IPv6 Transition Techniques

Nalini Elkins
CEO
Inside Products, Inc.
Nalini.elkins@insidethestack.com

Why Transition Techniques?

IPv4 Only

IPv6 Only

Timeline?

IPv4 Only

2012

IPv6 Only

2022

How to get from here to there?

IPv4 Only

2012

Translation
Tunneling
Dual Stack

IPv6 Only

2022

Why Now?

IANA ran out of IPv4 addresses in 2011

RIR	Projected Exhaustion Date	Remaining Addresses in RIR Pool (/8s)
APNIC:	19-Apr-2011	0.9290
RIPENCC:	28-Jul-2012	1.8280
ARIN:	04-Feb-2013	3.5250
LACNIC:	17-Jan-2014	3.4247
AFRINIC:	28-Oct-2014	4.1924

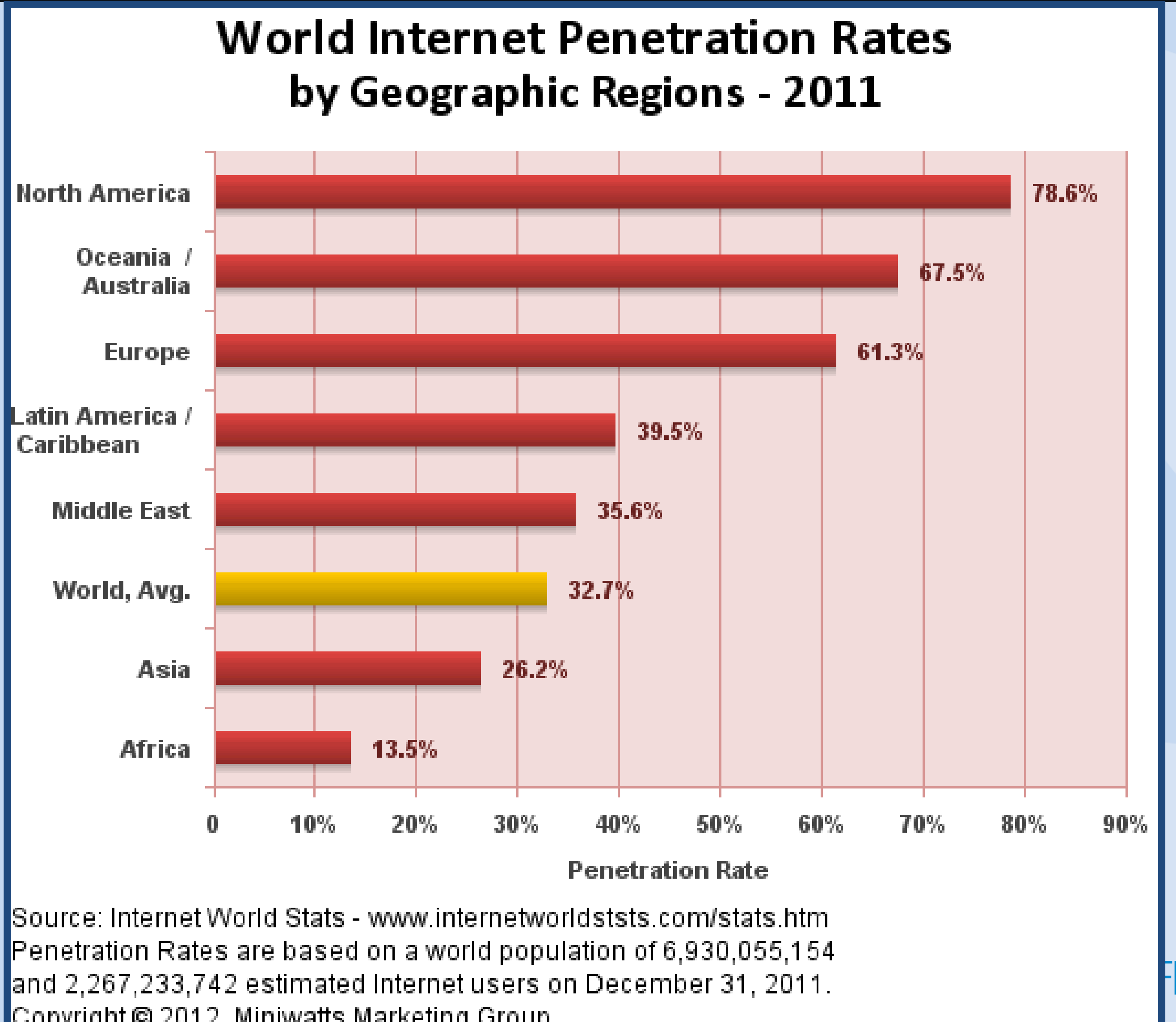
So, now what?

- In the next 5 years:
  - Some ISP will run out of IPv4 addresses
  - Some customers of that ISP will get IPv6 addresses.
  - How will they get to IPv4 only websites: for example: www.mybank.com?
  - Yes, ISPs are offering tunneling but…
    - What is the performance?
    - Security risks?
    - What will it cost?

The Killer App!

The Internet!

Internet Penetration by Continent



FEST '12

2012 and 2014 Federal Mandates

- Upgrade public/external facing servers and services (e.g. web, email, DNS, ISP services, etc) to operationally use native IPv6 by the end of FY 2012;
- Upgrade internal client applications that communicate with public Internet servers and supporting enterprise networks to operationally use native IPv6 by the end of FY 2014;

How to Start?

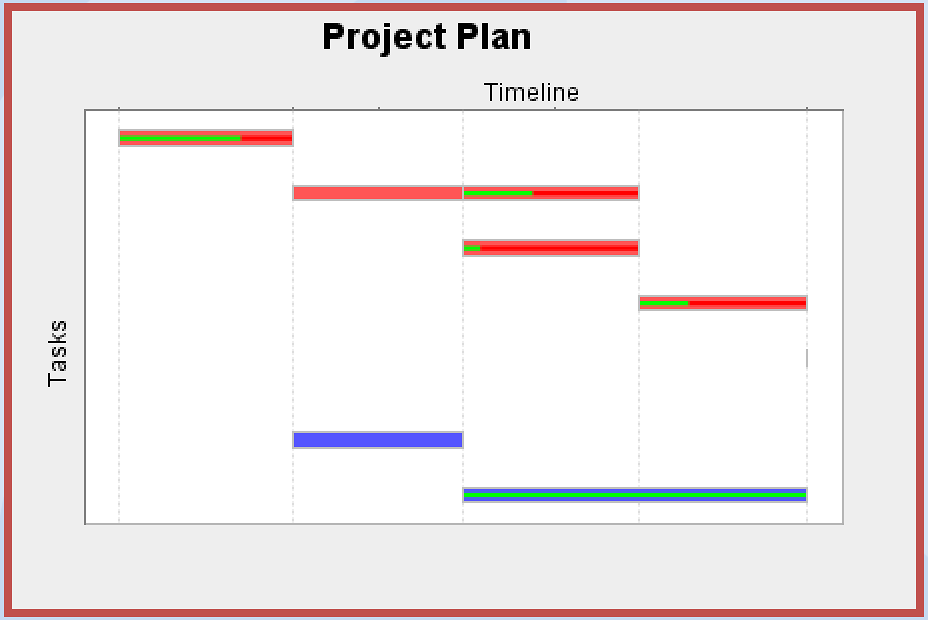
- Organizations are like ships.
- Larger the ship, larger the turning radius.

- Many groups need to be involved (security, applications, network hardware, systems, operations, help desk, vendors.)
- Lights out data centers / automated operations
- Team approach is imperative.

What is this team going to do?

- A roadmap for implementation.
- Timelines and schedules.

- Tasks to be done
  - IPv6 Address Allocation
  - IPv6 Addressing Plan
  - Impact on IPv4 Communications
  - Impact on Applications
  - Types of IPv4/IPv6 Communications
  - Impact on SLAs
  - IPv6 Security
  - Network Services Supported (DNS, DHCP)
  - Campus Networks
  - New IPv6 Capabilities (e.g., mobility, sensors)



But First….

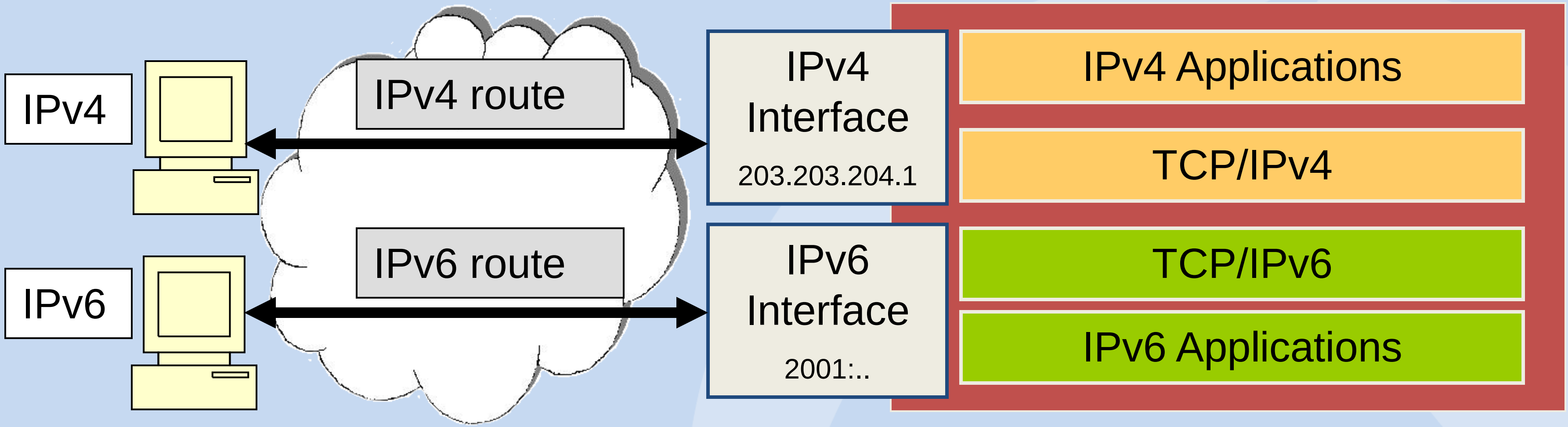
- External facing equipment! (Web server, DNS, email)
- Possible government interface
- How?

Agenda

- In this session, we will discuss:
  - Dual stack mode
  - Tunneling
  - Translation

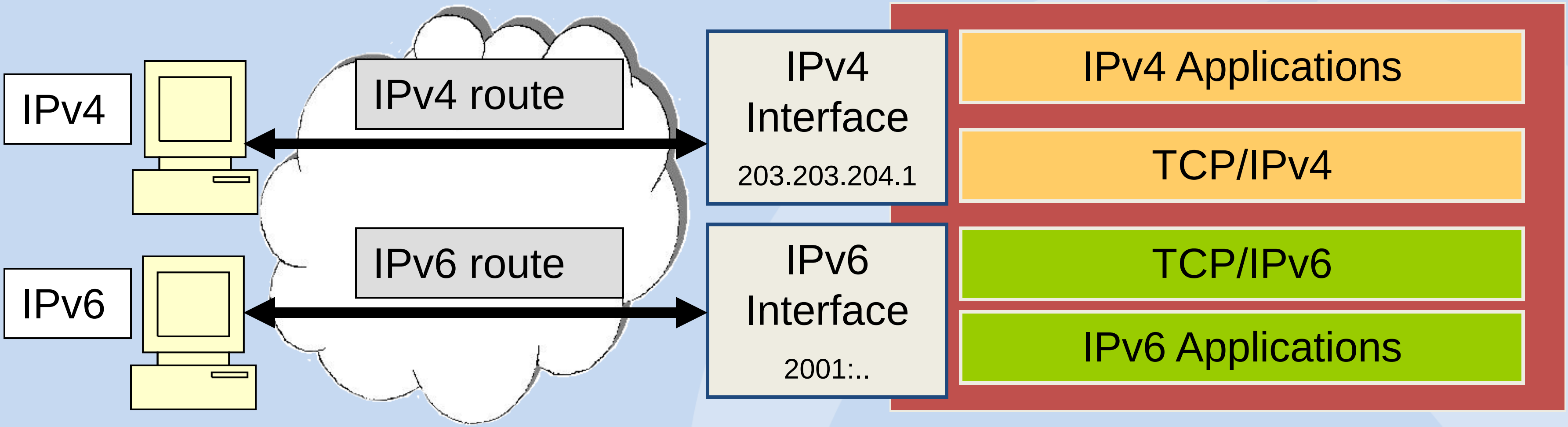
- For each method, we will discuss the
  - Technology,
  - Benefits,
  - Drawbacks,
  - Security issues

Dual Stack Mode



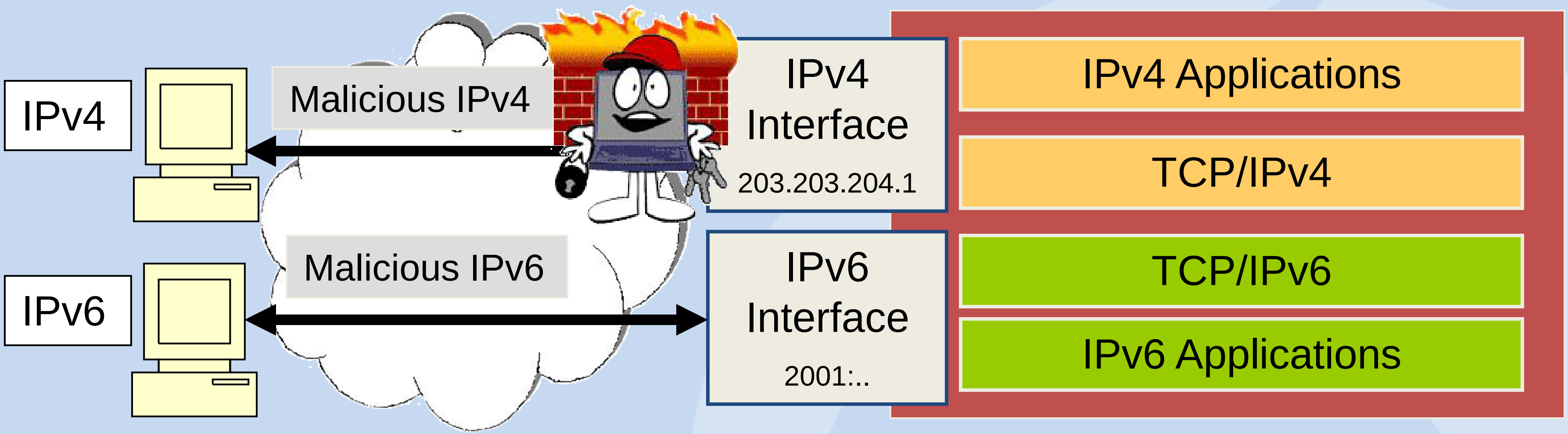
- Either a router or a host may be dual stack.
- A dual stack node runs both an IPv4 and IPv6 TCP/IP stack.
- Such nodes can send and receive both IPv4 and IPv6 packets over separate routes.

Dual Stack Mode – Implications?



- Are all applications going to be rewritten to support IPv6?
- What is preferred?
- What is the performance?
- Will IPv6 impact IPv4?

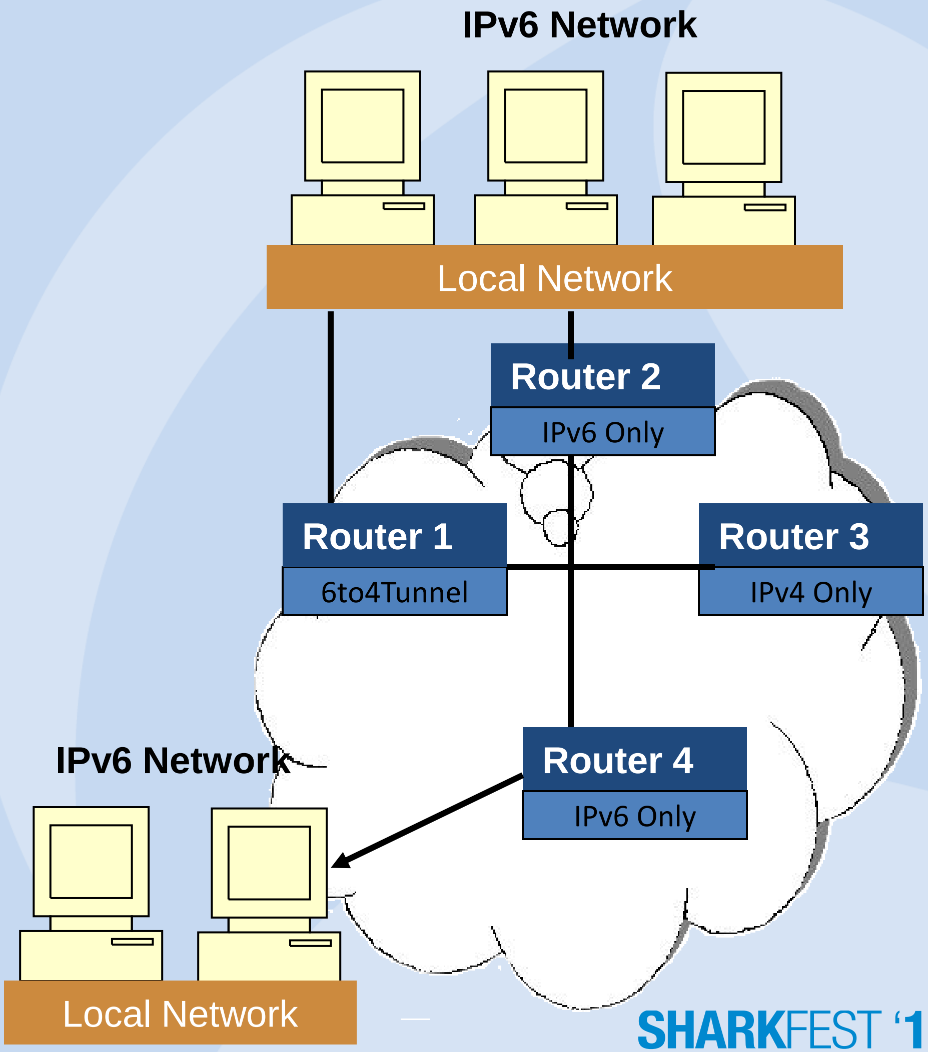
Security Issues Dual Stack



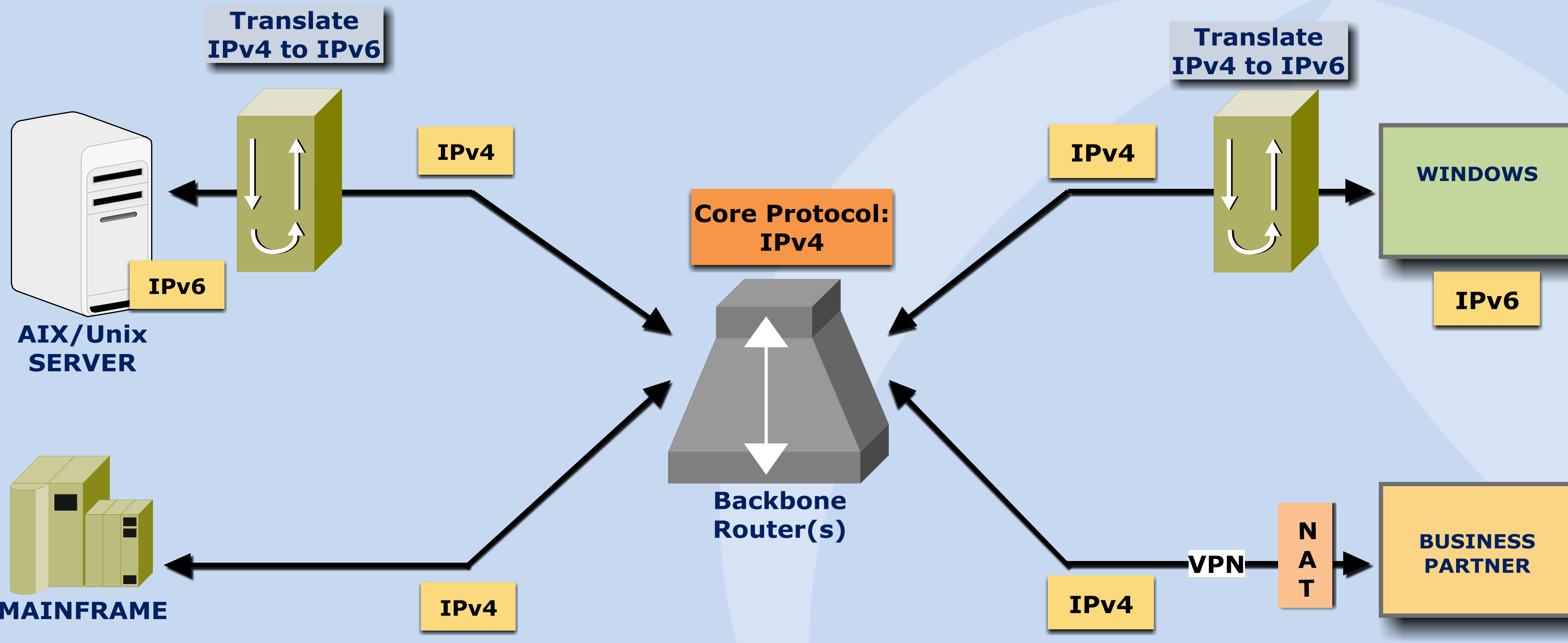
- A firewall may not be enforcing the same policy for IPv4 as for IPv6 traffic.
- Dual stack nodes within the network could be subject to different attacks than native IPv4.

Other Methods

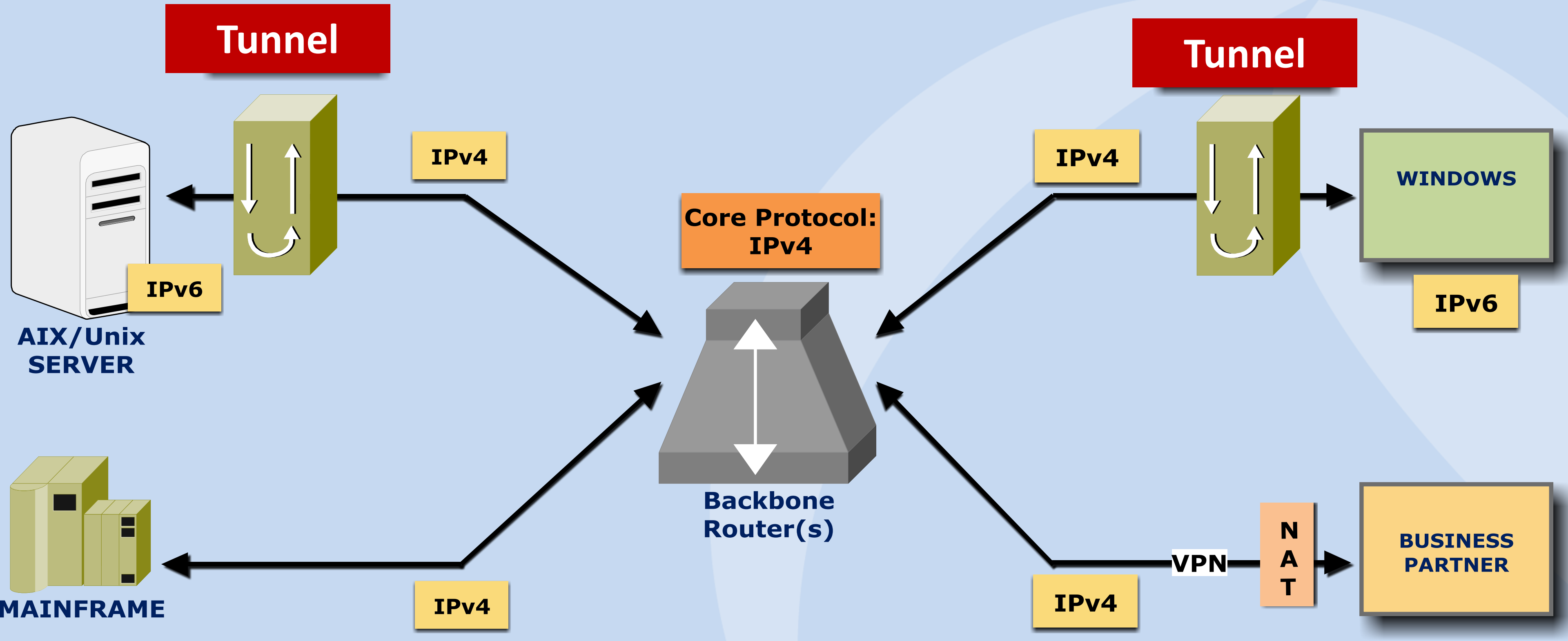
- Tunneling
  - Static
  - Manual
  - 6 to 4 tunnels
  - Teredo
  - Automatic tunnels (ISATAP)
  - GRE (with IPSec)
  - Tunnel broker
  - 6RD
  - Carrier Grade NAT (CGN)
- Translation
  - NAT64
  - DNS64
  - Network Address Translation with Protocol Translation (NAT-PT)
  - Transport Relay Translator (TRT)
  - Bump in the Stack (BIS)
  - Bump in the API (BIA)
  - NAT66



Where does translation happen?



Where does tunneling happen?



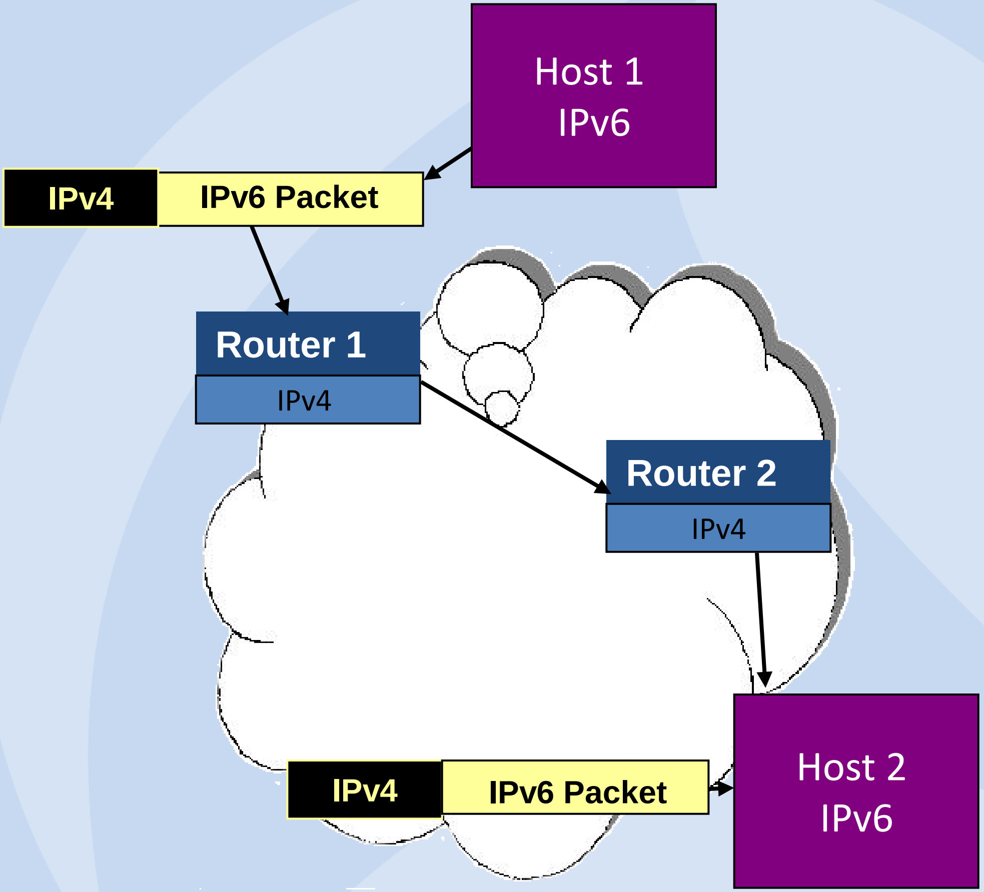
Results of bringing up IPv6 on Windows XP

Shortcut to cmd

```
C:\WINDOWS\system32>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.100
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        IP Address. . . . . . . . . . . . : fe80::211:d8ff:fe39:292b%4
        Default Gateway . . . . . . . . . : 192.168.1.1

Ethernet adapter Local Area Connection 2:

        Connection-specific DNS Suffix  . :
        Autoconfiguration IP Address. . . : 169.254.100.29
        Subnet Mask . . . . . . . . . . . : 255.255.0.0
        IP Address. . . . . . . . . . . . : 2001:5c0:8fff:fffe::3f53
        IP Address. . . . . . . . . . . . : fe80::2ff:8cff:fe10:3976%5
        Default Gateway . . . . . . . . . : 2001:5c0:8fff:fffe::3f52

Tunnel adapter Teredo Tunneling Pseudo-Interface:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : fe80::5445:5245:444f%6
        Default Gateway . . . . . . . . . :

Tunnel adapter Automatic Tunneling Pseudo-Interface:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : fe80::5efe:169.254.100.29%2
        Default Gateway . . . . . . . . . :

Tunnel adapter Automatic Tunneling Pseudo-Interface:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : fe80::5efe:192.168.1.100%2
        Default Gateway . . . . . . . . . :
```

Tunneling Overview

- IPv6 hosts
- IPv4 network
- IPv6 packet is encapsulated in an IPv4 datagram (may be header or header and upper layer protocol ex. UDP)
- At destination, packet is decapsulated. (fragments reassembled, etc)



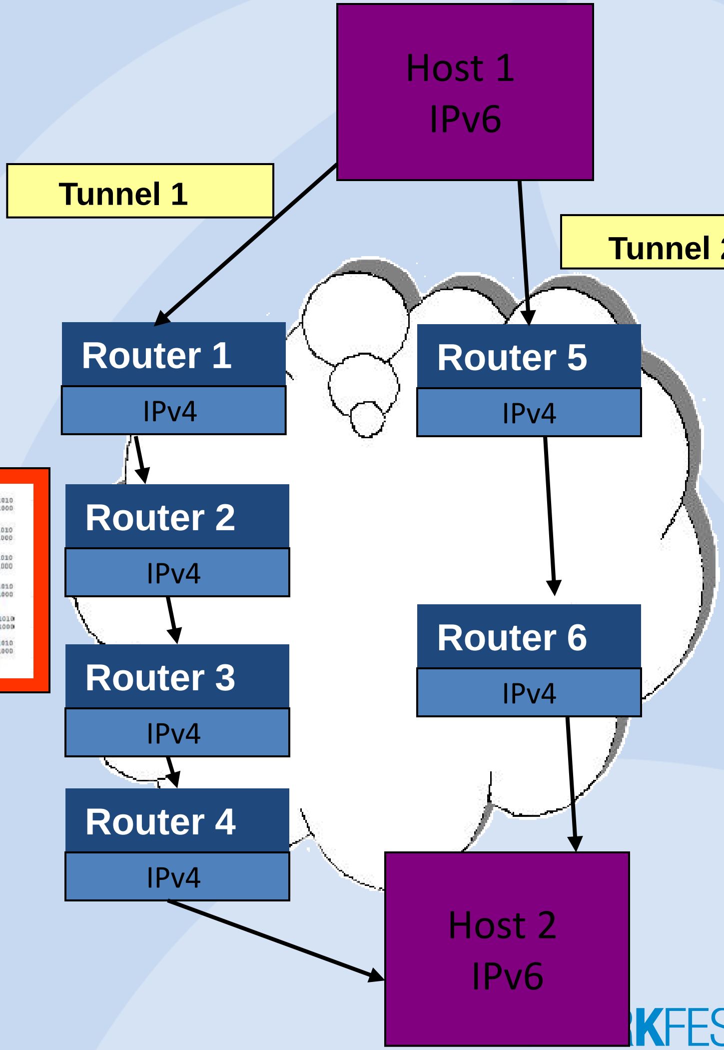
No.	Time	Source	Destination	Protocol	Info
154	7.773438	2001:638:902:1:201:2ff:fee2:7596	2002:5183:4383::5183:4383	TCP	ftp > 1026 [SYN, ACK] Se
198	9.736328	2001:638:902:1:201:2ff:fee2:7596	2002:5183:4383::5183:4383	FTP	Response: 220-
227	11.501953	2001:638:902:1:201:2ff:fee2:7596	2002:5183:4383::5183:4383	FTP	Response: 220 6bone.info
267	13.439453	2001:638:902:1:201:2ff:fee2:7596	2002:5183:4383::5183:4383	FTP	Response: 331 Guest logi
328	15.809571	2001:638:902:1:201:2ff:fee2:7596	2002:5183:4383::5183:4383	FTP	Response: 230 Guest logi
384	18.028321	2001:638:902:1:201:2ff:fee2:7596	2002:5183:4383::5183:4383	FTP	Response: 502 Unknown co
441	19.948243	2001:638:902:1:201:2ff:fee2:7596	2002:5183:4383::5183:4383	FTP	Response: 215 UNIX Type:
513	22.985352	2001:638:902:1:201:2ff:fee2:7596	2002:5183:4383::5183:4383	FTP	Response: 214-

```
⊞ Frame 154 (98 bytes on wire, 98 bytes captured)
⊞ Ethernet II, Src: 1a:43:20:00:01:00 (1a:43:20:00:01:00), Dst: 01:00:01:00:00:00 (01:00:01:00:00:00)
⊟ Internet Protocol, Src: 139.18.25.33 (139.18.25.33), Dst: 81.131.67.131 (81.131.67.131)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 84
    Identification: 0x29fb (10747)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 16
    Protocol: IPv6 (0x29)
  ⊞ Header checksum: 0x474d [correct]
    Source: 139.18.25.33 (139.18.25.33)
    Destination: 81.131.67.131 (81.131.67.131)
⊟ Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 24
    Next header: TCP (0x06)
    Hop limit: 63
    Source address: 2001:638:902:1:201:2ff:fee2:7596
    Destination address: 2002:5183:4383::5183:4383
⊟ Transmission Control Protocol, Src Port: ftp (21), Dst Port: 1026 (1026), Seq: 0, Ack: 1, Len: 0
    Source port: ftp (21)
    Destination port: 1026 (1026)
    Sequence number: 0    (relative sequence number)
    Acknowledgement number: 1    (relative ack number)
    Header length: 24 bytes
  ⊞ Flags: 0x0012 (SYN, ACK)
    Window size: 32768
    Checksum: 0x4194 [correct]
  ⊞ Options: (4 bytes)
  ⊞ [SEQ/ACK analysis]
```

IPv6 packet inside an IPv4 packet. Tunneling method is being used.

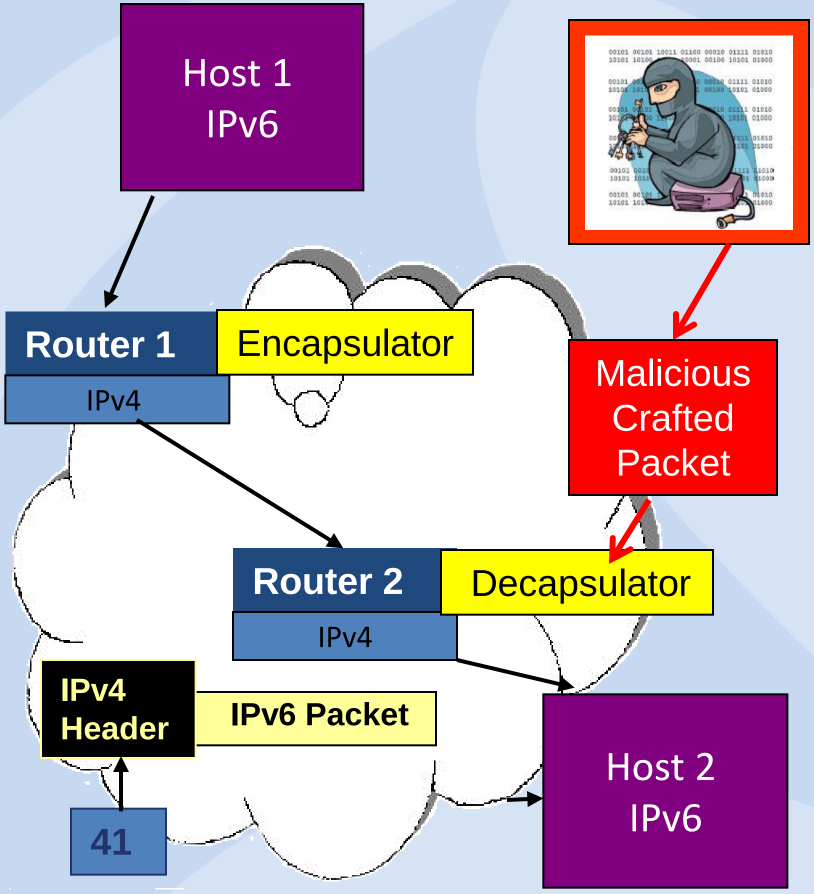
Security Issues Tunneling

- Shorter tunnels are preferable to longer ones.
- Shorter tunnels will have fewer hops or routers.
- Each router can be a potential attack point.



Packets Sent To Decapsulator

- A packet can be sent directly to the tunnel decapsulator.
- The tunnel decapsulators should make these checks:
  - IPv4 source address of the packet must be the same as configured for the tunnel end-point,
  - IPv4 and IPv6 packets are received from an expected interface,
  - IPv6 packets with several obviously invalid IPv6 source addresses received from the tunnel should be discarded.



Crafted Packet

```
⊞ Frame 9 (182 bytes on wire, 182 bytes captured)
⊞ Ethernet II, Src: 3com_03:04:05 (00:01:02:03:04:05),
⊟ Internet Protocol Version 6
    Version: 6
    Traffic class: 0x00
    Flowlabel: 0x00000
    Payload length: 43008
    Next header: IPv6 fragment (0x2c)
    Hop limit: 255
    Source address: ::
    Destination address: ::
⊟ Fragmentation Header
    Next header: IPv6 routing (0x2b)
    Offset: 48
    More fragments: Yes
    Identification: 0x00370037
⊟ Routing Header, Type 0
    Next header: IPv6 fragment (0x2c)
    Length: 9 (80 bytes)
    Type: 0
    Segments left: 0
    address 0: ::
    address 1: ::
    address 2: ::
    address 3: ::
    address 4: ::7005:917c:ffff:ffff
⊟ Fragmentation Header
    Next header: IPv6 hop-by-hop option (0x00)
    Offset: 0
    More fragments: No
    Identification: 0x00000000
⊟ Hop-by-hop Option Header
```

- Here is an IPv6 packet which I crafted with multiple routing and fragmentation headers.
- Such a packet should easily be sent to the tunnel decapsulator address.
- All that is needed is the IP address of the tunnel endpoint.

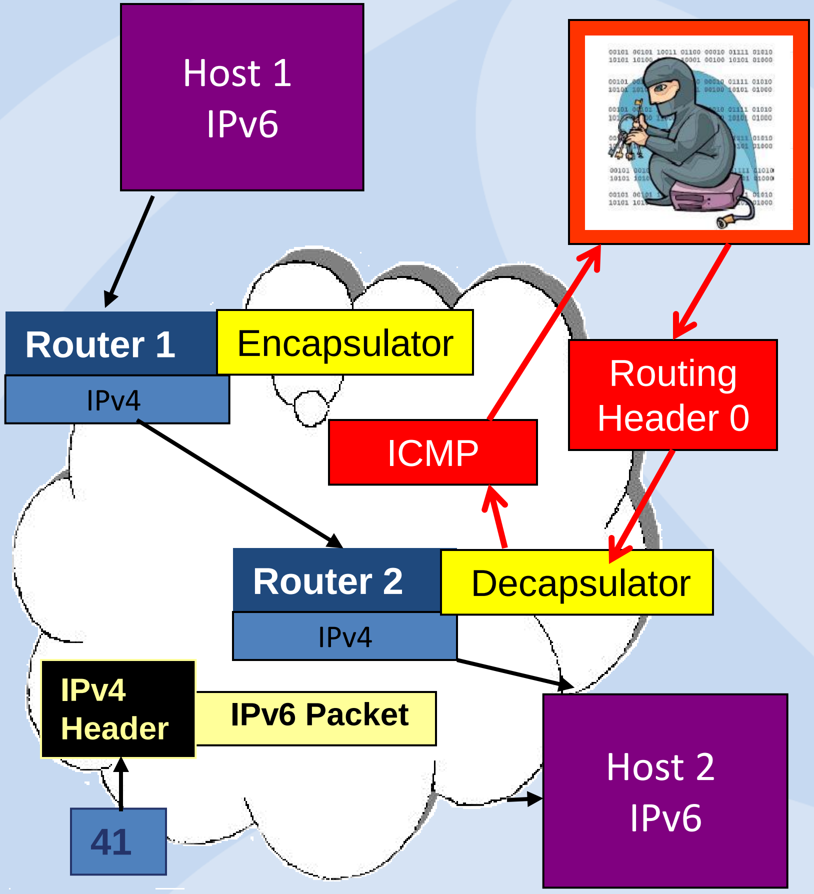
RFC5095 (Deprecation of Type 0 Routing Headers in IPv6)

- An IPv6 node that receives a packet with a destination address assigned to it and that contains an RH0 extension header MUST NOT execute the algorithm specified in the latter part of Section 4.4 of [RFC2460] for RH0. Instead, such packets MUST be processed according to the behaviour specified in Section 4.4 of [RFC2460] for a datagram that includes an unrecognised Routing Type value, namely:

- If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet, whose type is identified by the Next Header field in the Routing header.

- If Segments Left is non-zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0, message to the packet‘s Source Address, pointing to the unrecognized Routing Type.

- IPv6 implementations are no longer required to implement RH0 in any way.

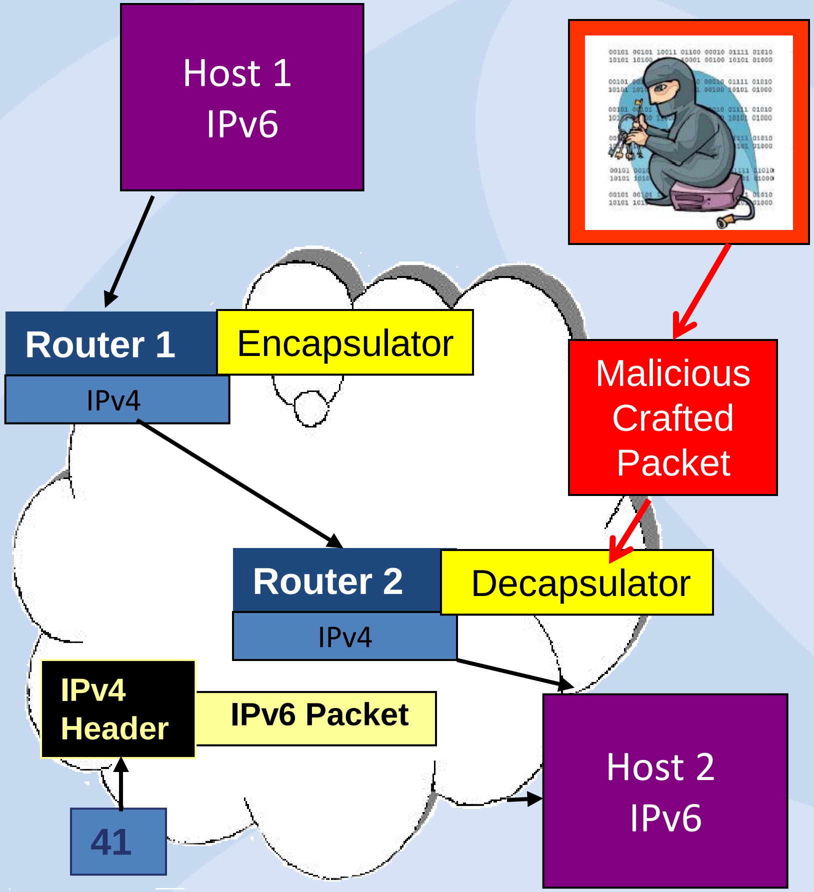


General Tunneling Threats

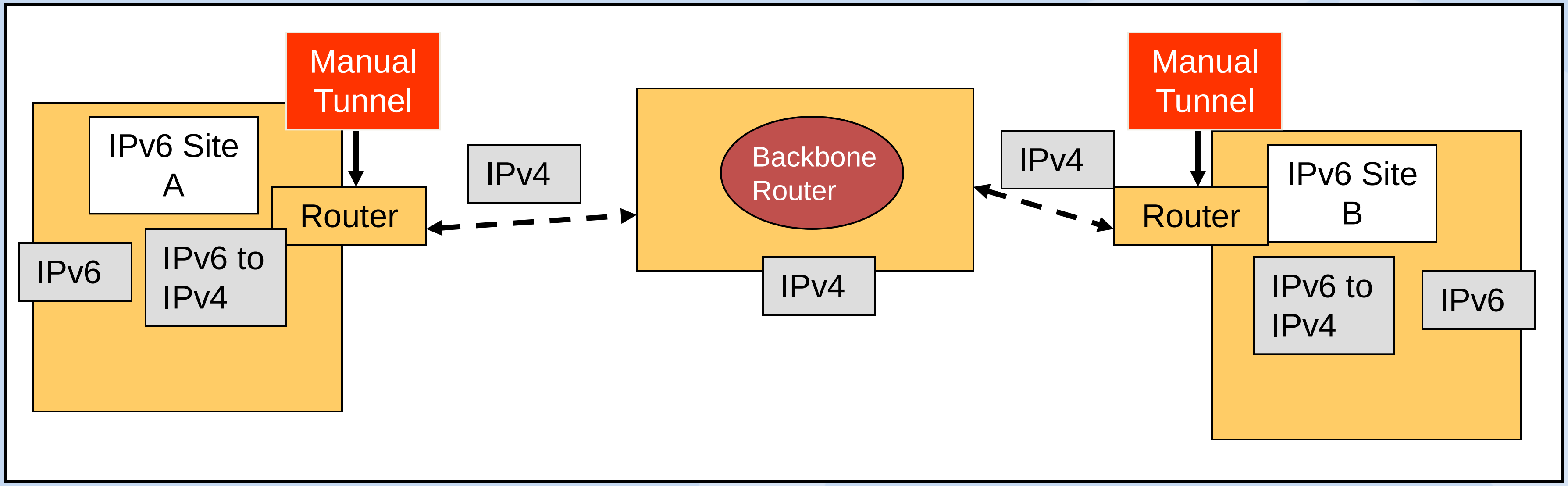
- The firewall does not know how to inspect packets for tunnels. Malicious IPv6 packets get through.
- The IPv6 addresses inside the packet is not subject to filtering by the firewall. So, malicious packets can be sent anywhere.
- The embedded IPv6 packet can contain a routing header which can create routing loops or network congestion. These packets may not be filtered at the routers.
- The embedded IPv6 packet can lead to a node pretending to be a router which then injects malicious packets into the network.
- Embedded IPv6 packets with malicious intent can also be sent directly to the tunnel end-point (decapsulator)

Tunneling Best Practices

- A tunneling scheme with authentication should be used. For example, Generic Routing Encapsulation (GRE) with IPSec.
- When dropping packets, the node should do this silently. That is, it should not send a message, such as an ICMP error because this could be used to probe the acceptable tunnel endpoint address or to create a denial of service reflector attack by generating many ICMP messages.



Manual Tunnels



- With manually configured IPv6 tunnels, an IPv6 address is configured on a tunnel interface.
- Manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination.
- The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.
- Notice that in the above scenario, the conversion is done at the boundary and the backbone routes remain IPv4.

Sample Manual Tunnel Configuration

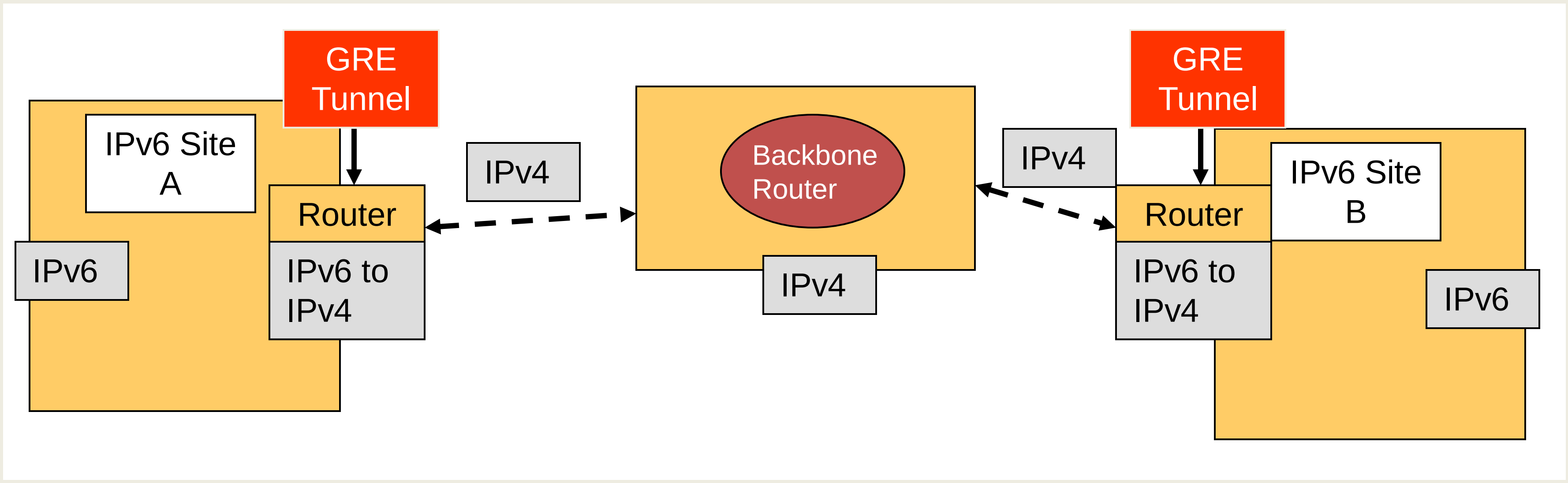
Router A Configuration

```
interface ethernet 0
ip address 192.168.99.1 255.255.255.0

interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::3/127
 tunnel source ethernet 0
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

Router B Configuration

```
interface ethernet 0
ip address 192.168.30.1 255.255.255.0

interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::2/127
 tunnel source ethernet 0
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

GRE Tunnels



- GRE stands for Generic Route Encapsulation
- With GRE IPv6 tunnels, an IPv6 address is configured on a tunnel interface.
- IPv4 addresses are assigned to the tunnel source and the tunnel destination.
- The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.
- Notice that in the above scenario, the conversion is done at the boundary and the backbone routes remain IPv4.

GRE Tunnels

IPv4 in IPv6

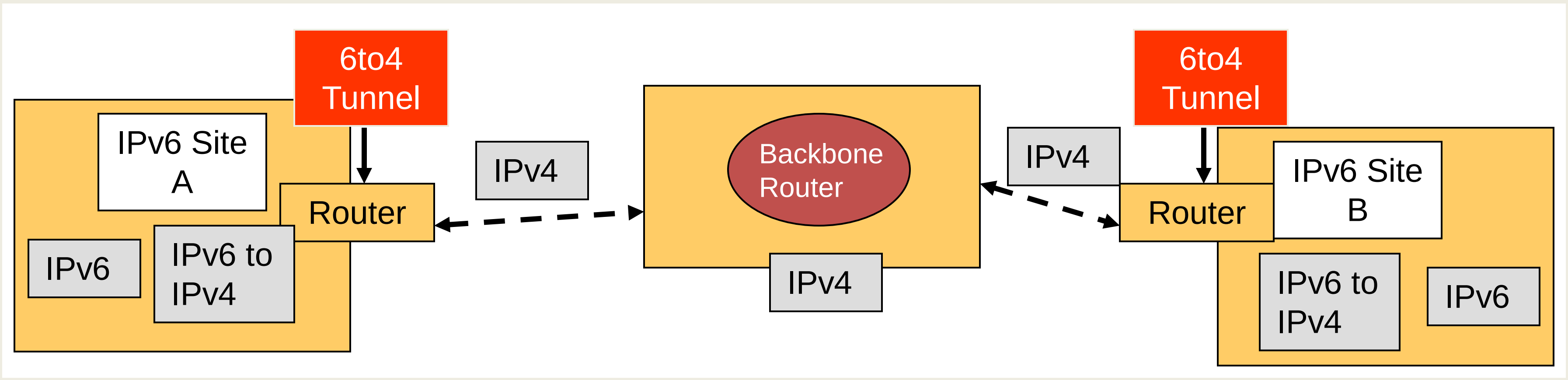
IPv6 in IPv4

- GRE tunnels can run over an IPv6 network or IPv4 network.
- You may also do GRE with IPSec.
- It may be that GRE tunnels are what one might want to do in both IPv6 and IPv4 situations. If GRE tunnels are the policy, then no matter what the underlying network, this is the tunneling mechanism.

Security Issues Manual / GRE Tunnels

- Manual and GRE tunnels are subject to the general threats discussed previously. However, if GRE with IPSec is used, then it is likely impossible to connect to the decapsulator.
- The only additional threat is if tunnels are misconfigured, then traffic may end up in the wrong user's hands. Because there may be many manual tunnels, it is possible that one might be misconfigured.

6to4 Tunnels



- **6to4** allows IPv6 packets to be transmitted over an IPv4 network.
- It is described in RFC 3056: Connection of IPv6 Domains via IPv4 Clouds.
- 6to4 is a router to router tunneling mechanism.
- The tunnel is configured dynamically.
- 6to4 is intended only as transition mechanism and is not meant to be used permanently.

6to4 Tunneling Interface

```
Interface 3: 6to4 Tunneling Pseudo-Interface
  Guid {A995346E-9F3E-2EDB-47D1-9CC7BA01CD73}
  does not use Neighbor Discovery
  does not use Router Discovery
  routing preference 1
  link MTU 1280 (true link MTU 65515)
  current hop limit 128
  reachable time 26000ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
  default site prefix length 48
```

- 6to4 interface is automatically created if supported by router
- RFC 3056:Connection of IPv6 Domains via IPv4 Clouds

Operational Differences

- There are operational differences for 6to4 tunnels in various platforms.
- The z/OS Communications Server mainframe at the current time (z/OS v. 1.13) cannot be a tunnel endpoint.
- The 6to4 interface is automatically created in Windows XP and above.
- Most, if not all, Unix implementations support 6to4.
- Cisco routers support 6to4 tunnels

Sample 6to4 Configuration

```
interface Ethernet0
description IPv4 uplink
ip address 192.168.99.1
255.255.255.0

interface Ethernet1
description IPv6 local network 1
ipv6 address 2002:c0a8:6301:1::1/64

interface Ethernet2
description IPv6 local network 2
ipv6 address 2002:c0a8:6301:2::1/64

interface Tunnel0
description IPv6 uplink
no ip address
ipv6 address 2002:c0a8:6301::1/64
tunnel source Ethernet 0
tunnel mode ipv6ip 6to4

ipv6 route 2002::/16 tunnel 0
```

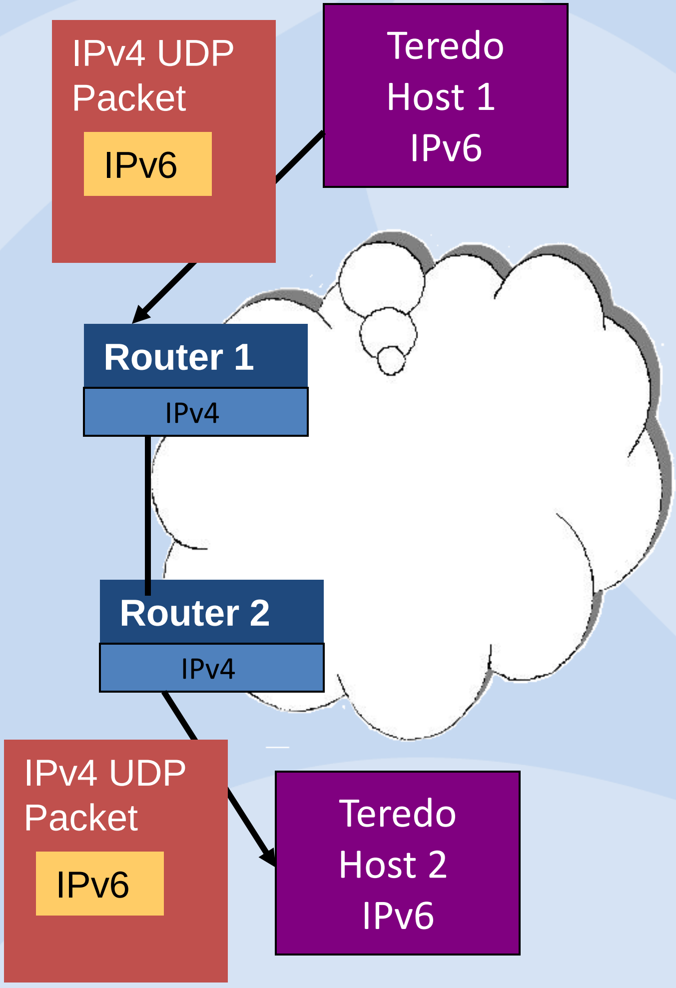
- For example, within the Cisco IOS, only the tunnel source address is given.
- The tunnel destination is determined by the IPv4 address of the border router extracted from the IPv6 address that starts with the prefix 2002::/16.
- The format is 2002:border-router-IPv4-address::/48.

6to4 Specific Security Issues

- The 6to4 mechanism introduces more security considerations:
  - All 6to4 routers must accept and decapsulate IPv4 packets from every other 6to4 router, and from 6to4 relays.
  - 6to4 relay routers must accept traffic from any native IPv6 node.
- Thus, addresses within the IPv4 and IPv6 headers may be spoofed, and this leads to various types of threats, including different flavors of Denial of Service attacks.
- The 6to4 specification outlined a few security considerations and rules but was ambiguous as to their exact requirement level. Moreover, the description of the considerations was rather short, and some of them have proven difficult to understand or impossible to implement.

Why Teredo?

- To use 6to4 – you need a 6to4 router.
- Teredo can work without such a router.
- 6to4 also may not work with NATs. NATs may not do Protocol 41 translation.
- Teredo encapsulates the IPv6 packet as an IPv4 UDP message, containing both an IPv4 and UDP header.
- UDP messages can be translated universally by NATs and can traverse multiple layers of NATs.



Teredo Tunneling

```
Interface 4: Teredo Tunneling Pseudo-Interface
  Guid {F0DB2AF0-D5CB-4E00-B5A3-8550D208BDF9}
  zones: link 4 site 2
  cable unplugged
  uses Neighbor Discovery
  uses Router Discovery
  routing preference 2
  link-layer address: 0.0.0.0:0
    preferred link-local fe80::5445:5245:444f, life infinite
    multicast interface-local ff01::1, 1 refs, not reportabl
    multicast link-local ff02::1, 1 refs, not reportable
  link MTU 1280 (true link MTU 1280)
  current hop limit 128
  reachable time 19000ms (base 30000ms)
  retransmission interval 1000ms
  DAD transmits 0
  default site prefix length 48
```

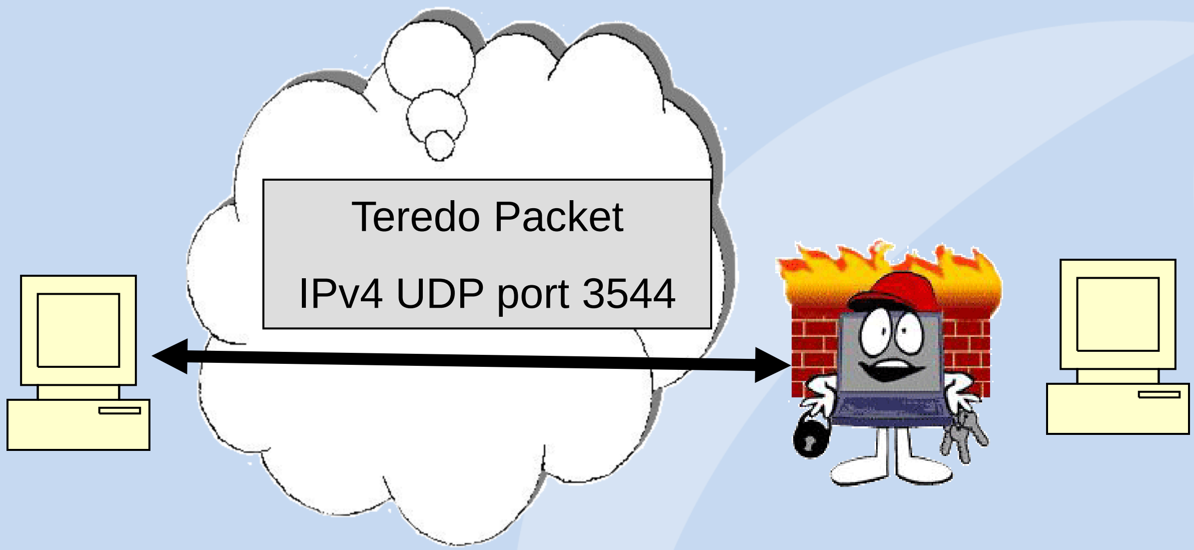
- Teredo interface is automatically created
- RFC 4380:Teredo: Tunneling IPv6 over UDP

Teredo Specific Threats

- The IPv4 address and port is contained in the client's Teredo address. It is 'obfuscated' but since the obfuscation algorithm is clearly spelled out in the RFC, the obfuscation can be easily reversed by a novice programmer.
- The following is from RFC4380 : Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs):

```
The Teredo addresses are composed of 5 components:

+-------------+-------------+-------+------+-------------+
| Prefix      | Server IPv4 | Flags | Port | Client IPv4 |
+-------------+-------------+-------+------+-------------+
   - Prefix: the 32-bit Teredo service prefix.
   - Server IPv4: the IPv4 address of a Teredo server.
   - Flags: a set of 16 bits that document type of address and NAT.
   - Port: the obfuscated "mapped UDP port" of the client Teredo service.
   - Client IPv4: the obfuscated "mapped IPv4 address" of the client.

In this format, both the "mapped UDP port" and "mapped IPv4 address" of the
client are obfuscated.  Each bit in the address and port number is reversed;
this can be done by an exclusive OR of the 16-bit port number with the
hexadecimal value 0xFFFF, and an exclusive OR of the 32-bit address with the
hexadecimal value 0xFFFFFFFF.
```

Block Teredo Port



- Teredo embeds IPv6 packets in UDP.   Firewalls may not find and inspect Teredo traffic.

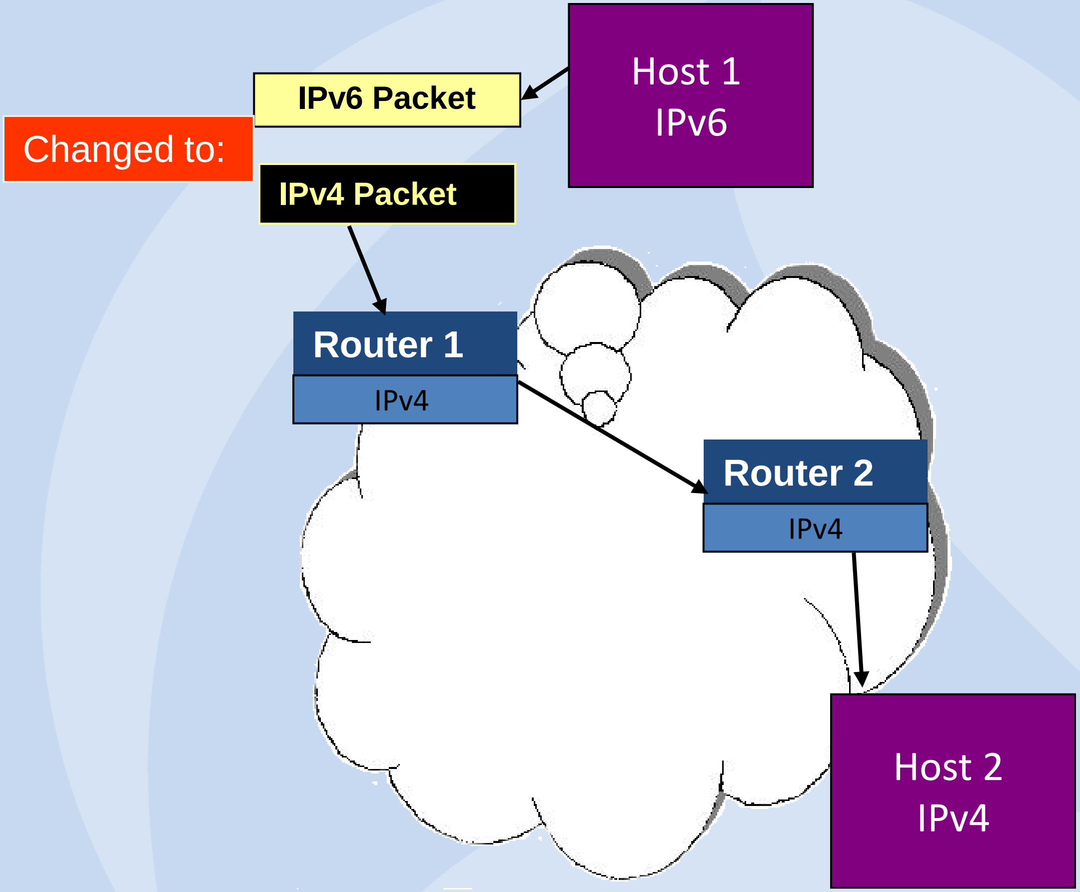
- One simple method to deal with Teredo is to block the port used: UDP port 3544. This prevents a Teredo client from connecting to its server.

Translation Overview

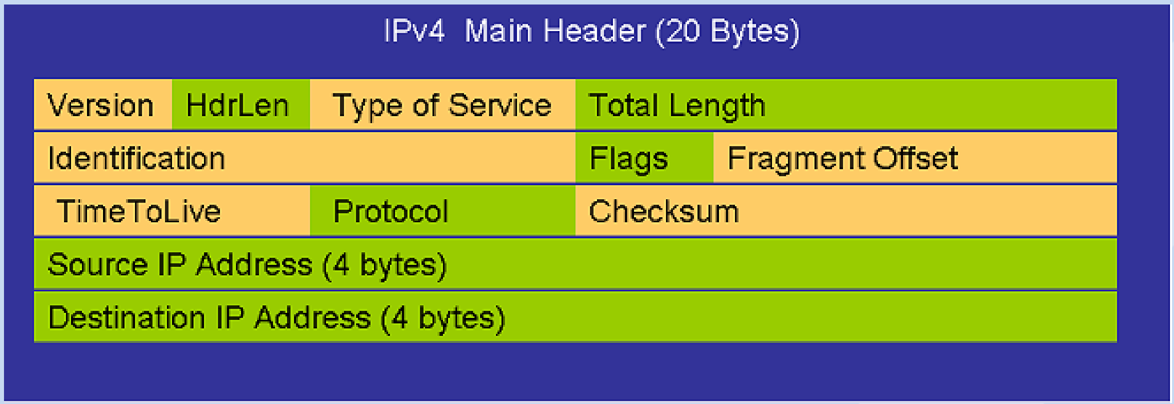
- Translation will change IPv6 packets to IPv4 or vice versa. IPv6-only hosts may need to communicate with IPv4-only hosts.

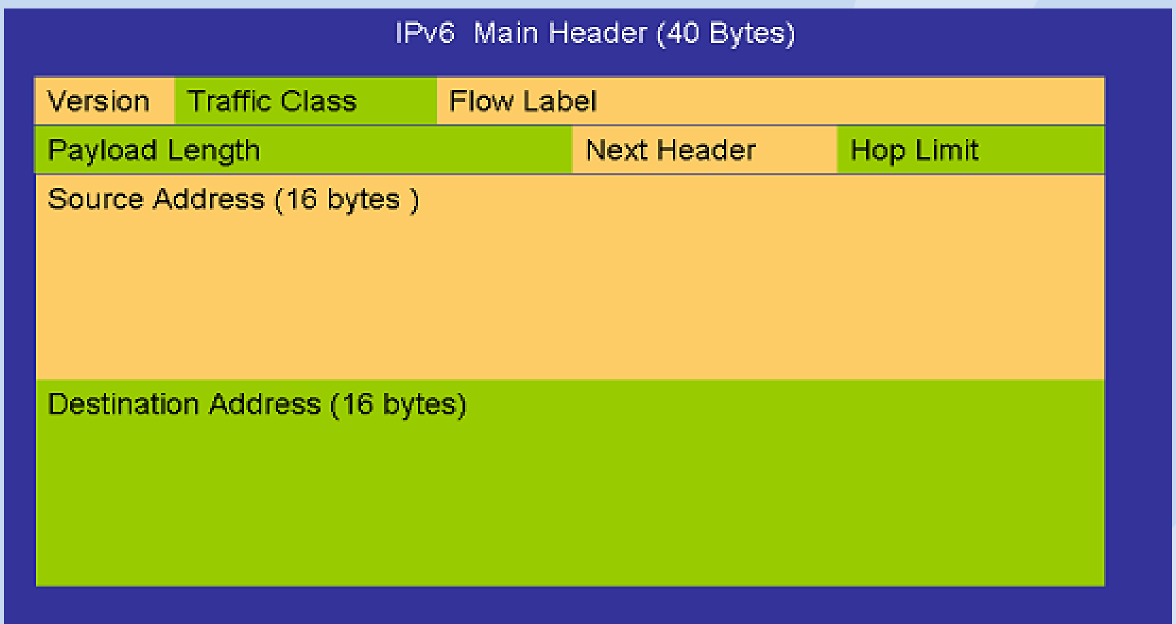
- What is needed is:
  - Convert IPv4 header to IPv6 header (or vice versa)
  - Get a common address
  - Provide routing

- Methods
  - NAT64 – DNS64
  - Network Address Translation with Protocol Translation (NAT-PT)
  - Transport Relay Translator (TRT)
  - Bump in the Stack (BIS)
  - Bump in the API (BIA)



SIIT (Header Rewriting)





- SIIT is described in RFC 2765 : Stateless IP/ICMP Translation Algorithm.
- SIIT allows you to take an IPv4 packet and rewrite the headers to form an IPv6 packet or vice versa.

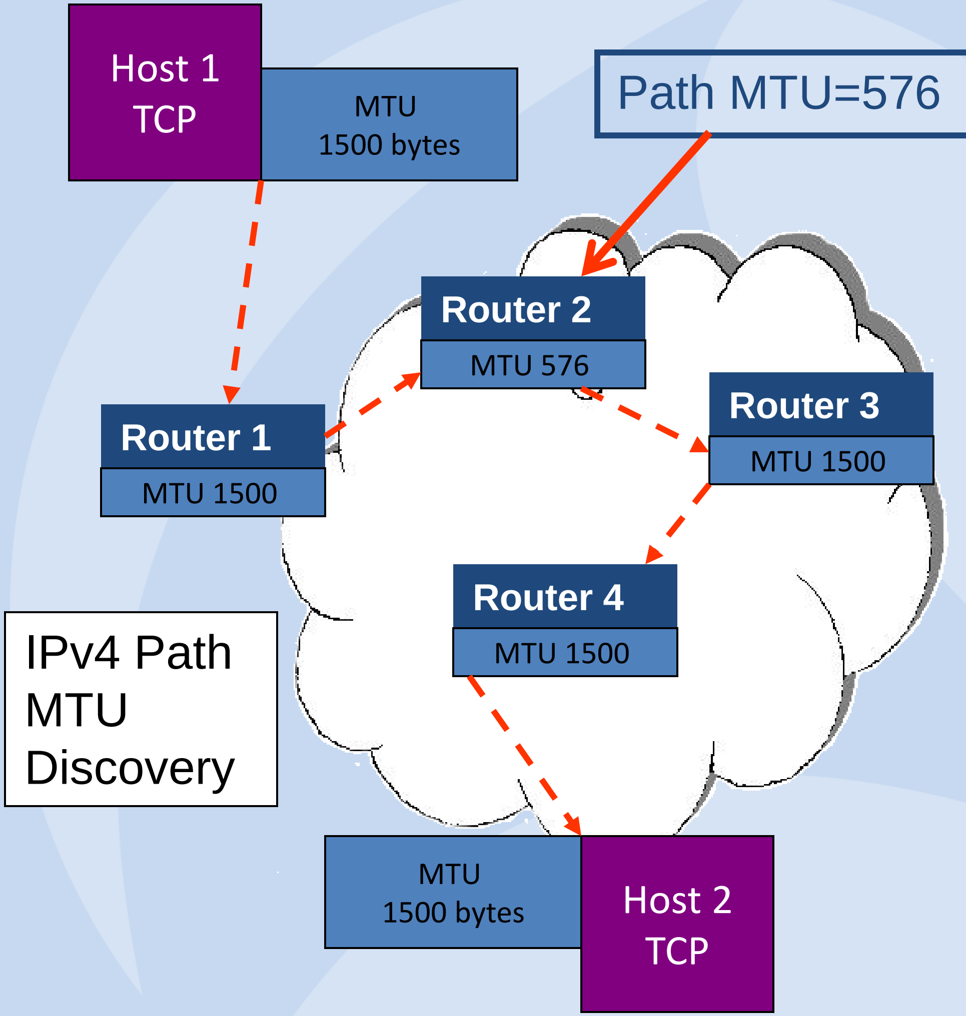
Rewriting Issues

- Rewriting headers is relatively straightforward for IP, TCP and UDP headers.
- ICMPv6 to ICMPv4 is more challenging because ICMPv6 has many functions which have no counterpart in ICMPv4. SIIT specifies the techniques for doing ICMPv6 / ICMP4 translation.
- SIIT also needs to work with a method such as NAT64 or NAT-PT to translate the addresses and then tunneling or other method for routing.

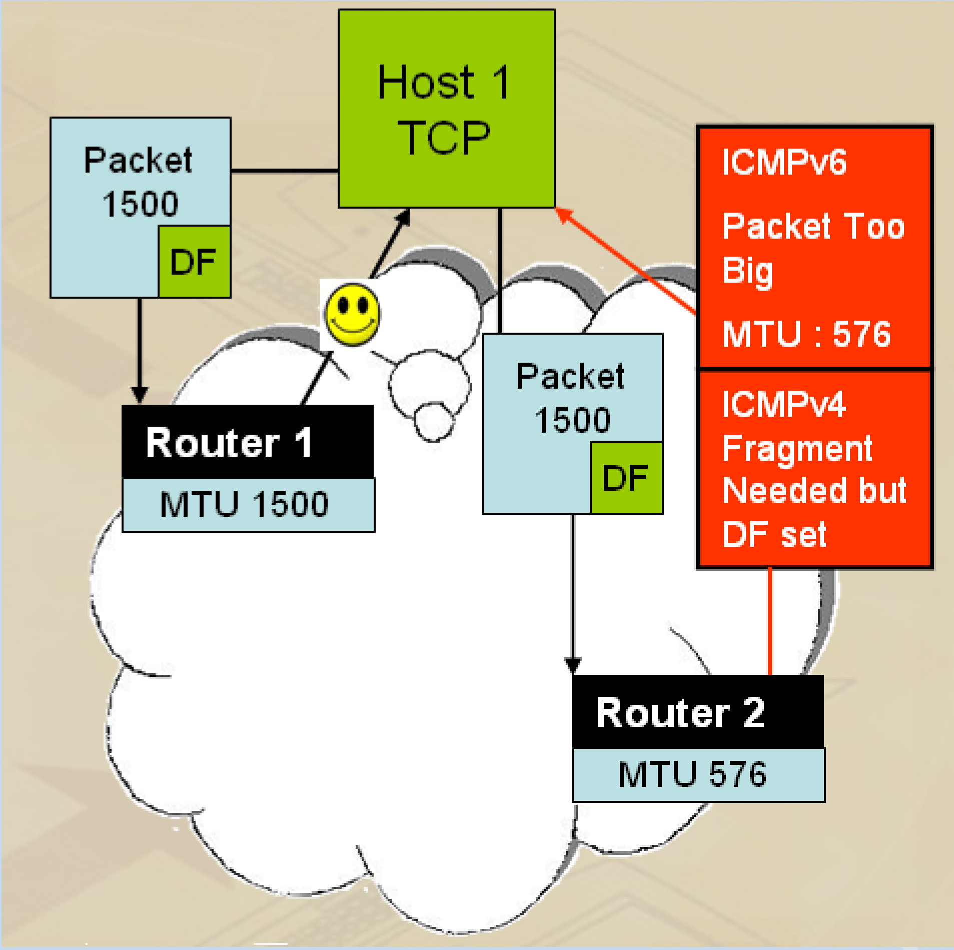
```
              ICMPv6 Messages

Type   Name
----   ---------------------------
 128   Echo Request
 129   Echo Reply
 130   Multicast Listener Query
 131   Multicast Listener Report
 132   Multicast Listener Done
 133   Router Solicitation
 134   Router Advertisement
 135   Neighbor Solicitation
 136   Neighbor Advertisement
 137   Redirect Message
 138   Router Renumbering
 139   ICMP Node Info. Query
 140   ICMP Node Info. Response
 141   Inverse Neighbor Discovery
       Solicitation Message
```

SIIT and Path MTU Discovery

- An issue for header rewriting is dealing with packet fragmentation and path MTU.
- One of the differences between IPv4 and IPv6 is that in IPv6 path MTU discovery is mandatory but it is **optional** in IPv4.
- This is because in IPv6 routers will never fragment a packet - only the sender can do fragmentation



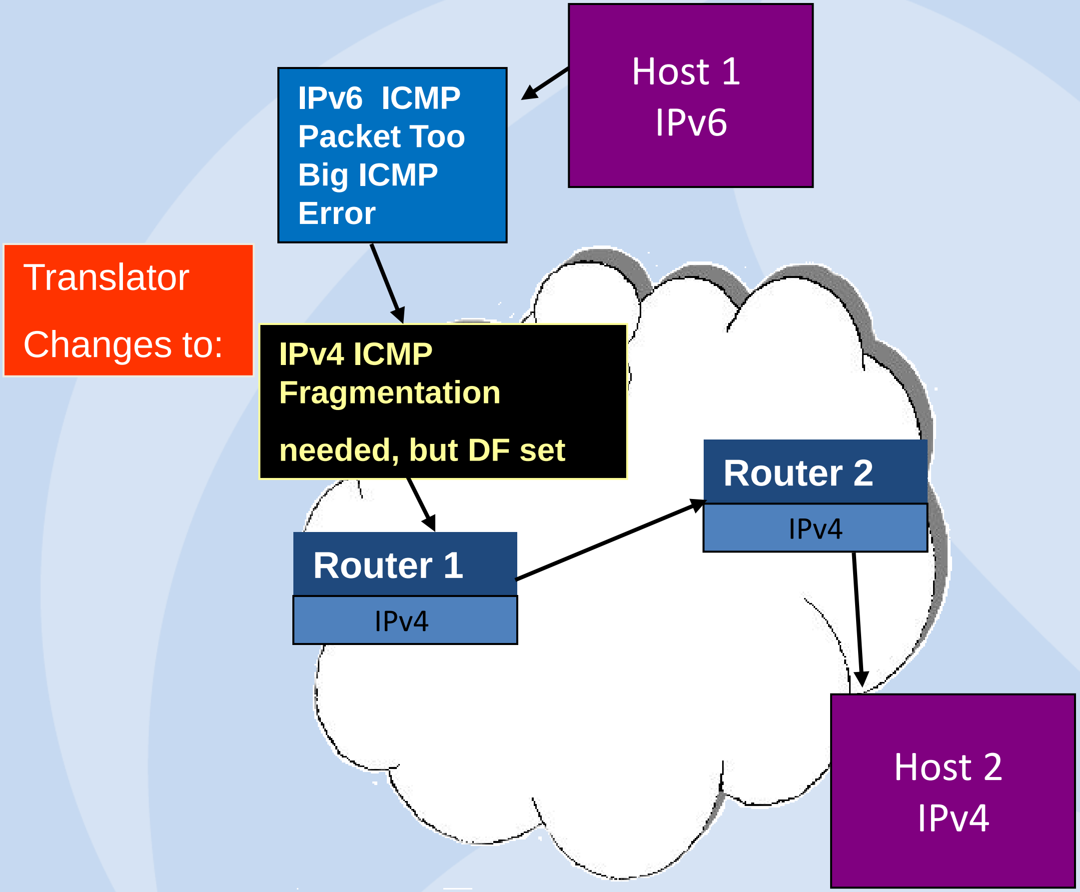
So, what's the problem?



- When IP4 does Path MTU discovery – there is no problem!
- The IPv4 node performs path MTU discovery by setting the DF bit in the header, the path MTU discovery can operate end-to-end i.e. across the translator.
- In this case either IPv4 or IPv6 routers might send back ICMP error messages to the sender.
- IPv6 will send an ICMPv6 Packet Too Big message; IPv4 will send an ICMP Destination Unreachable, Fragmentation Needed but Do Not Fragment Set to the sender.

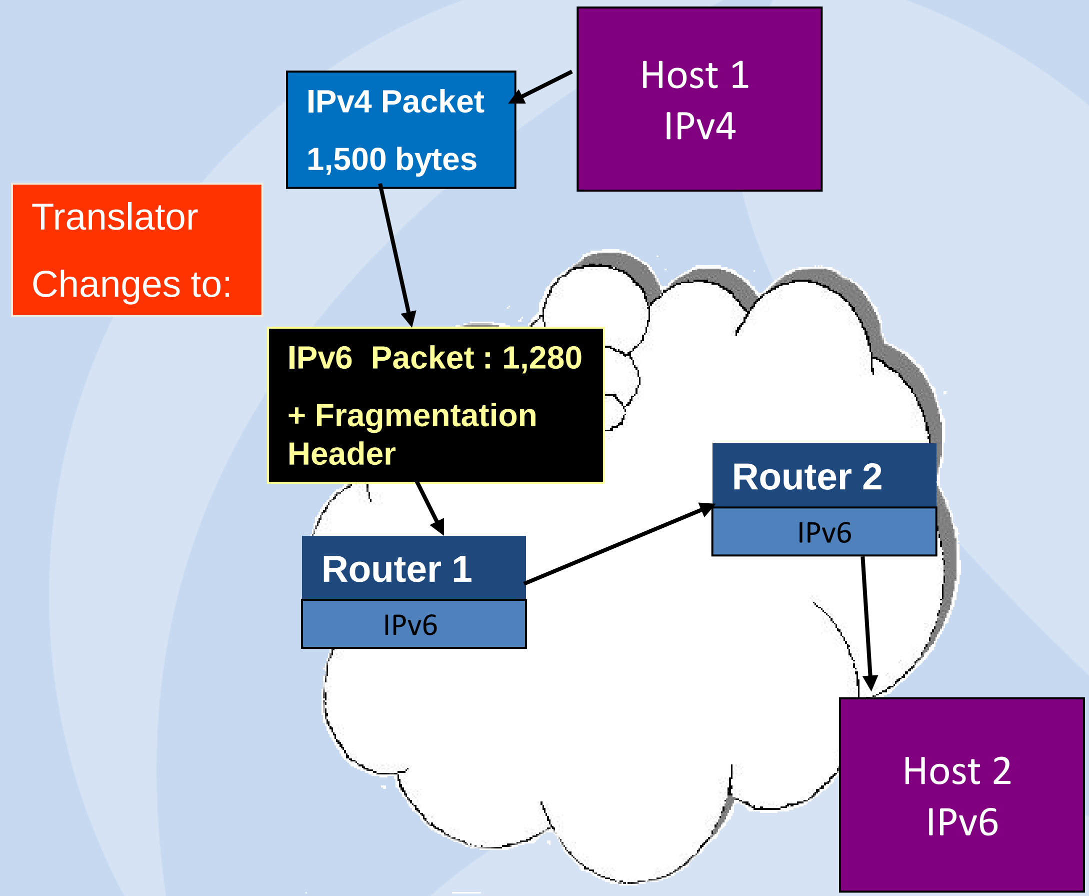
SIIT – ICMP Translation

- When the ICMP errors are sent by the IPv6 routers they will pass through a translator which will translate the ICMP error to a form that the IPv4 sender can understand.
- ICMPv4 Fragmentation Needed but Do Not Fragment set is changed to ICMPv6 Packet Too Big error.



SIIT – No IPv4 Path MTU

- However, when the IPv4 sender does not perform path MTU discovery, the translator has to ensure that the packet does not exceed the path MTU on the IPv6 side.
- This is done by fragmenting the IPv4 packet so that it fits in the minimum MTU of 1280 bytes for an IPv6 packet and adding an IPv6 fragmentation extension header.
- According to the RFC, if PMTU is not done, the translator MUST always include an IPv6 fragment header to indicate that the sender allows fragmentation.



SIIT Drawbacks

- **No IPv4 Options :** The translation function for SIIT does not translate any IPv4 options. (Not often used. Partial list on next page.)
- **Not All IPv6 Extension Headers Supported :** IPv6 routing headers, hop-by-hop extension headers, and destination options headers are not translated.
- **Best Effort Translation:** Translation can only be done on a best effort approach due to the significant differences between the IPv4 and IPv6 headers.
- **No IPv4 Multicast:** IPv4 multicast addresses can not be mapped to IPv6 multicast addresses. For instance, ::ffff:224.1.2.3 is an IPv4 mapped IPv6 address with a class D address, however it is not an IPv6 multicast address. While the IP/ICMP header translation aspect of SIIT in theory works for multicast packets, the address mapping limitation makes it impossible to apply the techniques for multicast traffic.

IP Options

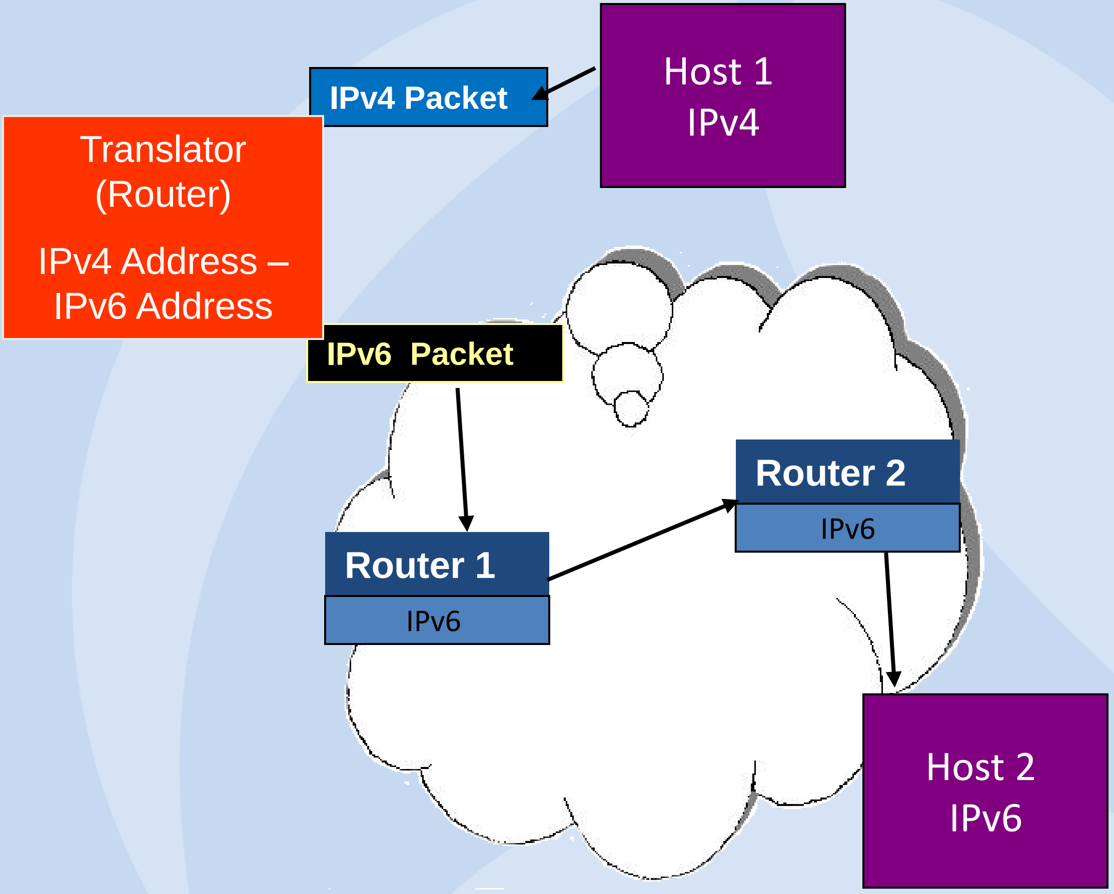
Copy	Class	Number	Value	Name	Reference
0	0	0	0	EOOL - End of Options List	[RFC791,JBP]
0	0	1	1	NOP - No Operation	[RFC791,JBP]
1	0	2	130	SEC - Security	[RFC1108]
1	0	3	131	LSR - Loose Source Route	[RFC791,JBP]
0	2	4	68	TS - Time Stamp	[RFC791,JBP]
1	0	5	133	E-SEC - Extended Security	[RFC1108]
1	0	6	134	CIPSO - Commercial Security	[draft-ietf-cipso-ipsecurity-01]
0	0	7	7	RR - Record Route	[RFC791,JBP]
1	0	8	136	SID - Stream ID	[RFC791,JBP]
1	0	9	137	SSR - Strict Source Route	[RFC791,JBP]
0	0	10	10	ZSU - Experimental Measurement	[ZSu]
0	0	11	11	MTUP - MTU Probe	[RFC1191]*
0	0	12	12	MTUR - MTU Reply	[RFC1191]*
1	2	13	205	FINN - Experimental Flow Control	[Finn]
1	0	14	142	VISA - Expermental Access Control	[Estrin]
0	0	15	15	ENCODE - ???	[VerSteeg]
1	0	16	144	IMITD - IMI Traffic Descriptor	[Lee]
1	0	17	145	EIP - Extended Internet Protocol	[RFC1385]
0	2	18	82	TR - Traceroute	[RFC1393]
1	0	19	147	ADDEXT - Address Extension	[Ullmann IPv7]
1	0	20	148	RTRALT - Router Alert	[RFC2113]
1	0	21	149	SDB - Selective Directed Broadcast	[Graff]
1	0	22	150	- Unassigned (Released 18 October 2005)	
1	0	23	151	DPS - Dynamic Packet State	[Malis]

Security Issues Translation

- In general, the issues with translation are:
  - Single point of failure
  - Man-in-the-Middle
  - No IPsec
  - No DNS-SEC
  - Address Depletion Denial of Service Attack
  - Resource Depletion Denial of Service Attack
  - Bypass firewall filters

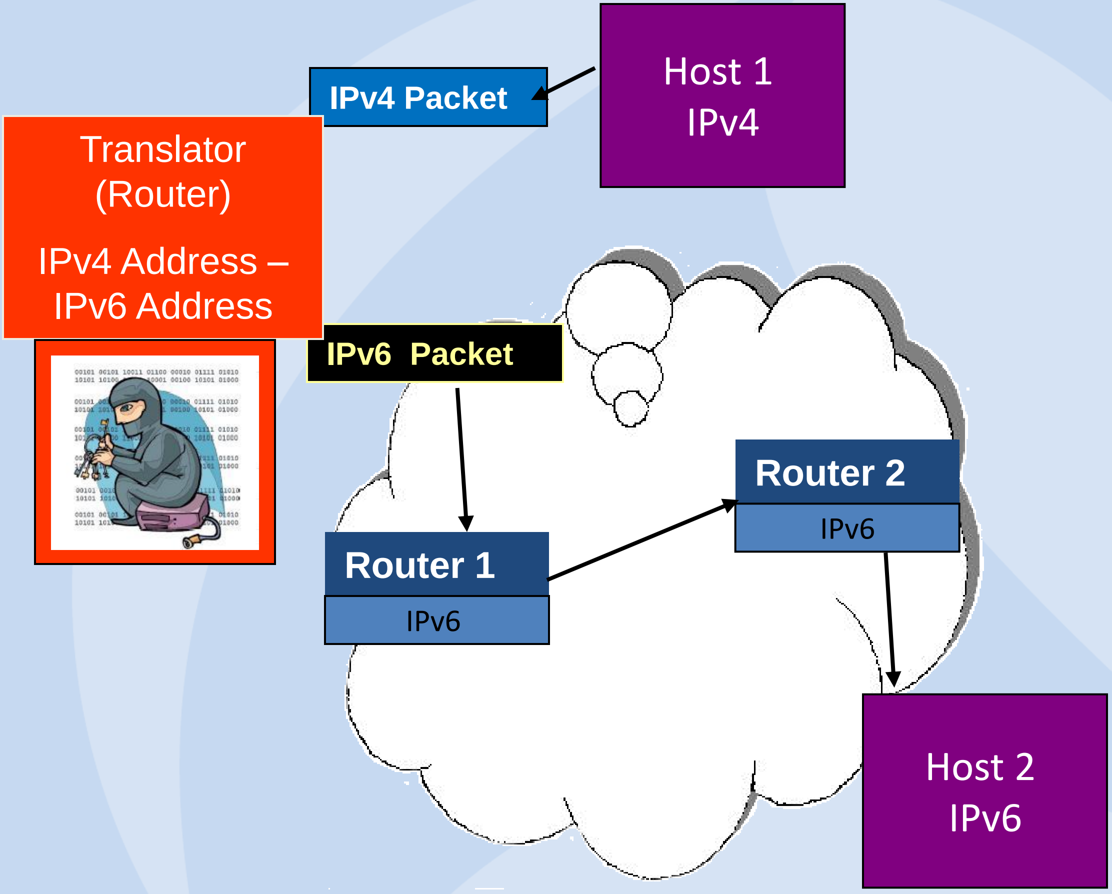
Single Point of Failure

- When doing translation, the packet must flow in and out of the same translation device (generally a router) because the device is keeping track of the session.
- This creates a single point of failure.



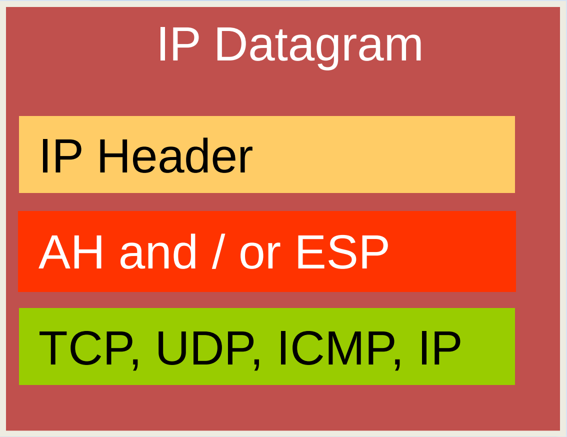
Man-in-the-Middle

If a fixed prefix is used for the translation router or if an attacker were somehow able to give the IPv6 node a fake prefix, the attacker would be able to steal all of the node's outbound packets or snoop the inbound packets.



IPSec

- **No End-To-End AH Protocol (IPsec) :** It is not possible to use end-to-end AH through the translator.
- **ESP Tunnel Mode Difficulties (IPsec) :** It is difficult to use ESP in tunnel mode through the translator.



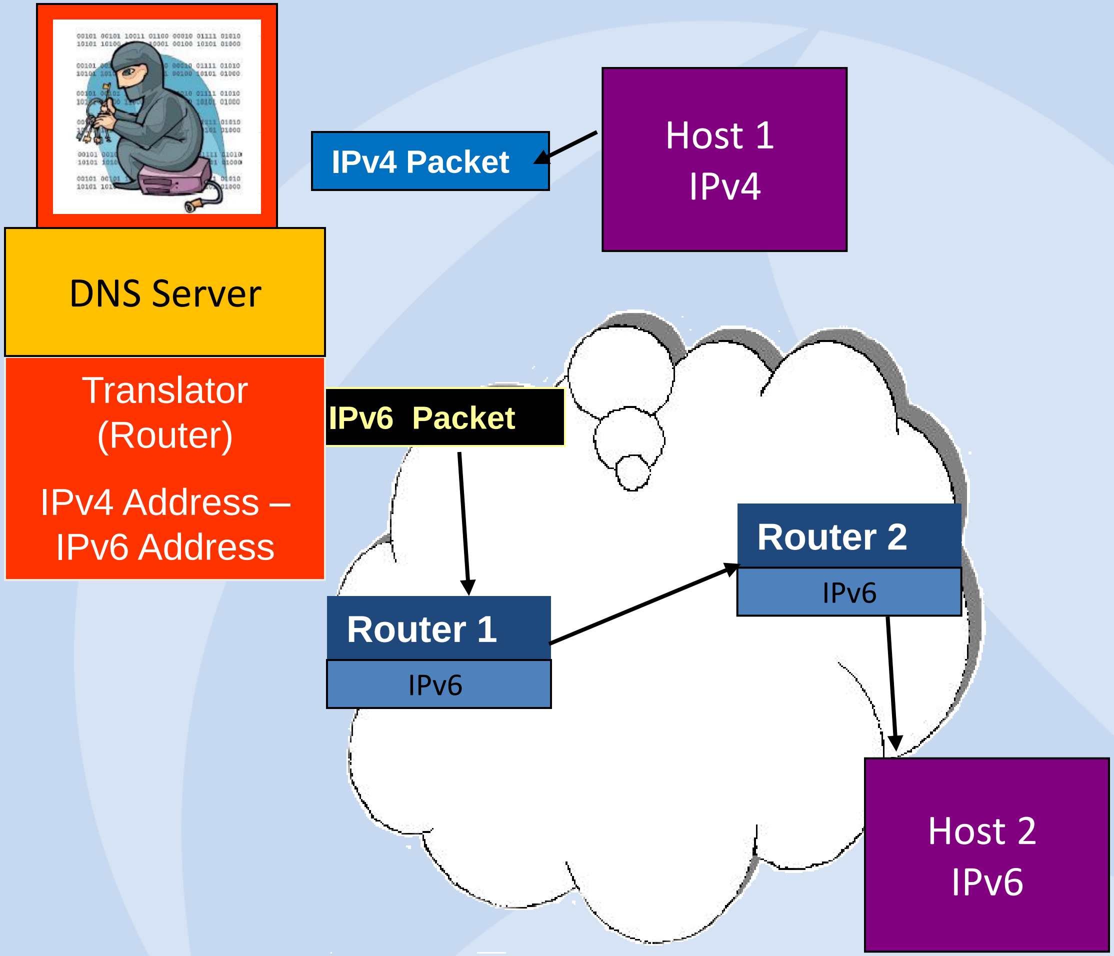
AH = integrity

ESP = integrity and confidentiality

No DNS-SEC

- DNS-SEC generally will not work with translation.

- This means that it is possible for an attacker to modify records from DNS to the IPv4 nodes.



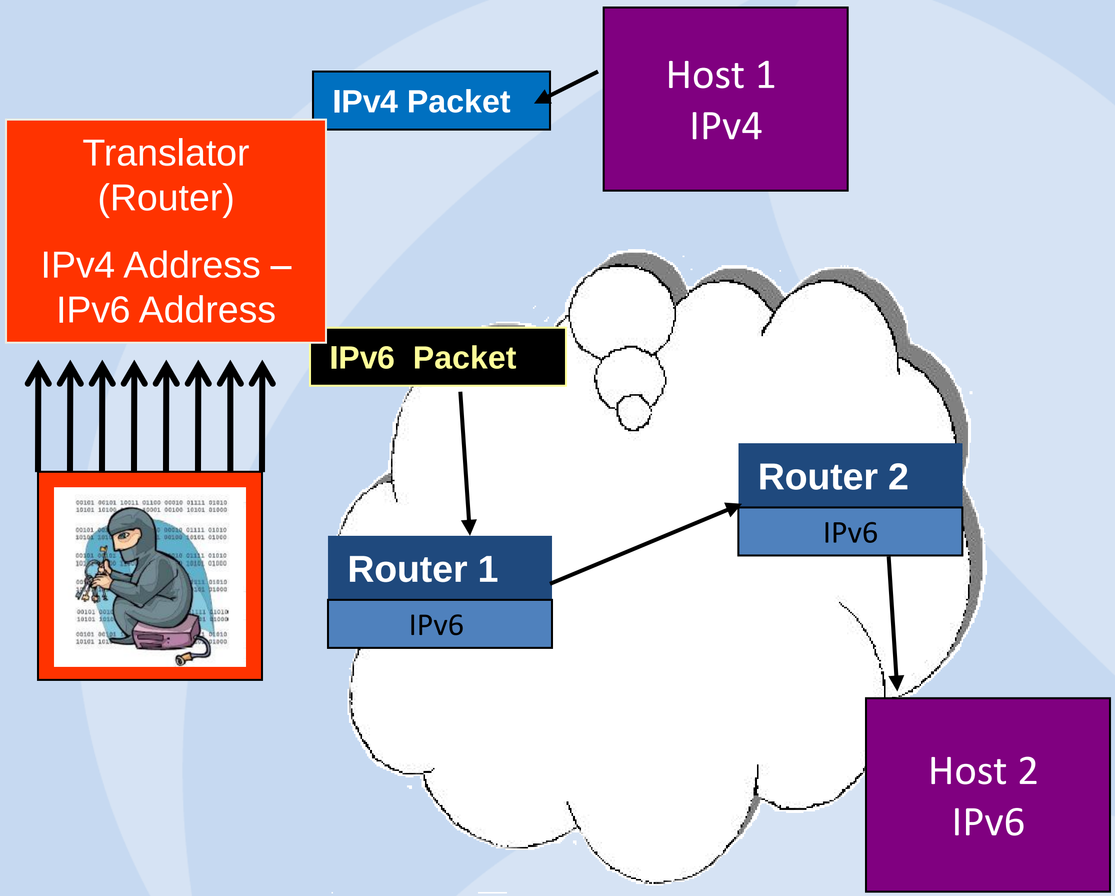
Denial of Service Attacks

- **Address Depletion Denial of Service Attack**

  If dynamic address allocation is being used, and if an attacker within the area serviced by the translation device asks for many connections, then the pool of addresses may get used up, resulting in a Denial of Service attack.

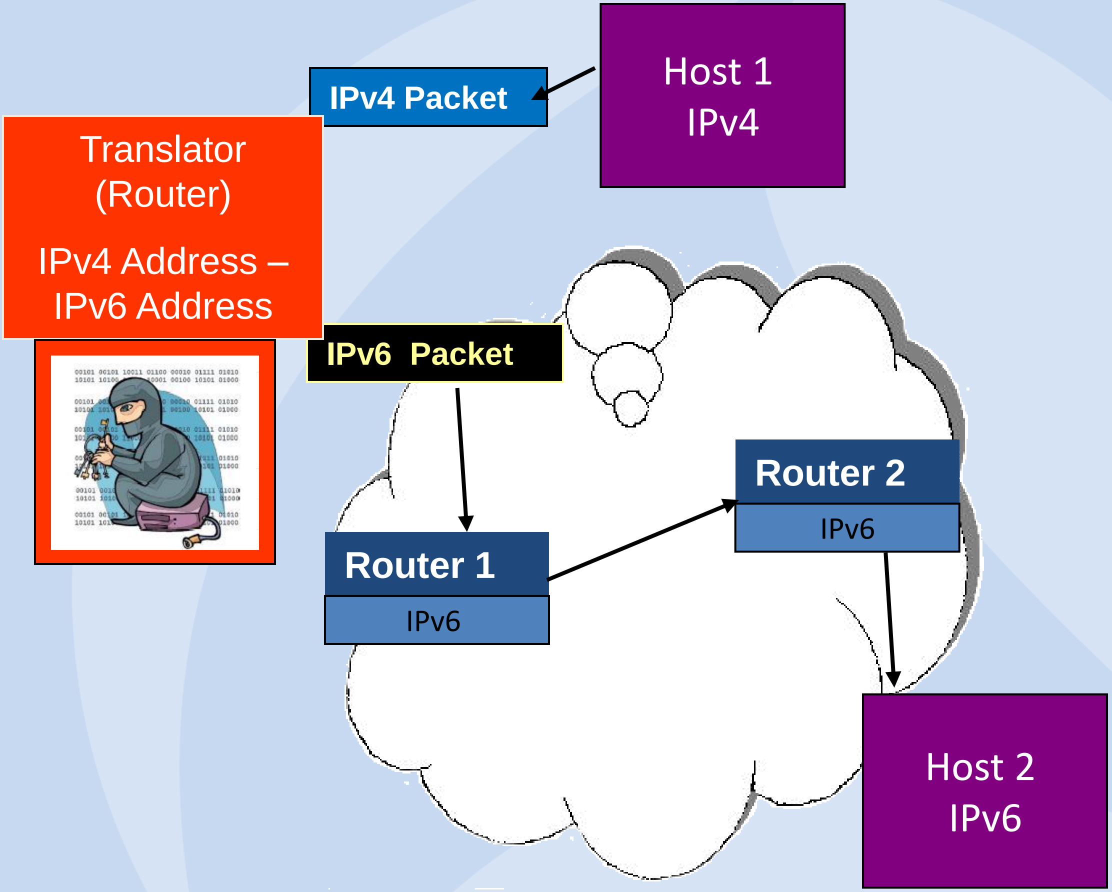
Resource Depletion Denial of Service Attack

- An attacker that knows the IP address of the translation device can send packets directly to it.
- This can use up the device's resources, preventing legitimate nodes from accessing its services.



Bypass Firewall Filters

- A malicious party may try to use a translation device system to bypass access (ingress) filtering for IPv4.
- The IPv6 filters may not be properly set up.
- The translation device systems should implement access controls.



Summary

It seems like ALL my options are bad!!!

Contact:
Nalini.elkins@insidethestack.com