Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July 2021: 5589- 5604

Research Article

Hybrid approach for migration of IPv6 to IPv4 Network for enhancing security in Virtual Private Cloud

Syed Amjad Ali[1], Dr. C Ram Singla[2], Dr Najmuddin aamer[3]
[1]Research Scholar, Department of Electronics & Communication Engg; Shri JJT University Jhunjhunu Rajasthan
[2] Dean, Department of Electronics & Communication Engg; Shri JJT University Jhunjhunu Rajasthan
[3]HOD Computer Engineering Department Theem college of Engineering, Boisar

Abstract:
Internet Protocol version 4 (IPv4) addresses are now in short supply. Because of IPv6's strong features and limitations, many Internet Service Providers (ISPs), researchers, and end users are transitioning from IPv4 to IPv6. For ordinary users, many tunnelling strategies have been used to move to IPv6. These strategies, on the other hand, produce a slew of challenges, including compatibility, complexity, connectivity, and traffic. Because devices do not support IPv6 addresses directly, they are unable to connect with each other due to the distinct header structure and incompatibility of IPv4 and IPv6. The network's performance is also being harmed as a result of the massive increase in data transmission volume. In this study, we suggest merging two tunnelling approaches, IPv6 Rapid Deployment (6RD) and Teredo, to give complete IPv6 connectivity while also improving network performance. Jumbo frames are used to carry large amounts of data to boost network throughput. The primary goal of combining the two methodologies is to create a hybrid network rendering that supports full IPv6 connectivity. Not only does the suggested solution support IPv6, but it also improves network performance. According to simulation results, throughput and packet delivery ratio increase by 9000 bytes and 98 percent, respectively.

Keywords: IPv4, IPv6, Virtual Private Cloud, security, migration

1. Introduction

In the early 1990s, it was identified that IPv4 address space would not be sufficient by 2000. The IPv4 had exhausted in the year 2011 in IANA. This exhaustion has demanded more address spaces which implies the any other solution to handle the situation. Several temporary solutions had evolved along with the demand of new address space for the future use of IP in the Internet world. Some temporary solutions were offered to deal with the shortage of address space, such as NAT (Network Address Translator) or CIDR (Classless Inter Domain Routing) besides with the start of development of new enhanced IP protocol work. A new IPv6 has been designed and enhanced in many ways. The number of bits in the address space is increased from 32 to 128. The flow label field of IPv6 supports payload identification to QoS handling. The IPv6 has extension headers instead of option filed as in the IPv4. This extension header can increase the size of the packets. The DHCP (Dynamic Host Configuration Protocol) is used for IPv6 configuration and hence there is no manual configuration is required. The expansion of Internet usage in the recent decades causes the shortage of IPv4 address space. This matter will threaten the continuation of an internet service that working on IPv4. Internet Engineering Task Force (IETF) has begun in 1994, designed and developed an enhanced size of address space namely Internet Protocol version 6 (IPv6). The IPv6 is a new protocol to replace IPv4 over the coming years.

The new protocol is designed to support the growing use of the Internet, and the address security problems. IPv6 uses a 128-bit address size and will allow for 3.4x1038 distinguish addresses, which are enough to cover any number of Internet connecting devices.

In the recent decades, the IoT demands large number of IP address in the new inter world. As IoT devices are required to connecting to the Internet, each IoT required a IP for their communication with the rest of the world. As the available Pv4 address is very limited, they migrating into IPv6 gradually. But this migration required another decade of time to use IPv6 only networks. The deficiency of IPv4 space headed to deploy IPv6 address space. This deployment may take a decade of time. The Co-existence of IPv4 and IPv6 is unavoidable; hence the transition mechanisms are the dire need of the Internet world. The carried research work provides a solution for the of co-existence of IPv4 and IPv6. The IPv6 yielded the deployment challenges in IPv4-based infrastructures. The replacement of IPv4-based infrastructure with IPv6 is costly and impractical for the small size organizations. The IETF IPng Transition Working Group has proposed many transition strategies to deploy IPv6 into existing networks successfully. The encapsulation strategy is employed in transition mechanisms as the IPv6 encapsulate into IPv4 packets. The transition strategy is an idea of using IPv6 over the IPv4.

2. Internet Protocol Version 4

IPv6 provides agencies with significant benefits by allowing them to improve operational efficiencies and public services. Many of these advantages will not be fully appreciated until IPv6 usage is more widespread. The following are some of the advantages of IPv6:

- Addressing and Routing: IPv6's tremendously broad address space enables global connectivity to a greater number of electronic devices, including smartphones, laptops, in-vehicle computers, televisions, cameras, building sensors, medical devices, and so on.
- Security: IPv6 security is provided via IPsec, which offers authentication, encryption, and integrity protection at the network layer when enabled and configured with the proper key infrastructure.
- IPv6 Address Auto-Configuration: IPv6 address auto-configuration allows basic devices to gain out-of-the-box plug-and-play network access, which is critical for self-organizing networks.
- Mobile Device Support: IPv6-enabled applications can take advantage of seamless mobility. Mobile IPv6 provides the mobility by allowing devices to roam across multiple networks without losing network access.
- Interagency Communication Tools that are Peer-to-Peer (P2P) Collaboration: Media-streaming apps will benefit from true end-to-end connectivity offered by the IPv6 address space and the absence of network address translation (NAT).

This will make it simple to send timely video feeds and high-quality information to millions of people. IPv6 enables an integrated, well-architected platform with all of the above advantages as well as headroom for future development and enhancement.

3. IPv6 termination for HTTP(S), SSL Proxy, and TCP Proxy Load Balancing

Your users' IPv6 connections are accepted by the load balancer, which then proxies them to your backends. For the following, you can configure both IPv4 and IPv6 external addresses:

- SSL proxy load balancers • TCP proxy load balancers • external HTTP(S) load balancers
- Load balancing for IPv6 on a global scale (click to enlarge)
- IPv6 termination allows you to handle IPv6 requests from your users and forward them to your backends via IPv4. You can perform the following with IPv6:
- For multi-region deployment, use a single anycast IPv6 address. For application instances running across several regions, you simply require one load balancer IPv6 address. This implies your DNS server only has one AAAA record and you won't have to load balance between

numerous IPv6 addresses. Clients do not have to cache AAAA entries because there is just one address to cache. User queries for the IPv6 address are load balanced to the nearest healthy backend with available capacity.

- Load balance IPv6 client traffic via HTTP, HTTPS, HTTP/2, TCP, and SSL/TLS.
- With a single IPv6 load balancer address, you can overflow across regions. The global load balancer automatically routes requests from users to the next nearest area with available resources if backends in one region are out of resources or unhealthy. When the nearest region has available resources, global load balancing switches back to this region to serve. The Premium Tier of Network Service Tiers is required for global load balancing.
- Use a dual stack. Create two load balancer IP resources—one for IPv6 and the other for IPv4—and associate them with the same IPv4 application instances to serve both IPv6 and IPv4 clients. Clients connecting to IPv4 addresses connect to IPv4 addresses, while IPv6 clients connect to IPv6 addresses. These clients are then immediately load balanced to the nearest available healthy backends. We don't charge for IPv6 forwarding rules, therefore you only have to pay for IPv4 ones.

Using the same backends for IPv4 and IPv6 traffic

As demonstrated in the following figure, configuring IPv6 termination for your load balancers allows your backends to appear as IPv6 apps to your IPv6 clients.

Termination of IPv6 for load balancing (click to enlarge)

When a user uses IPv6 to connect to the load balancer, the following happens:

1. Your load balancer waits for user connections using its IPv6 address and forwarding rule.
2. An IPv6 client establishes an IPv6 connection with the load balancer.
3. The load balancer serves as a reverse proxy, terminating IPv6 client connections. It sends the request to a backend using an IPv4 connection.
4. On the return path, the load balancer gets the IPv4 response from the backend and forwards it to the original client through an IPv6 connection.

Allocation of IPv6 addresses for load balancer forwarding rules

When you set up an external load balancer, you provide it one or more global forwarding rules, each with a publicly routed IPv4 or IPv6 IP address (or both). This IP address can be used in your site's DNS records.

When creating a forwarding rule, you have the option of using a static IP address designated for your project or having the forwarding rule acquire an ephemeral IP address automatically. Your project is assigned a static IP address, which you can keep until you decide to release it. For as long as the forwarding rule persists, an ephemeral address is part of it. The ephemeral address is released back into the Google Cloud pool if you delete the forwarding rule.

If your load balancer requires both an IPv4 and an IPv6 address, you can set two forwarding rules, assigning an IPv4 address to one and an IPv6 address to the other. After that, you can link both rules to the same load balancer.

Format for IPv6 addresses

IPv6 forwarding rules on Google Cloud are assigned a /64 IPv6 address range. The load balancer permits traffic from IPv6 addresses with the least significant 64 bits set to 0, yet the gcloud command-line tool lists IPv6 addresses with the least significant 64 bits set to 0. As a result, depending on which IPv6 server IP address the client connects to, various load balancer IPv6 addresses in the allocated range may appear in X-Forwarded-For headers.

HTTP(S) Load Balancing client IP header with IPv6 termination

The original source IP address is substituted with the load balancer's IP address when the load balancer proxies the IPv6 connection from the client to an IPv4 connection to your backend. Backends, on the other hand, frequently require knowledge of the original source IP address for logging, decision-

making, and other purposes. The original IPv6 client IP address is included in an HTTP header that is propagated to the backends by Google Cloud.

IPv6 HTTP headers are comparable to IPv4 headers. The following is the format for requests:

• X-Forwarded-For: CLIENT IP ADDRESS, GLOBAL FORWARDING RULE EXTERNAL IP ADDRESSES X-Forwarded-For: CLIENT IP ADDRESS, GLOBAL FORWARDING RULE EXTERNAL IP ADDRESSES X-Forwarded-For:

The IP address of the load balancer is displayed in the final element. The client IP address as seen by the load balancer is shown in the second to final element. There could be other components in the mix. When the client or intermediary proxies add further information to the X-Forwarded-For header, Before delivering the request to the load balancer, add X-Forwarded-For headers.

This is an example of an X-Forwarded-For header:

X-Forwarded-For: 2607:f8b0:4005:801::200e, 2001:db8:abcd:1::1234, 2607:f8b0:4005:801::200e

The client's IPv6 address is 2001:db8:abcd:1::1234. The external HTTP(S) load balancer's IPv6 address is 2607:f8b0:4005:801::200e.

4. Requirements for Transition

There is no need for global coordination during the changeover. Your websites and Internet service provider (ISP) can both make their own transitions. Furthermore, during the shift, every attempt has been made to reduce the amount of dependencies. The change, for example, does not necessitate upgrading routers to IPv6 before upgrading hosts.

During the transformation process, different sites face distinct challenges. Early adopters of IPv6 are also likely to have different issues than IPv6 users in production. The transition tools currently available are defined in RFC 1933. The paucity of IPv4 address space, or the requirement to employ new features in IPv6, or both, are the reasons for the change. For existing protocols, the IPv6 definition specifies 100 percent compatibility. During the change, current applications must also be compatible.

The following terminology have been specified to help you understand how the transition works.

- An IPv4-only node is a host or router that exclusively uses IPv4. IPv6 is not understood by an IPv4-only node. Before the changeover, there is an installed base of IPv4 hosts and routers that are IPv4-only nodes.
- IPv6/IPv4 node — A dual-stack node is a host or router that implements both IPv4 and IPv6.
- IPv6-only node — An IPv6-only host or router that does not support IPv4.
- IPv6 node — Any IPv6-enabled host or router. IPv6 nodes are both IPv6/IPv4 and IPv6-only nodes.
- IPv4 node — Any host or router that supports IPv4, including IPv6/IPv4 and IPv4-only nodes.
- 6to4 router – Any border router having an IPv4 network connection configured with a 6to4 pseudo-interface. A 6to4 router is the terminus of a 6to4 tunnel through which packets are forwarded to another IPv6 site.
- IPv6 host with a 6to4–derived address - Any IPv6 host having an interface set with a 6to4–derived address.
- Site - A section of the Internet's private topology that does not transport transit traffic for anybody and everyone. The site has the ability to cover a broad geographic area. A multinational corporation's private network, for example, is one site.

Transitional Tools that are standardised

- When you upgrade your hosts and routers to IPv6, the hosts and routers preserve their IPv4 capabilities, according to RFC 1933. As a result, IPv6 is backwards compatible with all IPv4 protocols and applications. Dual-stack hosts and routers are what they're called.
- These hosts and routers communicate with each other using a name service, such as DNS, to determine which nodes are IPv6 competent.

- IPv4 addresses can be included in IPv6 address forms.
- You can tunnel IPv6 packets over IPv4 packets to get around routers that haven't yet been upgraded to IPv6.

5. Dual-Stack Implementation

The phrase "dual-stack" usually refers to a full duplicate of the protocol stack from applications to the network layer. The OSI and TCP/IP protocols, which run on the same system, are an example of total duplication. In the context of the IPv6 transition, however, dual-stack refers to a protocol stack that includes both IPv4 and IPv6. The rest of the stack is exactly the same. As a result, the same transport protocols, such as TCP and UDP, can be used on both IPv4 and IPv6 networks. Additionally, the same programmes can be executed on both IPv4 and IPv6 networks.

The following figure illustrates dual-stack protocols through the OSI layers.

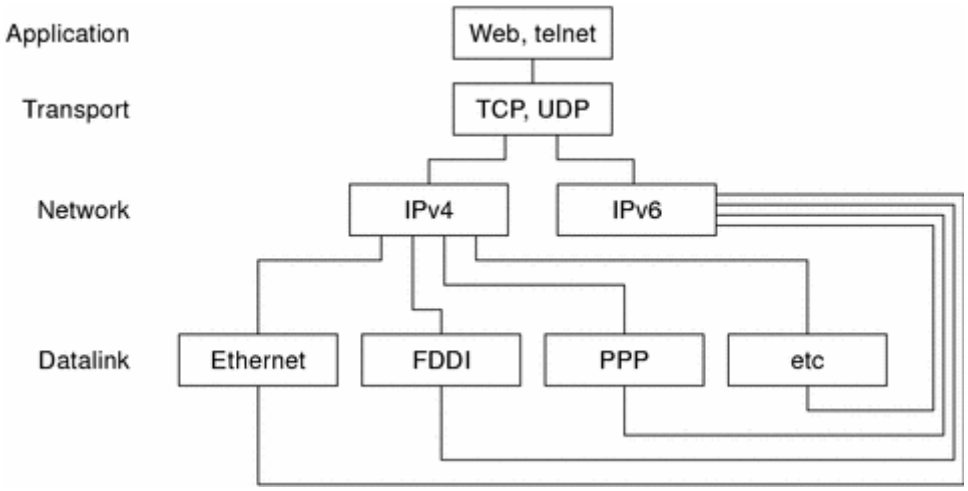

Figure 1 Dual-Stack Protocols

Subsets of both hosts and routers are upgraded to support IPv6 in addition to IPv4 in the dual-stack strategy. The dual-stack strategy ensures that upgraded nodes can always use IPv4 to communicate with IPv4-only nodes.

Setting up Name Services

In order to select which IP version to use when sending, a dual node must first establish whether the peer can support IPv6 or IPv4. A dual node can choose which IP version to use since it has control over the information that goes into the name service. In the name service, you specify an IPv4 node's IP address and an IPv6 node's IP address. As a result, a dual node's name service has both addresses.

The existence of an IPv6 address in the name service indicates that the node can be reached via IPv6. The node, on the other hand, can only be reached by nodes that have obtained information from that name service. Placing an IPv6 address in NIS, for example, indicates that the IPv6 host can be reached through IPv6. The IPv6 host, on the other hand, can only be reached by IPv6 and dual nodes in that NIS domain. The placement of an IPv6 address in global DNS necessitates the node's accessibility from the Internet's IPv6 backbone. This condition is identical to that of IPv4. For instance, the mail delivery function necessitates the existence of IPv4 addresses for nodes that may be contacted via IPv4. The HTTP proxy operation is in a similar condition. When there is no IPv4 reachability due to firewalls, for example, the name service must be partitioned into an inside firewall and an outside firewall database. As a result, the IPv4 addresses are only accessible when the IPv4 addresses may be reached.

The type of address that can be received from the name service is unrelated to the protocol used to access the name service. When a dual node communicates with IPv4-only nodes, this name service support and dual-stacks enable the dual node to use IPv4. Additionally, when a dual node communicates

with IPv6 nodes, this name service capability allows the dual node to use IPv6. The destination, on the other hand, must be reachable through an IPv6 route.

Using IPv4-Compatible Address Formats

In many instances, you can represent a 32-bit IPv4 address as a 128-bit IPv6 address. The transition mechanism defines the following two formats.

- **IPv4–compatible address**

000 ... 000	IPv4 Address

- **IPv4–mapped address**

000 ... 000	0xffff	IPv4 Address

An IPv6 node is represented using the compatible format. You can use this format to configure an IPv6 node to utilise IPv6 even if it doesn't have a genuine IPv6 address. Because you may use automated tunnelling to bridge IPv4–only routers with this address format, you can experiment with different IPv6 deployments. The IPv6 stateless address autoconfiguration technique, on the other hand, cannot be used to configure these addresses. Existing IPv4 techniques, such as DHCPv4 or static configuration files, are required for this mechanism to work.

An IPv4 node is represented using the mapped address format. The socket API is the only currently defined application of this address format. Both IPv6 and IPv4 addresses can use the same address format in an application. An IPv4 address can be represented as a 128-bit mapped address in the common address format. IPv4–to–IPv6 protocol translators, on the other hand, allow these addresses to be used.

Mechanism of Tunneling

All routers in the path between two IPv6 nodes do not need to implement IPv6 to minimise any dependencies during the changeover. Tunneling is the name for this mechanism. IPv6 packets are encapsulated within IPv4 packets, which are then routed through IPv4 routers. The tunnelling process across IPv4 routers is depicted in the diagram below (R).

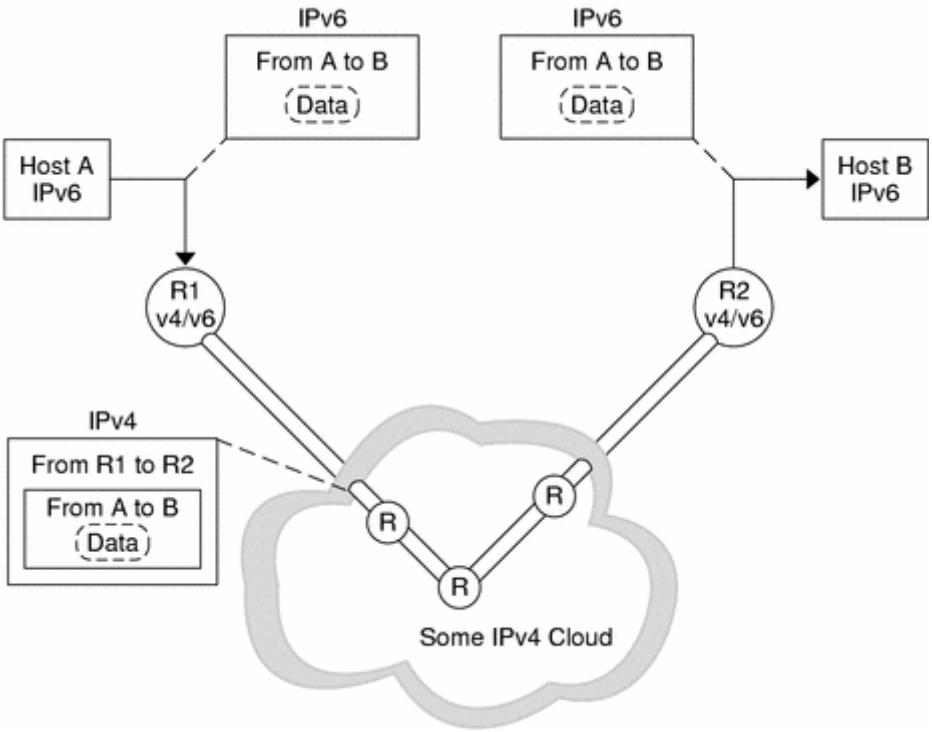

Figure 2 Tunneling Mechanism

The following are the various applications of tunnelling in the transition:

- As seen in the preceding illustration, configured tunnels between two routers
- Tunnels that automatically terminate at the dual hosts

The Internet currently uses a configured tunnel for additional reasons, such as the MBONE, the IPv4 multicast backbone. In terms of operation, the tunnel consists of two routers that are configured to have a virtual point-to-point link over an IPv4 network. For the foreseeable future, this type of tunnel will most likely be used on some areas of the Internet.

Tunnels that operate automatically

IPv4–compatible addresses are required for automatic tunnels. When IPv6 routers are unavailable, automatic tunnels can be utilised to connect IPv6 nodes. By setting an automatic tunnelling network interface, these tunnels can begin on either a dual host or a dual router. Tunnels always come to a halt on the dual host. The endpoint of the tunnel, the destination IPv4 address, is dynamically determined by extracting the address from the IPv4–compatible destination address.

Interaction with Programs

The utilisation of IPv6 is dependent on the applications, even on a node that has been upgraded to IPv6. A networking API that queries the name service for IPv6 addresses may not be used by an application. The application may make use of an API, such as sockets, that necessitates changes in the code. Additionally, the API provider, such as a java.net class implementation, may not support IPv6 addresses. In either case, the node sends and receives IPv4 packets in the same way that an IPv4 node would.

Within the Internet community, the following terms have become commonplace:

- IPv6 incompatibility—This programme is unable to handle IPv6 addresses. Nodes without an IPv4 address are unable to communicate with this application.
- IPv6–aware—This programme can communicate with nodes that don't have an IPv4 address, allowing it to handle larger IPv6 addresses. The address may be transparent to the programme in some circumstances, such as when the API hides the information and format of the real address.
- IPv6–enabled—In addition to being IPv6–aware, this programme can leverage IPv6–specific features such flow labels. The programmes that are enabled can still use IPv4 albeit in a degraded manner.
- IPv6–necessary—This application necessitates the use of an IPv6–specific functionality. This application is not compatible with IPv4.

6. Interoperability between IPv4 and IPv6

Existing IPv4 apps must continue to interact with newer IPv6–enabled applications during the gradual transition from IPv4 to IPv6. Initially, companies supply dual-stack systems for the host and router. A dual-stack is a protocol stack that supports both IPv4 and IPv6. IPv4 programmes continue to execute on a dual– stack with at least one IPv6 interface that is likewise IPv6 enabled. These applications do not require any modifications or porting.

On a dual-stack, IPv6 applications can also use the IPv4 protocol. IPv6 apps use an IPv6 address that is mapped to an IPv4 address. IPv6's design eliminates the requirement for separate IPv4 and IPv6 applications. You don't need an IPv4 client on a dual host to "speak" to a server on an IPv4-only host, for example. You also don't need an IPv6 client to communicate with an IPv6 server. All you have to do now is port their IPv4 client software to the new IPv6 API. Only IPv4–only servers can connect with the client. The client can also communicate with IPv6 servers running on dual or IPv6–only hosts.

The client's address from the name server decides whether IPv6 or IPv4 is utilised. For example, if a server's name server has an IPv6 address, the server is running IPv6.

The compatibility of IPv4 and IPv6 clients and servers is summarised in the table below. The table assumes the dual-stack host has an IPv4 and IPv6 address in each name service database.

Table 1 Client-Server Applications: IPv4 and IPv6 Interoperability

Type of Application (Type of Node)	**IPv6-Unaware Server (IPv4-Only Node)**	**IPv6-Unaware Server (IPv6-Enabled Node)**	**IPv6-Aware Server (IPv6-Only Node)**	**IPv6-Aware Server (IPv6-Enabled Node)**
IPv6-unaware client (IPv4-only node)	IPv4	IPv4	X	IPv4
IPv6-unaware client (IPv6-enabled node)	IPv4	IPv4	X	IPv4
IPv6-aware client (IPv6-only node)	X	X	IPv6	IPv6
IPv6-aware client (IPv6-enabled node)	IPv4	(IPv4)	IPv6	IPv6

X means that the server cannot communicate with the client.

IPv4 indicates that compatibility is contingent on the client's choice of address. The client will fail if it selects an IPv6 address. An IPv4 datagram is delivered successfully when an IPv4 address is returned to the client as an IPv4–mapped IPv6 address.

Most IPv6 solutions in the early stages of deployment are dual-stack nodes. Most vendors do not release IPv6–only implementations at first.

7. Scenarios for Site Transition

During the changeover period, different processes are required for each site and ISP. Examples of site transition situations are provided in this section.

The update of name services to accommodate IPv6 addresses is the first stage in transitioning a site to IPv6. Upgrade your DNS server to one that supports the new AAAA (quad-A) protocol, such as BIND 4.9.4 and later. For storing IPv6 addresses, two new NIS maps and a new NIS+ table have been added. Any Solaris system may generate and administer the new NIS maps and NIS+ table.

You can begin switching hosts after the name service is able to distribute IPv6 addresses. You can change hosts in a variety of ways:

- One host at a time should be upgraded. Automatic tunnelling and IPv4–compatible addresses are recommended. There are no routers that need to be upgraded. For the first experimental transition, use this procedure. This solution only provides a portion of the IPv6 benefits. This approach does not support IP multicast or stateless address autoconfiguration. This scenario can be used to test if programmes work over IPv6. The application's ability to leverage IPv6 IP-layer security is also verified in this situation.
- One subnet at a time should be upgraded. Use tunnels between the routers that have been configured. At least one router per subnet gets upgraded to dual in this case. The site's dual routers are connected via tunnels that have been configured. The hosts on such subnets can then

take advantage of all IPv6 functionalities. You can remove the configured tunnels when more routers are upgraded in this stepwise method.

- Prior to upgrading any host, upgrade all routers to dual. Despite the fact that this strategy appears to be well-organized, it does not provide any IPv6 benefits until all routers have been upgraded. The incremental deployment technique is constrained in this case.

As a Transition Mechanism, 6to4

6to4 is a popular interim solution for transitioning from IPv4 to IPv6 addressing in the Solaris operating system. 6to4 allows isolated IPv6 sites to interact over an IPv4 network that does not support IPv6 via an automated tunnel. You must configure a border router on your IPv6 network as one endpoint of the 6to4 automatic tunnel to use 6to4 tunnels. After then, the 6to4 router can join a tunnel to another 6to4 site or, if necessary, a native IPv6, non-6to4 site.

This area contains reference materials for the 6to4 disciplines listed below:

- The 6to4 tunnel's topology
- 6to4 addressing, including the advertisement format
- Packet flow via a 6to4 tunnel
- Tunnel topology between a 6to4 router and a 6to4 relay router
- Things to think about before configuring 6to4 relay router support

Participants in a 6to4 Tunnel

The following figure shows a 6to4 tunnel between two 6to4 sites.

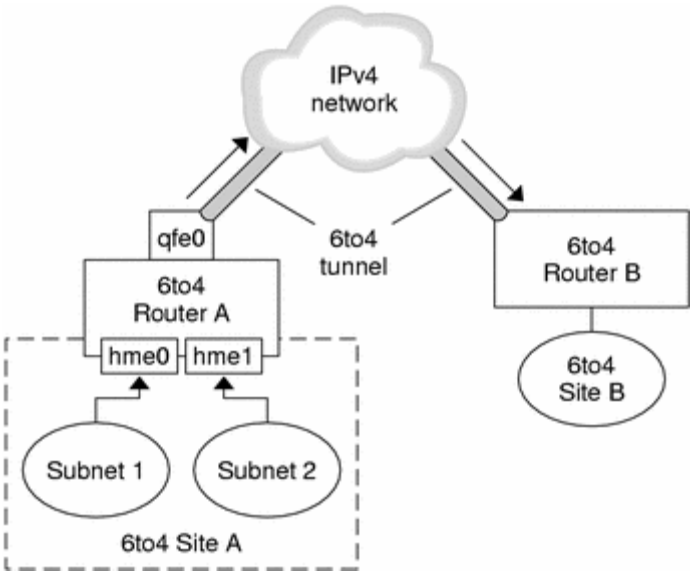

Figure 3 Tunnel Between Two 6to4 Sites

Site A and Site B, two isolated 6to4 networks, are depicted in the diagram. Each site has a router connected to an IPv4 network from the outside. A 6to4 tunnel via the IPv4 network joins the 6to4 sites in the diagram.

You must configure at least one router interface for 6to4 support before an IPv6 site can become a 6to4 site. This interface is responsible for connecting to the IPv4 network from the outside. On qfe0, the address you configure must be globally unique. The interface qfe0 on border Router A connects Site A to the IPv4 network in the previous diagram. Before you may configure qfe0 as a 6to4 pseudo-interface, it must already be configured with an IPv4 address.

6to4 Site A is divided into two subnets, each of which is connected to Router A's hme0 and hme1 interfaces. Upon receiving the advertising from Router A, all IPv6 hosts on either subnet of Site A automatically reconfigure with 6to4–derived addresses.

Site B is the tunnel's opposite terminal from Site A. A border router on Site B must be setup for 6to4 compatibility in order to accept traffic from Site A correctly. Otherwise, packets received from Site A are ignored and dropped by the router.

Addressing with a 6:4 Ratio

In /etc/inet/ndpd.conf, just like with native IPv6 routers, you must advertise the subnet prefixes produced from the site 6to4 prefix. The pieces of a prefix for a 6to4 site are shown in the following diagram, as defined in 6to4 Prefix Format and 6to4 Advertisement Example.

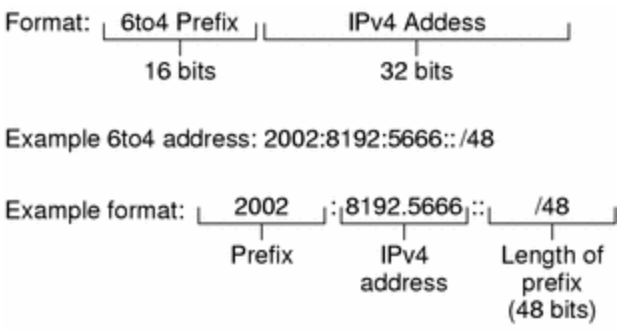

Figure 4 Parts of a Site Prefix

The next figure shows the parts of a subnet prefix for a 6to4 site, such as you would include in the ndpd.conf file.

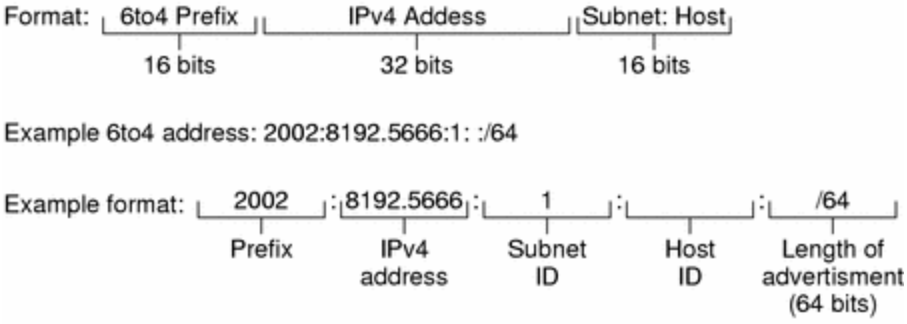

Figure 5 Parts of a Subnet Prefix

6to4 Prefix Format

The format line in the previous figure contains the following parts.

Part	Length	Definition
Prefix	16 bits	6to4 prefix 2002 (0x2002).
IPv4 address	32 bits	Unique IPv4 address that is already configured on the 6to4 interface. For the advertisement, you specify the hexadecimal representation of the IPv4 address, rather than the IPv4 dotted–decimal representation.
Subnet ID	16 bits	Subnet ID, which must be a value that is unique for the link at your 6to4 site.

6to4 Advertisement Example

The example in the previous figure has the following values.

Advertisement Part	Corresponding Value
6to4 prefix	2002
IPv4 address	8192:56bb, which corresponds to IPv4 address 129.146.87.188
Subnet ID	1
/64	Length of prefix

6to4-Derived Addressing on a Host

When an IPv6 host receives the 6to4–derived prefix by way of a router advertisement, the host automatically reconfigures a 6to4–derived address on an interface. The address has the following form.

```
prefix:IPv4 address:subnet ID:host ID/64
```

The results of ifconfig –a on a host with a 6to4 interface might resemble the following:

```
qfe1:3: flags=2180841<UP,RUNNING,MULTICAST,ADDRCONF,ROUTER,IPv6>
 mtu 1500 index 7
      inet6 2002:8192:56bb:9258:a00:20ff:fea9:4521/64
```

The 6to4–derived address follows inet6 in the output from ifconfig.

Address Part	Corresponding Value
Prefix	2002, which is the 6to4 prefix
IPv4 value	8192:56bb, which is the IPv4 address, in hexadecimal notation, for the 6to4 pseudo-interface that is configured on the 6to4 router
subnet ID	9258, which is the address of the subnet of which this host is a member
MAC address	a00:20ff:fea9:4521, which is the link layer address of the host interface that is now configured for 6to4

Packet Flow Through the 6to4 Tunnel

This section explains how packets get from a host at one 6to4 site to a host at another 6to4 site. The structure depicted in Figure 4–3 is used as an example in the next scenario. The scenario also assumes that the 6to4 routers and 6to4 hosts are already set up.

1. A host on 6to4 Site A's Subnet 1 transmits a message to a host on 6to4 Site B's Subnet 2. A source 6to4–derived address and a destination 6to4–derived address are included in each packet header in the flow.
2. Outgoing packets are received by 6to4 Router A, which forms a tunnel to 6to4 Site B over an IPv4 network.
3. Each 6to4 packet is encapsulated in an IPv4 header by Site A's router. The packet is then forwarded via the IPv4 network using regular IPv4 routing protocols.

4. Any IPv4 routers that the packets meet forward the packets using the destination IPv4 address. The interface on Router B, which also functions as the 6to4 pseudo-interface, has a globally unique IPv4 address.
5. When packets from Site A arrive at Router B, the IPv6 packets are decapsulated from the IPv4 header.
6. Router B subsequently forwards the packets to the receiver host at Site B using the IPv6 packet's destination address.

Tunnels to a 6to4 Relay: Considerations Router

Endpoints for tunnels from 6to4 routers that need to communicate with native IPv6, non-6to4 networks are 6to4 relay routers. Relay routers act as a link between the 6to4 site and the native IPv6 site. Because this technique is extremely insecure, the Solaris operating system disables 6to4 relay router capability by default. If your location requires a tunnel, however, you can use the 6to4relay command to enable the tunnelling scenario below.

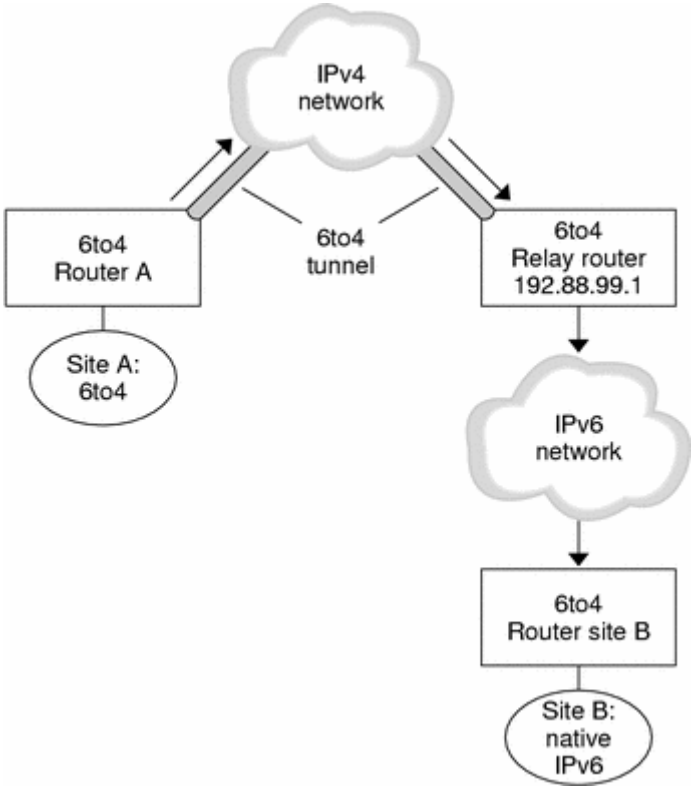

Figure 6 Tunnel From a 6to4 Site to a 6to4 Relay Router

6to4 Site A must communicate with a node at native IPv6 Site B, as shown in Figure 4–6. The diagram depicts the traffic flow from Site A to a 6to4 tunnel on an IPv4 network. The tunnel's endpoints are 6to4 Router A and a 6to4 relay router. The IPv6 network, to which IPv6 Site B is connected, is beyond the 6to4 relay router.

8. Results and Analysis

There's a difference between a 6to4 site and a native IPv6 site.

The packet flow from a 6to4 site to a native IPv6 site is described in this section. The scenario depicted in Figure 4–6 is used as an example in the article.

1. A host on 6to4 Site A sends a message to a host on native IPv6 Site B as the destination. The source address of each packet header in the flow is a 6to4–derived address. A regular IPv6 address is used as the destination address.

2. 6to4 Router A receives the outgoing packets and forms a tunnel to a 6to4 relay router over an IPv4 network.

The IP 192.88.99 is used by 6to4 relay routers in the 6to4 relay router anycast group.

1. The default address for 6to4 relay routers is this anycast address. You can override the default and give the IPv4 address of a specific 6to4 relay router if necessary.
2. The 6to4 router at Site A encapsulates each packet in an IPv4 header with the 6to4 relay router's IPv4 address as its destination. The packet is forwarded over the IPv4 network by the 6to4 router using regular IPv4 routing protocols. The packets are forwarded to the 6to4 relay router by any IPv4 routers they meet.
3. The packets destined for the 192.88.99.1 anycast group are retrieved by the physically closest anycast 6to4 relay router to Site A.
4. The 6to4 packets' IPv4 header is decapsulated by the relay router, disclosing the native IPv6 destination address.
5. The relay router subsequently transfers the newly IPv6–only packets over the IPv6 network, where a router at Site B eventually retrieves them. The packets are subsequently forwarded to the destination IPv6 node by the router.

6to4 Relay Router Support Security Issues

A tunnel between a 6to4 router and a 6to4 relay router is insecure by definition. In such a tunnel, security issues such as the following are inevitable.

- Despite the fact that 6to4 relay routers encapsulate and decapsulate packets, they do not inspect the data contained within the packets.
- On tunnels to a 6to4 relay router, address spoofing is a serious problem. The 6to4 router is unable to match the IPv4 address of the relay router with the IPv6 address of the source for incoming traffic. As a result, spoofing the IPv6 host's address is simple. The 6to4 relay router's address can also be faked.
- There is no trust mechanism between 6to4 routers and 6to4 relay routers by default. As a result, a 6to4 router is unable to determine whether or not the 6to4 relay router can be trusted, or even if it is a genuine 6to4 relay router. If there isn't a trust relationship between the 6to4 site and the IPv6 destination, both sites are vulnerable to attacks.

The Internet Draft Security Considerations for 6to4 Relay Routers explains these and other security issues with 6to4 relay routers. Support for 6to4 relay routers should generally be enabled only for the following reasons:

- Your 6to4 website is designed to communicate with a secure IPv6 network. On a campus network with isolated 6to4 sites and native IPv6 sites, for example, you might enable 6to4 relay router functionality.
- There is a strong business purpose for your 6to4 site to communicate with native IPv6 hosts.
- You've put in place the security tests and trust models recommended in the Internet Draft, Security Considerations for 6to4.

6to4 Router Issues That Have Been Recognized

6to4 configuration is affected by the following known bugs:

- 4709338 – RIPng implementation that understands static routes is required.
- 4152864 – Using the same tsrc/tdst pair for two tunnels works

Static Routes are being implemented at the 6to4 site (BugID 4709338)

On 6to4 sites with routers that are internal to the 6to4 border router, the following problem occurs. The static route 2002::/16 is automatically added to the routing table on the 6to4 router when you configure the 6to4 pseudo-interface. The Solaris RIPng routing system has a restriction that prohibits this static route from being published to the 6to4 site, as described in Bug 4709338.

For Bug 4709338, you can use one of the following workarounds.

- Add the 2002::/16 static route to all intra-site routers within the 6to4 site's routing tables.
- On the internal router of the 6to4 site, use a routing protocol other than RIPng.

Tunnels with the Same Source Address Configuration (BugID 4152864)

When two tunnels are configured with the same tunnel source address, which is a severe issue for 6to4 tunnels, Bug ID 4152864 describes the problems that ensue.

Other Methods of Transition

If the dual nodes have an IPv4 address, the previously defined techniques handle interoperability between dual nodes and IPv4 nodes. Interoperability between IPv6-only nodes and IPv4-only nodes is not handled by the mechanisms. Interoperability between dual nodes with no IPv4 address and IPv4-only nodes is also not handled by the procedures. The vast majority of implementations can be made dual. A dual solution, on the other hand, necessitates adequate IPv4 address space to assign one address to each node that needs to communicate with IPv4-only nodes.

There are several options for achieving this interoperability without requiring any new transition mechanisms.

- Use application layer gateways (ALG) at the point where IPv6-only nodes and the rest of the Internet meet. HTTP proxies and mail relays are two examples of ALGs in use today.
- Network address translators (NAT) units for IPv4 are already on the market. The NAT boxes translate between private IP addresses on the inside, such as network 10—see RFC 1918—and public IP addresses on the outside. These businesses will very certainly upgrade their NAT devices to provide IPv6–to–IPv4 address translation.

9. Conclusion & Future scope

Both ALG and NAT methods, unfortunately, provide single points of failure. The Internet becomes less effective as a result of these measures. The Internet Engineering Task Force is working on a better solution for interoperability between IPv6-only nodes and IPv4-only nodes. One suggestion is to employ header translators in conjunction with a method of dynamically allocating IPv4–compatible addresses. Another suggestion is to assign IPv4–compatible addresses on demand and utilise IPv4 in IPv6 tunnelling to connect IPv6-only routers.

If the IPv6 addresses in use may be represented as IPv4 addresses, the stateless header translator converts between IPv4 and IPv6 header formats. IPv4 compatibility is required for the addresses. Alternatively, the addresses must be IPv4-mapped. The IPv6 protocol already has support for these translators. Except for encrypted packets, the translation can take place without any data loss. Source routing, for example, is a rarely used feature that can result in data loss.

References

[1]. S. E. Deering, ‘‘Internet protocol, version 6 (IPv6) specification”, 1998.
[2]. (13/2/2018). IPv6 Adoption – Google Internet Statistics. Available: https://www.google.com/intl/en/ipv6/statistics.html.
[3]. A. Shubair, Survey of security issues in IPv4 to IPv6 tunnel transition mechanisms, Int. J. Secur. Netw. 12 (2) (2017) 83–102.
[4]. J.H. Jafarian, E. Al-Shaer, Q. Duan, An effective address mutation approach for disrupting reconnaissance attacks, IEEE Trans. Inf. Forensics Secur. 10 (12) (2015) 2562–2577.
[5]. J.M. Ehrenfeld, Wannacry, cybersecurity and health information technology: a time to act, J. Med. Syst. 41 (7) (2017) 104.
[6]. F. Gont, ‘‘A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC) RFC No 7217”, 2014.
[7]. S. Hagen, IPv6 Essentials, third ed., O’Reilly Media Inc, California, USA, 2014.
[8]. F. Gont and T. Chown, Network reconnaissance in IPv6 networks RFC No 7707, 2016.

[9]. R. Asati, H. Singh, W. Beebee, C. Pignataro, E. Dart, and W. George, Enhanced Duplicate Address Detection RFC No 7527, 2015.

[10]. H. Rafiee and C. Meinel, ‘‘SSAS: A simple secure addressing scheme for IPv6 autoconfiguration,” in Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on, 2013, pp. 275-282: IEEE

[11]. S. Groat, M. Dunlop, R. Marchany, J. Tront, ‘‘The Privacy Implications of Stateless IPv6 Addressing,”, in: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, ACM, 2010, p. 52.

[12]. T. Savolainen, J. Soininen, B. Silverajan, Ipv6 addressing strategies for iot, IEEE Sensors J. 13 (10) (2013) 3511–3519.

[13]. T. Narten R. Draves S. Krishnan ‘‘Privacy Extensions for Stateless Address Autoconfiguration in IPv6 RFC No 4941,” 2007.

[14]. W. Haddad, E. Nordmark, F. Dupont, M. Bagnulo, B. Patil, Privacy for mobile and multi-homed nodes: MoMiPriv problem statement, Internet Draft (2005).

[15]. R. Koodli ‘‘IP Address Location Privacy and Mobile IPv6: Problem Statement RFC No 4882,” 2007.

[16]. A.O. Ade-Ibijola, A simulated enhancement of Fisher-Yates algorithm for shuffling in virtual card games using domain-specific data structures, Int. J. Comput. Appl. 54 (11) (2012).

[17]. C.C. Zou, D. Towsley, W. Gong, On the performance of Internet worm scanning strategies, Perform. Eval. 63 (7) (2006) 700–723.

[18]. F. Gont, W. Liu, A Method for Generating Semantically Opaque Interface Identifiers (IIDs) with the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) RFC No 7943, 2016

[19]. Deering, S., & Hinden, R. (December1998). Internet Protocol Version 6 (IPv6) Specification, IETF RFC 2460.

[20]. Madhav Panthee, Dr. Yogesh Kumar Sharma, Review of E-Government Implementation, International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 6, Issue 1, March 2019, pp. 26-30

[21]. Narten, T., Nordmark, E., Simpson, W., & Soliman, H. (September 2007). Neighbor Discovery for IP version 6 (IPv6),IETF RFC 4861.

[22]. Thomson, S., Narten, T., & Jinmei, T. (September 2007). IPv6 Stateless Address Autoconfiguration, IETF RFC 4862 .

[23]. Conta, A., Deering, S., & Gupta, M. (March 2006). Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification),IETF RFC 4443.

[24]. Hinden, R., & Deering, S. (February 2006). IP Version 6 Addressing Architecture, , IETF RFC 4291.

[25]. Kent, S., & Seo, K. (December 2005). Security Architecture for the Internet Protocol , IETF RFC 4301.

[26]. Ziring N. (May 2006). Router Security Configuration Guide Supplement - Security for IPv6 Routers. [Online]. Available:www.nsa.gov/ia/_files/routers/I33-002R-06.pdf

[27]. Hermann, P.-Seton (2002). Security Features in IPv6. [Online]. Available: www.sans.org/reading_room/whitepapers/.../security_features_in_ipv6_380

[28]. Sotillo, S. (2006). IPv6 Security Issues. [Online]. Available: www.infosecwriters.com/text_resources/pdf/IPv6_SSotillo.pdf

[29]. Sailan, M.K., Hassan, R., & Patel, A. (2009). A Comparative Review of IPv4 and IPv6 for Research Test Bed. 2009 International Conference on Electrical Engineering and Informatics, Selangor, Malaysia.

[30]. Caicedo, C.E., Joshi, J.B.D., & Tuladhar, S.R.(2009). IPv6 Security Challenge. Computer, 42, 36-42.

[31]. Dr. Yogesh Kumar Sharma and Dr. Surender (2013), "Future Role of Zigbee Technology in Wireless Communication System", Paper published in Grip - The Standard Research International Referred Online Research Journal, ISSN-2278-8123, Issue No. XVI, Pp. 18-31.

[32]. Davies, J. (2003). Understanding IPv6, Microsoft Press.

[33]. Kanda, M. (2004). IPsec: a basis for IPv6 security. [Online]. Available:http://www.ipv6style.jp/en/tech/20040707/index.shtml.

[34]. Radwan, A.M. (2005). Using IPSec in IPv6 Security. [Online]. Available:http://www.uop.edu.jo/csit2006/vol2%20pdf/pg471.pdf

[35]. Saito, Y. (December 2003). IPv6 and New Security Paradigm. NTT Communications, Doc. No. 79

[36]. Dr.Yogesh Kumar Sharma, Rokade Monika D, Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic, National Conference on "Recent Innovations in Engineering and Technology" MOMENTUM-19 63-69 Sharadchandra Pawar College of Engineering, Dumbarwadi, Tal-Junnar, Dist-Pune-410504

[37]. Cisco Systems Report (2004). IPv6 SECURITY Session Sec-2003.

[38]. Seth, B., Dalal, S., Le, D. N., Jaglan, V., Dahiya, N., Agrawal, A., & Verma, K. D. (2021). Secure Cloud Data Storage System Using Hybrid Paillier-Blowfish Algorithm. CMC-Computers Materials & Continua, 67(1), 779-798.

[39]. Seth, B., Dalal, S., Jaglan, V., Le, D. N., Mohan, S., & Srivastava, G. (2020). Integrating encryption techniques for secure data storage in the cloud. Transactions on Emerging Telecommunications Technologies, e4108

[40]. Oh, H., Chae, K., Bang, H., & Na, J. (February 2006). Comparisons analysis of Security Vulnerabilities for Security Enforcement in IPv4/IPv6. Advanced Communication Technology, 2006. ICACT 2006.