**IPv4/IPv6 Coexistence**

http://www.ciscopress.com/articles/article.asp?p=2104947

The transition from IPv4 to IPv6 will not be something that is done overnight; it will take a number of years before IPv6 has anywhere near 100 percent implementation. In these intervening years (including now), a number of mechanisms have been (and will be) developed to make the transition as easy as possible.

The first of the available options is referred to as dual stack. When using this method, an organization essentially does not transition to IPv6 but simply builds a parallel IPv6 network next to their existing IPv4 network.

The second of the available options is tunneling. The basic idea behind tunneling methods is that IPv6 will be tunneled over an existing IPv4 network. A number of different tunneling methods are available and can be selected based on the requirements of the situation.

The third of the available options is translation. The idea behind translation is that at a boundary router between an IPv4 and an IPv6 network a translation process maps an IPv4 address to an IPv6 address (or vice versa).

**Dual Stack**

When a network is configured as dual stack, each device on the network is configured with both an IPv4 address and an IPv6 address, the idea being that once all the devices have implemented IPv6, the IPv4 part of the network will be depreciated. This method is common for businesses looking to slowly convert their existing devices from IPv4 to IPv6. These companies can configure their routing infrastructure to support both IPv4 and IPv6 but bring their other network devices over to IPv6 at a slower pace.

It is also possible for individual devices to be configured as dual stack and use one of the tunneling technologies discussed in the next section.

**Tunneling**

The concept behind tunneling is not new; many people use tunneling daily, but just use it for other reasons. For example, many companies use IPsec or Secure Sockets Layer (SSL) tunnels to secure information when it is being transmitted over an untrusted network.

Many different tunneling methods are available. Which one to use depends on the specific implementation details. Table 1 lists some commonly available tunneling methods and their suggested usage.

|  |  |
| --- | --- |
| **Tunneling Method** | **Suggested Usage** |
| Manual | Used to provide a point-to-point IPv6 link over an existing IPv4 network; only supports IPv6 traffic. |
| GRE | Used to provide a point-to-point IPv6 link over an existing IPV4 network; supports multiple protocols, including IPv6. |
| 6to4 | Used to provide a point-to-multipoint IPv6 link over an existing IPv4 network; sites must use IPv6 addresses from the 2002::/16 range. |
| 6rd (or 6RD)  ( **rapid deployment)** | Used to provide a point-to-multipoint IPv6 link over an existing IPv4 network; sites can use IPv6 addresses from any range. |
| ISATAP ( **Intra-Site Automatic Tunnel Addressing Protocol)** | Used to provide point-to-multipoint IPv6 links over an existing IPv4 network. Designed to be used between devices inside the same site. |

**Translation**

The concept of address translation is also not a new concept to most network engineers; this is because Network Address Translation (NAT) is implemented between different IPv4 networks in almost every residential household. The concept behind this type of NAT and the newer technologies that support address translation between IPv4 and IPv6 networks is similar. IPv6 translation technologies differ from IPv6 tunneling technologies; this is because the translation technologies enable IPv4-only devices to speak to IPv6-only devices, which is not possible with any of the tunneling methods.

However, IPv4/IPv6 translation and IPv4-only translation entail a certain amount of complexity. What happens when an IPv6-only device is attempting to communicate with a device on the public IPv4 Internet and only an IPv4 DNS record (A) exists? In these situations, a secondary technology is required to step in and provide additional services for the connection to work.

The first method to be introduced to provide IPv6 translation services was Network Address Translation - Protocol Translation (NAT-PT). NAT-PT defined a mechanism to not only translate between IPv4 to IPv6 addresses but also a built-in ability to provide protocol translation services for Internet Control Message Protocol (ICMP), File Transfer Protocol (FTP), and Domain Name System (DNS). The component that was responsible for these translation services is called the application layer gateway (ALG).

The ALG piece of the NAT-PT method raised a number of issues. With additional testing and real-life experience, a new method was introduced that separated the address translation functionality and the application layer translation functionalities: NAT64 and DNS64.

DNS64 can synthesize IPv6 address resource records (AAAA) from IPv4 resource records (A); it does this by encoding the returned IPv4 address into a IPv6 address format.

**Summary**

The selection of an IPv6 transition mechanism depends greatly on the current status of an organization’s network and how fast they want to transition their devices from IPv4 to IPv6. Logic seems to say that those organizations with bleeding-edge technology tastes and small staffs will probably be (or are already) the first people in line to transition over to IPv6. Those larger companies that have tens of thousands of network devices will most likely transition a piece at a time following the experience level of each department.

The transition to IPv6 is coming, and all those network engineers reading this article should become experts in IPv6 as quickly as possible. The process of converting networks from IPv4 to IPv6 will shortly become a large-scale request, and those with the correct skills will be in demand, a fact even more important in the current economy.

Next up, [IPv6 Tunneling Technology Configuration](http://www.ciscopress.com/articles/article.aspx?p=2104948).

**Tunneling Configuration Options**

http://www.ciscopress.com/articles/article.asp?p=2104948

The configuration of each of these options ranges from simple to rather complex. Each of the following sections takes a look at the steps required to configure these options in a basic environment.

**Manual IPv6 Tunnels**

The configuration of manual IPv6 tunnels is one of the simplest to perform because these types of tunnel are limited to a single source and destination. Table 1 shows the steps required to configure a manual IPv6 tunnel.

**Table 1- Manual IPv6 Tunnel Configuration**

|  |  |  |
| --- | --- | --- |
| 1 | Enter global configuration mode. | router#**configure terminal** |
| 2 | Create a tunnel interface and enter interface configuration mode. | router(config)#**interface tunnel** *tunnel-num* |
| 3 | Assign an IPv6 address that will be used to communicate between each tunnel endpoint. Note that only two addresses are required. | router(config-if)#**ipv6 address** *ipv6-prefix / prefix-len* **[eui-64]** |
| 4 | Define the source for the tunnel; this can either be an IPv4 address (or a hostname that resolves to one) or an interface that has already been configured with an IPv4 address. | router(config-if)#**tunnel source** *source-ip|source-interface* |
| 5 | Define the destination for the tunnel; this can only be an IPv4 address (or a hostname that resolves to one). | router(config-if)#**tunnel destination** *destination-ip* |
| 6 | Configure the tunnel mode type. | router(config-if)#**tunnel mode ipv6ip** |

**GRE IPv6 Tunnels**

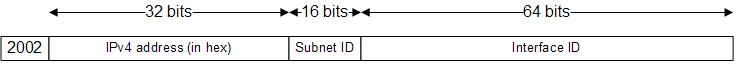
The configuration of an IPv6 generic routing encapsulation (GRE) tunnel is similar to that of a manual IPv6 tunnel. Table 2 shows the steps required to configure a GRE IPv6 tunnel.

**Table 2 - GRE IPv6 Tunnel Configuration**

|  |  |  |
| --- | --- | --- |
| 1 | Enter global configuration mode. | router#**configure terminal** |
| 2 | Create a tunnel interface and enter interface configuration mode. | router(config)#**interface tunnel** *tunnel-num* |
| 3 | Assign an IPv6 address that will be used to communicate between each tunnel endpoint. Note that only two addresses are required. | router(config-if)#**ipv6 address** *ipv6-prefix / prefix-len* [**eui-64**] |
| 4 | Define the source for the tunnel. This can either be an IPv4 address (or a hostname that resolves to one) or an interface that has already been configured with an IPv4 address. | router(config-if)#**tunnel source** *source-ip|source-interface* |
| 5 | Define the destination for the tunnel. This can only be an IPv4 address (or a hostname that resolves to one). | router(config-if)#**tunnel destination** *destination-ip* |
| 6 | Configure the tunnel mode type. | router(config-if)#**tunnel mode gre ip** |

**6to4 Tunnels**

With 6to4 tunnels, the configuration differs slightly because the tunnel is not point to point but point to multipoint. 6to4 tunnels are also limited in some ways because each of the IPv6 sites must use addresses from the 2002::/16 prefix range. The configuration of 6to4 tunnels also does not require the configuration of a tunnel destination. This is because the destination IPv4 address is included as part of the 6to4 address. Figure 1 shows how the IPv4 address of the 6to4 border router (the device that touches both the IPv4 and IPv6 networks) includes the IPv4 address within the 6to4 IPv6 address.

[](javascript:popUp('/content/images/art_wilkins_ipv6tuntechconfig/elementLinks/wilkins_ipv6_fig01.png'))

[*Figure 1*](javascript:popUp('/content/images/art_wilkins_ipv6tuntechconfig/elementLinks/wilkins_ipv6_fig01.png')) *6to4 Address Format*

Table 3 shows the steps required to configure 6to4 IPv6 tunnels.

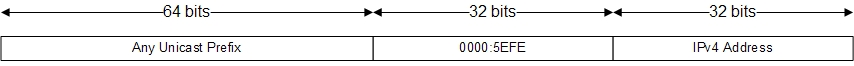
**Table 3 - 6to4 Tunnel Configuration**

|  |  |  |
| --- | --- | --- |
| 1 | Enter global configuration mode. | router#**configure terminal** |
| 2 | Create a tunnel interface and enter interface configuration mode. | router(config)#**interface tunnel** *tunnel-num* |
| 3 | Assign an IPv6 address that will be used to communicate between each tunnel endpoint. | router(config-if)#**ipv6 address** *ipv6-prefix / prefix-len* **[eui-64]** |
| 4 | Define the source for the tunnel. This can either be an IPv4 address (or a hostname that resolves to one) or an interface that has already been configured with an IPv4 address. | router(config-if)#**tunnel source** *source-ip|source-interface* |
| 5 | Configure the tunnel mode type. | router(config-if)#**tunnel mode ipv6ip 6to4** |
| 6 | Configure a IPv6 route for the 2002::/16 6to4 address range. | router(config-if)#**ipv6 route 2002::/16 tunnel** *tunnel-num* |

**ISATAP Tunnels**

The configuration of ISATAP tunnels is similar to 6to4 but is intended for a completely different purpose. ISATAP tunnels are intended to be used *inside* a site, not between them. A good example of this is when only specific machines on an otherwise IPv4 network run IPv6. ISATAP would allow these devices to communicate via IPv6 over the IPv4 network. Each of the routers that connect to segments where these clients exist would need to be configured for ISATAP. They would then advertise the configured ISATAP prefix so that clients could automatically configure themselves.

Like 6to4, ISATAP includes the IPv4 address as part of the IPv6 address. With ISATAP, this is done as part of the interface ID. Figure 2 shows how the IPv4 address of the ISATAP border router includes the iPv4 address within the ISATAP IPv6 address.

[](javascript:popUp('/content/images/art_wilkins_ipv6tuntechconfig/elementLinks/wilkins_ipv6_fig02.png'))

[*Figure 2*](javascript:popUp('/content/images/art_wilkins_ipv6tuntechconfig/elementLinks/wilkins_ipv6_fig02.png')) *ISATAP Address Format*

Table 4 shows the steps required to configure ISATAP IPv6 tunnels.

**Table 4 - ISATAP Tunnel Configuration**

|  |  |  |
| --- | --- | --- |
| 1 | Enter global configuration mode. | router#**configure terminal** |
| 2 | Create a tunnel interface and enter interface configuration mode. | router(config)#**interface tunnel** *tunnel-num* |
| 3 | Assign an IPv6 address that will be used to communicate between each tunnel endpoint. Configuring using the **eui-64** parameter will automatically configure the IPv6 address with the tunnel source IPv4 address. | router(config-if)#**ipv6 address** *ipv6-prefix / prefix-len* [**eui-64**] |
| 4 | Define the source for the tunnel. This can either be an IPv4 address (or a hostname that resolves to one) or an interface that has already been configured with an IPv4 address. | router(config-if)#**tunnel source** *source-ip|source-interface* |
| 5 | Configure the tunnel mode type. | router(config-if)#**tunnel mode ipv6ip isatap** |
| 6 | Enable IPv6 router advertisements. This enables clients to automatically configure an IPv6 address from the prefix configured on the router. | router(config-if)#**no ipv6 nd ra suppress** |

**Summary**

The configuration of IPv6 tunneling technologies can be as easy as a point-to-point link and as hard as configuring networkwide IPV6 addresses linked to IPv4 addresses. This article reviewed the basic steps required to get these technologies up and running. Take the time to test these technologies in a lab environment if possible to truly understand how they work and to ensure that they work the way you want.

This article intentionally left out the configuration for 6rd tunnels. Although these tunnels are certainly going to replace some of the current IPv6 tunneling implementations, it also has just become a supported feature on Cisco devices. A supporting article may be drafted in the future once supported hardware becomes available to the author.