One strategy for transitioning an existing IPv4 topology to IPv6 is to deploy IPv6 in isolated domains while maintaining an IPv4 infrastructure. This strategy allows you to deploy IPv6 incrementally with minimal disruption to the IPv4 infrastructure. To support this strategy, HP in turn supports the following transition mechanisms:

- Dual stack backbone.
- IPv6 over IPv4 tunnels.

This chapter describes these two transition mechanisms and explains how to configure IPv6 over IPv4 tunnels, clear IPv6 tunnel statistics, and display IPv6 tunnel information.

**Dual Stack Backbone**

*Dual stack backbone* is a IPv4-to-IPv6 transition strategy that requires backbone routers and end systems to run both IPv4 and IPv6 protocol stacks. If you implement a dual stack backbone, you should be able to enable IPv4 and IPv6 routing protocols and all other features, as you would if you were running only one protocol stack, without any limitations. However, before implementing this strategy, you must consider the following:

- You must make certain that each backbone router has enough memory to handle IPv4 and IPv6 forwarding processes, routing protocols, and route tables.
- You must define and maintain IPv4 and IPv6 address schemes in your topology.
- You must configure, maintain, and manage IPv4 and IPv6 routing protocols.

This section provides information about the following topics:

- End system dual stack operation.
- Backbone router dual stack operation.

**End System Dual Stack Operation**

When you configure an end system to run dual stacks, the following categories of applications can coexist on the end system:

- Existing applications that support the IPv4 stack and are not upgraded to support the IPv6 stack.
- Existing applications that support the IPv4 stack and are upgraded to support the IPv6 stack.
- Newly installed applications that support the IPv4 and IPv6 stacks.

Figure 8.1 shows how the various applications on an end system function with dual stacks. Non-upgraded applications use the IPv4 stack only, while upgraded and new applications use both IPv4 and IPv6 stacks.
If an application supports both IPv4 and IPv6 stacks, the application decides which address, and thereby, which protocol to use. Generally, IPv6 is chosen by default. After the application decides on the address, it connects the source node to the destination using the chosen address. In Figure 8.2, the application decides to use the IPv6 address.

Some applications that support both IPv4 and IPv6 stacks, such as Telnet, SSH, and ping, allow you to explicitly specify either an IPv4 or IPv6 address. For more information about these applications, see “Managing an HP Device Over IPv6” on page 12-1.

**Backbone Router Dual Stack Operation**

To implement a dual stack backbone, you must configure each backbone router to run both IPv4 and IPv6 protocol stacks. When implemented, IPv4 communication occurs using the IPv4 protocol stack and IPv4 packets are routed using IPv4 routing protocols. Likewise, IPv6 communication occurs using the IPv6 protocol stack, and IPv6 packets are routed using IPv6 routing protocols.

In other words, IPv4 and IPv6 forwarding processes, routing protocols, and route tables run in parallel to one another, which also means that they are separate and independent of one another. For example, you cannot redistribute the routes learned by an IPv4 routing protocol into an IPv6 routing protocol, or vice versa. The only situation in which the protocols intermingle is when tunneling IPv6 traffic over IPv4 tunnels (IPv6 packets are encapsulated within an IPv4 packet). For more information, see “IPv6 Over IPv4 Tunnels” next.

**IPv6 Over IPv4 Tunnels**

To enable communication between the isolated IPv6 domains using the IPv4 infrastructure, you can configure IPv6 over IPv4 tunnels. As shown in Figure 8.3, these tunnels encapsulate an IPv6 packet within an IPv4 packet.
HP supports the following IPv6 over IPv4 tunneling mechanisms:

- Manually configured tunnels
- Automatic 6to4 tunnels
- Automatic IPv4-compatible IPv6 tunnels

In general, a manually configured tunnel establishes a permanent link between routers in IPv6 domains, while the automatic tunnels establish a transient link that is created and taken down on an as-needed basis. (Although the feature name and description may imply otherwise, some configuration is necessary to set up an automatic tunnel.) Also, a manually configured tunnel has explicitly configured IPv4 addresses for the tunnel source and destination, while the automatic tunnels have an explicitly configured IPv4 address for the tunnel source and an automatically generated address for the tunnel destination.

**NOTE:** HP's implementation of IPv6 supports automatic IPv4-compatible IPv6 tunnels. However, because of this tunneling mechanism's inherent dependence on IPv4 addresses, which diminishes the benefits of IPv6, HP recommends using either manually configured tunnels or automatic 6to4 tunnels instead.

These tunneling mechanisms require that the router at each end of the tunnel run both IPv4 and IPv6 protocol stacks. (For information about configuring IPv4 and IPv6 protocol stacks on a router interface, see “Configuring IPv4 and IPv6 Protocol Stacks” on page 3-4.) The routers running both protocol stacks, or **dual-stack routers**, can interoperate directly with both IPv4 and IPv6 end systems and routers.

### Configuring a Manual IPv6 Tunnel

You can use a manually configured tunnel to connect two isolated IPv6 domains. You should deploy this point-to-point tunnel mechanism if you need a permanent and stable connection.

To configure a manual IPv6 tunnel, enter commands such as the following on a Routing Switch running both IPv4 and IPv6 protocol stacks on each end of the tunnel:

```
ProCurveRS(config)# interface tunnel 1
ProCurveRS(config-tnif-1)#ipv6 address 2001:b78:384d:34::/64 eui-64
ProCurveRS(config-tnif-1)#tunnel source ethernet 3/1
ProCurveRS(config-tnif-1)#tunnel destination 198.162.100.1
ProCurveRS(config-tnif-1)#tunnel mode ipv6ip
```

This example creates tunnel interface 1 and assigns a global IPv6 address with an automatically computed EUI-64 interface ID to it. The IPv4 address assigned to Ethernet interface 3/1 is used as the tunnel source, while the IPv4 address 192.168.100.1 is configured as the tunnel destination. Finally, the tunnel mode is specified as a manual IPv6 tunnel.

**Syntax:** interface tunnel <number>

For the <number> parameter, specify a value between 1 – 32.

**Syntax:** ipv6 address <ipv6-prefix>/<prefix-length> [eui-64]

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.
You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

**Syntax:** tunnel source <ipv4-address> | ethernet <port> | loopback <number> | tunnel <number> | ve <number>

You must specify the <ipv4-address> parameter using 8-bit values in dotted decimal notation.

The Ethernet | loopback | tunnel | ve parameter specifies an interface as the tunnel source. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, VE, or tunnel interface, also specify the loopback, VE, or tunnel number, respectively.

**Syntax:** tunnel destination <ipv4-address>

You must specify the <ipv4-address> parameter using 8-bit values in dotted decimal notation.

**Syntax:** tunnel mode ipv6ip

### Configuring an Automatic 6to4 Tunnel

An automatic 6to4 tunnel establishes a transient link between IPv6 domains, which are connected by an IPv4 backbone. When needed, a device on which an automatic 6to4 tunnel is configured in one domain can establish a tunnel with another similarly configured device in another domain. When no longer needed, the devices take down the tunnel.

Instead of a manually configured tunnel destination, an automatic 6to4 tunnel constructs a globally unique 6to4 prefix, which determines the tunnel destination. The 6to4 prefix has the following format:

2002:<ipv4-address>:/48

When two domains need to communicate, a device creates a tunnel using the 6to4 prefix. The software automatically generates the 6to4 prefix by concatenating a configured static IPv6 prefix of 2002 with the destination device's globally unique IPv4 address. (Each device in an IPv6 domain that needs to communicate over an automatic 6to4 tunnel must have one globally unique IPv4 address, from which the globally unique 6to4 prefix is constructed.) After the communication ends, the tunnel is taken down.

To configure an automatic 6to4 tunnel, enter commands such as the following on a device interface on each end of the tunnel. The devices at each end of the tunnel must run the IPv4 and IPv6 protocol stacks.

```
ProCurveRS(config)# interface tunnel 1
ProCurveRS(config-tunif-1)# ipv6 address 2001:b78:384d:34::/64 eui-64
ProCurveRS(config-tunif-1)# tunnel source ethernet 3/1
ProCurveRS(config-tunif-1)# tunnel mode ipv6ip 6to4
ProCurveRS(config-tunif-1)# exit
ProCurveRS(config)# ipv6 route 2002::/16 tunnel 1
```

This example creates tunnel interface 1 and assigns a global IPv6 address with an automatically computed EUI-64 interface ID to it. The IPv4 address assigned to Ethernet interface 3/1 is used as the tunnel source. A static IPv6 prefix (2002::/16), which is used in the construction of a globally unique 6to4 prefix, is configured for the tunnel interface. Finally, the tunnel mode is specified as an IPv6 tunnel using a 6to4 prefix.

**Syntax:** interface tunnel <number>

For the <number> parameter, specify a value between 1 – 32.

**Syntax:** ipv6 address <ipv6-prefix>/<prefix-length> [eui-64]

You must specify the <ipv6-prefix> parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the <prefix-length> parameter as a decimal value. A slash mark (/) must follow the <ipv6-prefix> parameter and precede the <prefix-length> parameter.

The eui-64 keyword configures the global or site-local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

**Syntax:** tunnel source <ipv4-address> | ethernet <port> | loopback <number> | tunnel <number> | ve <number>
You must specify the \texttt{<ipv4-address>} parameter using 8-bit values in dotted decimal notation.

The \texttt{ethernet} \texttt{loopback} \texttt{tunnel} \texttt{ve} parameter specifies an interface as a tunnel source. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, VE, or tunnel interface, also specify the loopback, VE, or tunnel number, respectively.

\textbf{Syntax:} \texttt{tunnel mode ipv6ip 6to4}

\textbf{Syntax:} \texttt{ipv6 route 2002::/16 tunnel <number>}

For the \texttt{<number>} parameter, you must specify the same tunnel number, which was created using the \texttt{interface tunnel} \texttt{<number>} command.

\section*{Configuring an Automatic IPv4-Compatible IPv6 Tunnel}

An IPv4-compatible IPv6 tunnel establishes a transient link between two IPv6 domains over an IPv4 backbone. When needed, a device on which an IPv4-compatible IPv6 tunnel is configured in one domain can establish a tunnel with another similarly configured device in another domain. When no longer needed, the devices take down the tunnel.

Instead of a manually configured tunnel destination, this tunnel constructs an IPv4-compatible IPv6 address, which determines the tunnel destination. When two domains need to communicate, a router creates a tunnel using the IPv4-compatible IPv6 address. The software automatically generates the IPv6 address by concatenating zeros in the high-order 96 bits and the device's globally unique IPv4 address in the low-order 32 bits (for example, 0:0:0:0:0:192.168.100.1). (Each router in an IPv6 domain that needs to communicate over an automatic IPv4-compatible IPv6 tunnel must have one globally unique IPv4 address, from which the globally unique IPv4-compatible IPv6 address is constructed.) After the communication ends, the tunnel is taken down.

\textbf{NOTE:} HP's implementation of IPv6 supports automatic IPv4-compatible IPv6 tunnels. However, because of this tunneling mechanism's inherent dependence on IPv4 addresses, which diminishes the benefits of IPv6, HP recommends using either manually configured tunnels or automatic 6to4 tunnels instead.

To configure an automatic IPv4-compatible IPv6 tunnel, enter commands such as the following on the device at each end of the tunnel. The devices at each end of the tunnel must run IPv4 and IPv6 protocol stacks.

\begin{verbatim}
ProCurveRS(config)# interface tunnel 1
ProCurveRS(config-tunl-1)#tunnel source ethernet 3/1
ProCurveRS(config-tunl-1)#tunnel mode ipv6ip auto-tunnel
\end{verbatim}

This example creates tunnel interface 1. The IPv4 address assigned to Ethernet interface 3/1 is used as the tunnel source. Specifying the tunnel mode as \texttt{ipv6ip auto-tunnel} sets up an IPv4-compatible tunnel using an IPv4-compatible IPv6 address.

\textbf{Syntax:} \texttt{interface tunnel <number>}

For the \texttt{<number>} parameter, specify a value between 1 – 32.

\textbf{Syntax:} \texttt{tunnel source \textlangle ipv4-address\textrangle} \texttt{ethernet} \texttt{\textlangle port\textrangle} \texttt{loopback} \texttt{\textlangle number\textrangle} \texttt{tunnel} \texttt{\textlangle number\textrangle} \texttt{ve} \texttt{\textlangle number\textrangle}

You must specify the \texttt{<ipv4-address>} parameter using 8-bit values in dotted decimal notation.

The \texttt{ethernet} \texttt{loopback} \texttt{tunnel} \texttt{ve} parameter specifies an interface as the tunnel source. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, VE, or tunnel interface, also specify the loopback, VE, or tunnel number, respectively.

\textbf{Syntax:} \texttt{tunnel mode ipv6ip auto-tunnel}

\section*{Clearing IPv6 Tunnel Statistics}

You can clear all IPv6 tunnel statistics (reset all fields to zero) or statistics for a specified tunnel interface.

For example, to clear statistics for tunnel 1, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI:

\begin{verbatim}
ProCurveRS# clear ipv6 tunnel 1
\end{verbatim}
**Syntax:** clear ipv6 tunnel <number>

The `<number>` parameter specifies the tunnel number.

**Displaying IPv6 Tunnel Information**

To display a summary of tunnel information, enter the following command at any level of the CLI:

```
ProCurveRS# show ipv6 tunnel
IP6 Tunnels
Tunnel Mode Packet Received Packet Sent
  1 configured 0 0
  2 configured 0 22419
  6 6to4 0 0
```

**Syntax:** show ipv6 tunnel

This display shows the following information.

<table>
<thead>
<tr>
<th>This Field...</th>
<th>Displays...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnel</td>
<td>The tunnel interface number.</td>
</tr>
<tr>
<td>Mode</td>
<td>The tunnel mode. Possible modes include the following:</td>
</tr>
<tr>
<td></td>
<td>• configured – Indicates a manually configured tunnel.</td>
</tr>
<tr>
<td></td>
<td>• 6to4 – Indicates an automatic 6to4 tunnel.</td>
</tr>
<tr>
<td></td>
<td>• auto – Indicates an automatic IPv4-compatible tunnel.</td>
</tr>
<tr>
<td>Packet Received</td>
<td>The number of packets received by a tunnel interface.</td>
</tr>
<tr>
<td>Packet Sent</td>
<td>The number of packets sent by a tunnel interface.</td>
</tr>
</tbody>
</table>

**Displaying Tunnel Interface Information**

For example, to display status and configuration information for tunnel interface 1, enter the following command at any level of the CLI:

```
ProCurveRS# show interfaces tunnel 1
Tunnel is up, line protocol is up
    Hardware is Tunnel
    Tunnel source ethernet 3/5
    Tunnel destination is not configured
    Tunnel mode ipv6ip auto-tunnel
    No port name
    MTU 1500 bytes
```

**Syntax:** show interfaces tunnel <number>

The `<number>` parameter indicates the tunnel interface number for which you want to display information.
This display shows the following information.

<table>
<thead>
<tr>
<th>This Field...</th>
<th>Displays...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnel interface status</td>
<td>The status of the tunnel interface can be one of the following:</td>
</tr>
<tr>
<td></td>
<td>• up – The tunnel interface is functioning properly.</td>
</tr>
<tr>
<td></td>
<td>• down – The tunnel interface is not functioning and is down.</td>
</tr>
<tr>
<td>Line protocol status</td>
<td>The status of the line protocol can be one of the following:</td>
</tr>
<tr>
<td></td>
<td>• up – The line protocol is functioning properly.</td>
</tr>
<tr>
<td></td>
<td>• down – The line protocol is not functioning and is down.</td>
</tr>
<tr>
<td>Hardware is tunnel</td>
<td>The interface is a tunnel interface.</td>
</tr>
<tr>
<td>Tunnel source</td>
<td>The tunnel source can be one of the following:</td>
</tr>
<tr>
<td></td>
<td>• An IPv4 address</td>
</tr>
<tr>
<td></td>
<td>• The IPv4 address associated with an interface/port.</td>
</tr>
<tr>
<td>Tunnel destination</td>
<td>The tunnel destination can an IPv4 address.</td>
</tr>
<tr>
<td>Tunnel mode</td>
<td>The tunnel mode can be one the following:</td>
</tr>
<tr>
<td></td>
<td>• ipv6ip auto-tunnel – Indicates an automatic IPv4-compatible tunnel.</td>
</tr>
<tr>
<td></td>
<td>• ipv6ip 6to4 – Indicates an automatic 6to4 tunnel.</td>
</tr>
<tr>
<td>Port name</td>
<td>The port name configured for the tunnel interface.</td>
</tr>
<tr>
<td>MTU</td>
<td>The setting of the IPv6 maximum transmission unit (MTU).</td>
</tr>
</tbody>
</table>