

Unknown	
Code 8, Source Route Isolated	Type 1, Code 0 - No Route to Destination.
Code 9, 10, Communication with Destination Administratively Prohibited	Type 1, Code 1 - Communication with Destination Administratively Prohibited.
Code 11, 12, Network/Host Unreachable for TOS	Type 1, Code 0 - No Route to Destination.
Redirect, Type 5	Single hop message. Silently discard.
Source Quench, Type 4	Obsoleted in ICMPv6. Silently discard.
Time Exceeded, Type 11	Type 3 - Time Exceeded. Code field unchanged.
Parameter Problem, Type 12	Type 4 - Parameter Problem. The pointer needs to be updated to point to the corresponding field in the translated and included IP header.

Stateless IP/ICMP Translation (6)

Table 10-5. Translated IPv4 header	
Header field	**Information**
Version	4
Internet Header Length	5 (no options)
TOS and Precedence	All 8 bits from the Traffic Class are copied.
Total Length	Payload Length from the IPv6 header plus length of the IPv4 header.
Identification	Zero
Flags	More Fragments Flag set to zero; Don't Fragment Flag set to one.
Fragment Offset	Zero
Time to Live	Hop Limit value copied from the IPv6 header. Since the translator is a router, the value has to be decremented by one (either before or after translation) and checked on the value. If zero, an ICMP TTL exceeded message has to be generated.
Protocol	Next Header field copied from the IPv6 header.
Header Checksum	Computed after generation of the IPv4 header.
Source Address	If the IPv6 address is an IPv4-translated address, the low-order 32 bits of the IPv4-translated source address are copied to the IPv4 Source Address field. Otherwise, NAT will assign an IPv4 address out of the configured address pool and copy it into the IPv4 Source Address field.
Destination Address	The low-order 32 bits of the IPv4-mapped destination address are copied to the IPv4 Destination Address field.
Options	If an IPv6 Hop-by-Hop Options header, Destination Options header, or a routing header

Stateless IP/ICMP Translation (7)

Table 10-6. Translating the Fragment header

Header field	Information
Total Length	Payload Length from the IPv6 header, minus 8 for the Fragment header, plus the size of the IPv4 header.
Identification	Copied from the low-order 16 bits in the Identification field in the Fragment header.
Flags	More Fragment flag copied from the M flag in the Fragment header. The Don't Fragment flag is set to zero so IPv4 routers can fragment the packet.
Fragment Offset	Fragment Offset field copied from IPv6 header.
Protocol	Next Header value copied from the Fragment header.

Stateless IP/ICMP Translation (8)

Table 10-7. Translation of ICMPv6 informational messages	
Message type	**Translation**
Echo Request and Echo Reply (types 128 and 129)	Adjust the type to 8 and 0, respectively, and adjust the ICMP checksum to take the type change into account and to exclude the ICMPv6 pseudoheader.
MLD Multicast Listener Query/Report/Done (types 130, 131, 132)	Single-hop message. Silently discard.
Neighbor Discover Messages (types 133 to 137)	Single-hop message. Silently discard.
Unknown Informational Messages	Silently discard.
Address Mask Request/Reply (types 17 and 18)	Obsoleted in ICMPv6. Silently discard.
Router Advertisement (type 9)	Single hop message. Silently discard.
Router Solicitation (type 10)	Single hop message. Silently discard.
Unknown ICMPv4 types	Silently discard.

Stateless IP/ICMP Translation (9)

Table 10-8. Translation of ICMPv6 error messages	
Message type	**Translation**
Destination Unreachable (type 1)	Set the Type field to 3 and the code field as follows:
Type 1, Code 0 - No Route to Destination	Type 3, Code 1 - Host Unreachable.
Type 1, Code 1 -	Type 3, Code 10 - Communication with Destination Administratively

Stateless IP/ICMP Translation (10)

Communication with Destination Administratively Prohibited	Prohibited.
Type 1, Code 2 - Beyond Scope of Source Address	Type 3, Code 1 - Host Unreachable.
Type 1, Code 3 - Address Unreachable	Type 3, Code 1 - Host Unreachable.
Type 1, Code 4 - Port Unreachable	Type 3, Code 3 - Port Unreachable.
Packet Too Big (type 2)	Type 3, Code 4 - Fragmentation Needed but DF set. The MTU field needs to be adjusted for the difference between the IPv6 and IPv4 headers, including a Fragment header, if present.
Time Exceeded (type 3)	Type 11, Code field unchanged.
Parameter Problem (type 4)	If the Code field is set to 1, translate to a Type 3, Code 2 - Protocol Unreachable, otherwise to Type 12, Code 0 - Parameter Problem. The pointer needs to be updated to point to the corresponding field in the translated and included IP header.
Unknown Error messages	Silently discard.