HP-UX IPv6 Transport Administrator's Guide

HP-UX 11i v3

Manufacturing Part Number : B2355-91068
Feburary 2007

United States

Legal Notices

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Warranty

A copy of the specific warranty terms applicable to your Hewlett- Packard product and replacement parts can be obtained from your local Sales and Service Office.

U.S. Government License

Copyright Notice

Trademark Notices

UNIX® is a registered trademark of The Open Group.

Intel® and Itanium® are registered trademarks of Intel Corporation.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

Printed in the US.

Contents

Contents

Contents

About This Document

This document describes how to install, configure, and troubleshoot HP-UX 11i v3 IPv6 transport software.

The document printing date and part number indicate the document's current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The document part number will change when extensive changes are made.

Document updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

The latest version of this document can be found on line at:
`docs.hp.com/hpux/netcom/index.html#IPv6`.

Intended Audience

This document is intended for system and network administrators responsible for installing, configuring, and managing IPv6 transport. Administrators are expected to have knowledge of Transmission Control Protocol/Internet Protocol (TCP/IP) networking concepts and network configuration. As well it is helpful to have knowledge of operating system concepts, commands, and configuration.

This document is not a tutorial.

New and Changed Documentation in This Edition

Overall, the document has been updated to reflect new features and changes starting with the HP-UX 11i v3 release of IPv6 transport. It also continues to provide relevant HP-UX 11i v2 IPv6 transport information that is not changed with HP-UX 11i v2 September 2004, but that has been present since the initial HP-UX 11i v2 release (July 2003).

Publishing History

Table 1　　Publishing History Details

Document Manufacturing Part Number	Operating Systems Supported	Publication Date
B2355-91068	11i v3	Feburary 2007
B2355-90795	11i v2	July 2003

What Is in This Document

This manual provides information for administering HP-UX 11i v3 IPv6 transport software. HP-UX 11i v3 IPv6 transport software uses the next generation Internet Protocol (IPv6) to connect HP-UX Servers and Workstations with other systems running IPv4 or IPv6 over IEEE 802.3, Ethernet or FDDI Local Area Networks. An IPv6 for HP-UX 11i v3 network can extend over routers into a Wide Area Network.

This manual is organized as follows:

Chapter 1　**Features Overview** provides a summary and overview of IPv6 features offered in HP-UX 11i v3.

Chapter 2　**Configuration** describes how to automatically or manually configure HP-UX 11i v3 IPv6 transport.

Chapter 3　**Troubleshooting** provides flowcharts to help diagnose HP-UX 11i v3 IPv6 software problems.

Chapter 4　**Utilities** describes useful tools for configuring, and maintaining HP-UX 11i v3 IPv6 software.

Chapter 5　**IPv6 Addressing and Concepts** describes IPv6 addressing and provides some basic IPv6 networking terminology.

Chapter 6　**IPv6 Software and Interface Technology** discusses IPv6 deployment and migration.

Appendix A　**IPv6 ndd Tunable Parameters** provides a list of supported `ndd` IPv6 tunable parameters that allow for advanced performance tuning.

If you are unfamiliar with IPv6 networking concepts, refer to Chapter 4, "IPv6 Addressing and Concepts," on page 39 and Chapter 5, "IPv6 Software and Interface Technology," on page 53, before configuring IPv6 interfaces.

HP-UX Release Name and Release Identifier

Each HP-UX 11i release has an associated release name and release identifier. The *uname* (1) command with the -r option returns the release identifier. This table shows the releases available for HP-UX 11i.

Table 2 HP-UX 11i Releases

Release Identifier	Release Name	Supported Processor Architecture
B.11.31	HP-UX 11i v3	Intel® Itanium®
B.11.23	HP-UX 11i v2	Intel® Itanium®
B.11.22	HP-UX 11i v1.6	Intel® Itanium®
B.11.20	HP-UX 11i v1.5	Intel® Itanium®
B.11.11	HP-UX 11i v1	PA-RISC

Related Documents

HP Documentation

Additional information about HP-UX 11i v3 IPv6 can be found within *docs.hp.com* in the *networking and communications* collection under *IPv6* at:

`http://www.docs.hp.com/hpux/netcom/index.html#IPv6`

Other documents in this collection (besides this guide) include:

HP-UX IPv6 Porting Guide

HP-UX IPv6 Transition Mechanisms (White Paper)

Related RFCs

As well, the IETF (Internet Engineering Task Force) RFCs listed below can be located at: `http://www.ietf.org/rfc.html`

Table 3 IPv6 RFCs Supported

RFCs	Description
RFC 1981	Path MTU Discovery for IPv6
*RFC 2292	Advanced Sockets API for IPv6
RFC 2373	IPv6 Addressing Architecture
RFC 2374	IPv6 Aggregatable Global Unicast Address Format
RFC 2375	IPv6 Multicast Address Assignments
RFC 2452	IPv6 MIB for TCP
RFC 2454	IPv6 MIB for UDP
RFC 2460	IPv6 Specification
RFC 2461	Neighbor Discovery for IPv6
RFC 2462	IPv6 Stateless Address Autoconfiguration
RFC 2463	ICMPv6 for IPv6 Specification
RFC 2464	Transmission of IPv6 Packets over Ethernet Networks
RFC 2465	MIB for IPv6: Textual Conventions and General Group
RFC 2466	MIB for IPv6: ICMPv6 Group
RFC 2467	Transmission of IPv6 Packets over FDDI Networks
RFC 2473	Generic Packet Tunneling in IPv6 Specification
RFC 2553	Basic Socket Interface Extensions for IPv6
RFC 2710	Multicast Listener Discovery (MLD) for IPv6 (Host Part)
RFC 2893	Transition Mechanisms for IPv6 Hosts and Routers
RFC 3019	IP Version 6 Management Inofrmation Base for The Multicast Listener Discovery Protocol
RFC 3056	Connection of IPv6 Domains via IPv4 Clouds (6to4)

- * Advanced Socket API features from RFC 2292bis, such as Routing Header, Hop-by-Hop, and Destination Option processing are also supported. The `inet6_rth_*()` and `inet6_opt_*()` functions provide these features. The `inet6_rth_*()` and `inet6_opt_*()` functions are in `/usr/lib/libipv6.1`.

HP Welcomes Your Comments

HP welcomes your comments concerning this document. HP is committed to providing documentation that meets your needs.

Please send comments to: netinfo_feedback@cup.hp.com

Please include document title, manufacturing part number, and any comment, error found, or suggestion for improvement you have concerning this document. Also, please include what we did right so we can incorporate it into other documents.

1 Features Overview

This chapter summarizes the features for HP-UX 11i v3 and an overview of HP-UX 11i v3 IPv6 transport functionality.

IPv6 Transport

IPv6 is the next generation Internet Protocol. The IPv6 protocol is also referred to as "IPng" (IP next generation). It provides the infrastructure for the next wave of Internet devices, such as PDAs, mobile phones and appliances; it also provides greater connectivity for existing devices such as laptop computers.

IPv6 was designed by the Internet Engineering Task Force (IETF) to improve upon the scalability, security, ease of configuration, and network management capabilities of IPv4. HP-UX 11i v3 IPv6 network transport software provides host support for IPv6.

NOTE This guide focuses on IPv6 transport, but be aware that IPv6 is also supported on the following HP-UX 11i v3 components: Internet Services, DCE, DLPI, FDDI, SAM-NNC, Libc, Commands, Desktop (CDE), X11R6-based applications, C2 Audit, EMS, Online Diagnostics, SNMP, nettl, IPSec, Kerberos Client, Service Guard, Glance, HP-UX Secure Shell, Apache, and JVM. Refer to product-specific documentation for more information.

IPv6 Transport Features Available with HP-UX 11i v3

This section describes IPv6 transport features available with HP-UX 11i v3.

- **netstat Enhanced to Support the Display of 64-bit MIB Counters:** `netstat` in HP-UX 11i v3 (for IPv4 and IPv6) supports the display of 64-bit MIB (Management Information Base) counters. Thus, some of the `netstat` fields have the potential to display widened output. This can cause a wraparound effect on 80-character displays.
- **IPoIB (IP over InfiniBand) Link Support:** HP-UX 11i v3 provides transport support for InfiniBand links that support IPoIB. The HP-UX 11i v3 transport software support is required for HP InfiniBand links to run IPoIB.

The HP-UX networking utilities `ifconfig`, `netstat`, `lanadmin`, `lanscan`, `arp`, `rtradvd` and `ndp` have all been enhanced to be capable of handling IPoIB-related data. (Note that `rarp` has not been enhanced for handling IPoIB.)

For more information on InfiniBand and the IPoIB protocol, refer to the *HP-UX InfiniBand Support Guide* available at `http://www.docs.hp.com/hpux/netcom/index.html#InfiniBand`.

- **Multicast Listener Discovery (MLD) Support (Host Portion Only):** The host part of Multicast Listener Discovery (MLD) protocol for IPv6 based on RFC 2710 "Multicast Listener Discovery (MLD) for IPv6", is supported. MLD is automatically enabled when an IPv6 interface is initialized. The Management Information Base for MLD, based on RFC 3019, is also supported.

  RFC 2710 specifies the protocol used by an IPv6 router to discover the presence of multicast listeners (that is, nodes wishing to receive multicast packets) on its directly attached links, and to discover specifically, which multicast addresses are of interest to those neighboring nodes. This protocol is referred to as Multicast Listener Discovery or MLD. MLD is derived from version 2 of IPv4's Internet Group Management Protocol, IGMPv2. One important difference to note is that MLD uses ICMPv6 (IP Protocol 58) message types, rather than IGMP (IP Protocol 2) message types.

  For more MLD information refer to RFC 2710, "Multicast Listener Discovery (MLD) for IPv6".

- **Router Advertisement:** Router Functionality as specified in RFC 2461 "Neighbor Discovery for IP Version 6 (IPv6)", is implemented with a daemon, `rtradvd`, and an accompanying configuration file, `/etc/rtradvd.conf`. The `rtradvd` daemon listens to router solicitation and sends router advertisement messages on demand or periodically (as described in RFC 2461). These advertisements allow any listening host to configure their addresses and some other parameters automatically without manual intervention. They can also choose a default router based on these advertisements

  Router advertisement is configured on a per interface basis. Refer to the `rtradvd.conf(4)` man page for more information.

- **IPv6 Transition Mechanism Enhancements:** HP-UX 11i v3 provides several IPv6 transition mechanism changes from those previously offered in base (default) HP-UX 11i v2. Highlights of these changes are provided below. There have been no changes to the dual stack mechanism, but several important changes to the tunneling mechanisms. The following RFCs are supported (the IETF documents listed below are available at http://www.ietf.org):

RFC 2473 - *Packet Tunneling in IPv6*

RFC 2893 - *Transition Mechanisms for IPv6 Hosts and Routers*

RFC 3056 - *Connection of IPv6 Domains via IPv4 Clouds*

IMPORTANT

As a result of supporting RFC 2893, tunnel configuration on HP-UX 11i v3, requires specific changes from tunnel configurations on base (default) HP-UX 11i v2. For detailed information, including specific configuration instructions, refer to relevant sections of Chapter 2, “Configuration,” of this guide.

SAM has not been enhanced to support the tunneling enhancements. HP-UX 11i v3 tunneling configuration must be done by editing the `/etc/rc.config.d/netconf-ipv6` file or by using the `ifconfig` command.

— **Configured tunneling is point-to-point with addresses assigned to tunnel endpoints:** In conformance with RFC 2893 (which obsoletes RFC 1933) configured tunnels are pseudo-interfaces with associated addresses. Previously, when conforming to RFC 1933, tunnels were implemented using special routing entries. The RFC 1933 implementation did not allow addresses to be associated with tunnels and hence, routing protocol daemons were not able to operate over tunnels. To overcome this problem, RFC 2893 specifies tunnels as IPv6 interfaces and requires them to be configured with at least (on primary interfaces) link-local addresses.

  As a result, the process for configuring tunnels using the `ifconfig` and `route` commands and the `/etc/rc.config.d/netconf-ipv6` file is different than it was in base (default) HP-UX 11i v2.

— **HP-UX server can be configured as a router in a point-to-point configured tunnel:** You can configure tunneling between the following network nodes: host->router; host->host; router-> host; and router->router. The HP-UX server can perform the role of the router in the tunnel configuration.

— **HP-UX server can be configured as a "6to4" router:** The HP-UX server can perform the role of a router in a "6to4" configuration. Prior to HP-UX 11i v2 PI, the HP-UX server was only able to perform the role of a host in a "6to4" configuration.

— **IP6-in-IP6 and IP-in-IP6 Support:** Two additional tunneling types are supported, IP6-in-IP6 and IP-in-IP6. IP6-in-IP6 tunnel configuration allows transmission of IPv6 packets encapsulated in an IPv6 header. IP-in-IP6 tunnel configuration allows transmission of IPv4 packets encapsulated in an IPv6 header.

IP6-in-IP tunnel configuration allows transmission of IPv6 packets encapsulated in an IPv4 header. IP6-in-IP represents the tunneling scenario where isolated IPv6 domains are communicating across IPv4 networks.

— **Automatic Tunneling using IPv4-compatible addresses is no longer supported:** Automatic Tunneling using the special IPv6 address type known as "IPv4-compatible address", is not supported.

NOTE This section focuses on HP-UX 11i v2 IPv6 *transport* features, but be aware that IPv6 is also supported on the following components: Internet Services, DCE, DLPI, FDDI, SAM-NNC, Libc, Commands, Desktop (CDE), X11R6-based applications, C2 Audit, EMS, Online Diagnostics, SNMP, nettl, IPSec, Kerberos Client, Service Guard, Glance, HP-UX Secure Shell, Apache, and JVM. Refer to product-specific documentation for more information.

- IPv6/IPv4 Dual Stack support: HP-UX 11i v2 IPv6 supports both IPv4 and IPv6 applications. Programmers can write IPv6 applications that communicate with both IPv6 and IPv4 peers. Existing IPv4 applications do not need to be modified.
- IPv6 tunneling enables IPv6/IPv4 hosts and routers to connect with other IPv6/IPv4 hosts and routers over the existing IPv4 network. IPv6 tunneling encapsulates IPv6 datagrams within IPv4 packets.

The encapsulated packets travel across an IPv4 network until they reach their destination host or router. The IPv6-aware host or router decapsulates the IPv6 datagrams, forwarding them as needed. IPv6 tunneling eases IPv6 deployment by maintaining compatibility with the large existing base of IPv4 hosts and routers.

- Fully supports Ethernet Links and FDDI links.
- MC/ServiceGuard Enablement for IPv6 support.
- IPv6 Stateless Address Autoconfiguration.
- IPv6 Neighbor Discovery.
- TCP/UDP over IPv6, PMTUv6, ICMPv6, IPv6 MIBs and Sockets APIs.
- Network Configuration and Troubleshooting Utilities for both IPv4 and IPv6: `ifconfig`, `netstat`, `ping`, `route`, `ndd`, `ndp` (neighbor-discovery command for IPv6 only) and `traceroute`. There have also been enhancements to `nettl` and `netfmt` for IPv6 tracing and formatting.
- The `netconf-ipv6` file stores IPv6 settings. The `/etc/rc.config.d/netconf-ipv6` configuration file stores IPv6 configuration information similar to IPv4's `/etc/rc.config.d/netconf` file.
- The `/etc/hosts` file now supports IPv6 and IPv4 addresses. The `/etc/hosts` file contains IP addresses and corresponding host names. The file can contain IPv4 and IPv6 addresses for the same host. Lookup policies are identical to IPv4. For example:

```
15.15.15.15   hpindon
3ffe:1111::1234 hpindon         hpindon6
```

- Name Service Switch: `/etc/nsswitch.conf` is a configuration file for the name service switch. The `ipnodes` entity specifies which name services resolve IPv6 addresses and host names. Refer to the `nsswitch.conf(4)` man page for more information.

Limitations

The following section describes limitations of IPv6 transport in HP-UX 11i v3.

`setparms` Not Enhanced for IPv6 Configuration

On HP-UX 11i v3, the `setparms` utility has not been enhanced to support IPv6 configuration.

Multihomed Host Limitation

In the absence of a router that is advertising prefixes, no more than one interface can be configured with IPv6 addresses on a host with multiple physical network interfaces. If multiple physical interfaces are configured with IPv6 addresses, and if there is no Router Advertisement received on any interfaces, the host has no way of knowing which interface to send packets out on. If packets are sent out on the interface that is on a different link than the destination node, then communication will fail. This configuration is neither recommended nor supported.

Distributed File System Limitations

NIS, and NFS are currently not supported over IPv6.

SAM Limitation

`SAM` has not been enhanced to support the tunneling enhancements available with HP-UX 11i v3. HP-UX 11i v3 tunneling configuration must be done by editing the `/etc/rc.config.d/netconf-ipv6` file or by using the `ifconfig` command.

Socket Caching Limitation

Socket caching does not support IPv6.

2 Configuration

This chapter summarizes the steps to configure LAN interfaces, assign IPv6 addresses, optionally enabling IPv6 tunneling through IPv4 networks, and assigning host names to IPv6 addresses.

The first interface configured on a physical LAN interface is called the **primary interface**. Additional interfaces configured on the same physical device are called **secondary interfaces**. You must configure an IPv6 primary interface to use IPv6 over that interface.

Configuring IPv6 Interfaces and Addresses

This section describes IPv6 interface and address configuration tasks that involve editing the `/etc/rc.config.d/netconf-ipv6` file.

Before configuring IPv6 interfaces, remember:

- To edit the `netconf-ipv6` file and to activate the configuration, you must have superuser capabilities.
- The `netconf-ipv6` file and the script that is executed are shell programs; therefore, shell programming rules apply.
- To activate the `netconf-ipv6` configuration, you must either reboot the system or use `ifconfig`, `route` commands with appropriate equivalent values. (Note: `ifconfig` and `route` configuration changes are ephemeral and do not permeate across reboots.) Refer to the "Activating netconf-ipv6 file Configuration" on page 18 for more information.
- To configure HP-UX 11i v2 IPv6, you may use SAM. Open SAM, proceed to the "Networking and Communications" area and navigate (proceed) as needed for your configuration tasks (Network Interface Cards, Name Service Switch, Hosts).

NOTE

`SAM` has not been enhanced to support the tunneling enhancements available with HP-UX 11i v3. HP-UX 11i v3 tunnel configuration must be done by editing the `/etc/rc.config.d/netconf-ipv6` file or by using the `ifconfig` command.

- `setparms` has not been enhanced to support IPv6 configuration.

Configure IPv6 interfaces and routing using one of the following methods:

- Stateless autoconfiguration
- Manual configuration

These methods are described in the following sections.

Stateless Autoconfiguration

Addresses on IPv6 interfaces, unlike IPv4 interfaces, can be configured without manual intervention. With stateless address autoconfiguration, the primary interface (`lanX:0`) is automatically assigned a link-local IPv6 address by the system when the interface is configured (marked “up”). This link-local IPv6 address is generated by prepending a fixed local address prefix (`fe80::`) to a token derived from the MAC address. (The address is verified to be unique.) This allows each IPv6 interface to have at least one source address that can be used by Neighbor Discovery.

If an IPv6 router on the network advertises network prefixes in router advertisements, IPv6 derives secondary IPv6 addresses based on the network interface identifier of the primary interface and on the network prefixes advertised. IPv6 assigns this address to a secondary interface for the network interface.

Refer to “Stateless Address Autoconfiguration” on page 44 in Chapter 4 of this guide, and the `ifconfig(1M)` man page for more information.

Configuring a Primary Interface (Required)

To configure a primary interface, edit the `IPV6_INTERFACE[0]` statement in the `/etc/rc.config.d/netconf-ipv6` file to specify the interface name, such as lan0. The interface name must be the name of the physical interface card, as reported by `lanscan`.

A sample `netconf-ipv6` file entry is as follows:

```
IPV6_INTERFACE[0]="lan0"
IPV6_INTERFACE_STATE[0]="up"
```

Again, in the above example, the address is automatically assigned. Note that autoconfiguration is not mandatory, manual specification of the address is also allowed and is described below.

Configuring Secondary Interfaces

If an IPv6 router that advertises network prefixes resides on the LAN, a secondary interface is automatically configured after the primary interface comes up. IPv6 builds additional secondary interfaces for each network prefix advertised.

If you manually configure a link-local address for the primary interface, then autoconfigured secondary addresses are derived from the interface identifier part of the manually configured address for the primary interface.

For example, if an IPv6 router on the LAN advertises two prefixes (such as `3ffe::/64` and `2000::/64`), HP-UX 11i v2 IPv6 configures two secondary interfaces.

Configuring Route Information

HP-UX 11i v2 IPv6 automatically configures network routes based on the prefix information received from an IPv6 router. HP-UX 11i v2 IPv6 automatically adds the router to its list of default gateways if the router advertises a non-zero router-lifetime value.

Manual Configuration

The following section describes the manual configuration process for HP-UX 11i v2 IPv6.

Configuring a Primary Interface

To configure an IPv6 link-local address for a primary interface, edit the `IPV6_INTERFACE[0]` statement in the `/etc/rc.config.d/netconf-ipv6` file to specify the interface name and the interface state, either `up` or `down`. The interface name must be the name of the physical interface card, as reported by `lanscan`.

To manually specify a link-local address for the primary interface, note that the universal/local "U" bit must be set to 0. That implies, that the manually configured address for the primary interface must match the pattern FE80::xMxx:xxxx:xxxx:xxxx where x are hexadecimal digits, and M is either 0, 1, 4, 5, 8, 9, C, or D. (To be more specific, break M down to the bit level and thus, M = yy0y, where y can be 0 or 1.)

A sample `netconf-ipv6` file entry is as follows:

```
IPV6_INTERFACE[0]="lan0"
IPV6_INTERFACE_STATE[0]="up"
IPV6_LINK_LOCAL_ADDRESS[0]= "fe80::1"
```

Note that if you do not specify a link-local address, then as described earlier in the autoconfiguration section, a link-local address is automatically configured for the primary interface based on the interface's 48-bit MAC address.

Configuring Secondary Interfaces

If no IPv6 Router on the LAN advertises network prefixes, you can add secondary interface entries to the `/etc/rc.config.d/netconf-ipv6` file. Editing the `netconf-ipv6` file allows you to identify the network interface name, IPv6 address, and prefix length and also to add entries to the network routing table.

A sample `netconf-ipv6` file entry is as follows:

```
IPV6_SECONDARY_INTERFACE_NAME[1]="lan0:1"
IPV6_ADDRESS[1]="2345::5432"
IPV6_PREFIXLEN[1]="64"
IPV6_SECONDARY_INTERFACE_STATE[1]="up"
DHCPV6_ENABLE[1]=0
```

Always set `DHCPV6_ENABLE` to 0.

For more information about specifying interface names for multiple interfaces, refer to Chapter 4, “IPv6 Addressing and Concepts,” on page 39.

Configuring a Default IPv6 Route

In the absence of router advertisements, you can add the default IPv6 router information to the `/etc/rc.config.d/netconf-ipv6` file. The routing configuration parameters have an index value, [x], that groups the routing parameters together.

A sample `netconf-ipv6` file entry is as follows:

```
IPV6_DESTINATION[0]="default"
IPV6_GATEWAY[0]="2008:7:6:5:4:3:2:1"
IPV6_ROUTE_COUNT[0]="1"
IPV6_ROUTE_ARGS[0]=""
```

Tunneling

HP-UX 11i v3 provides several important changes to tunneling from in base (default) HP-UX 11i v2 IPv6. These changes are also included in HP-UX 11i v2 PI (September 2004 release). Highlights of these changes are:

— Configured tunneling is point-to-point with an address assigned to both tunnel endpoints. As a result, you can no longer use the `route` command to configure a tunnel. You must use `ifconfig` and `/etc/rc.config.d/netconf-ipv6`, and be aware that the tunneling parameters have changed.

— The HP-UX server can be configured as a router in both point-to-point configured tunnels and in point-to-multipoint “6to4” tunnels. Prior to HP-UX 11i v2 PI (September 2004 release), the HP-UX 11i v2 node would only perform as a “6to4” host” not as a “6to4” router. Parameters for “6to4” router configuration are in `ifconfig` and in `/etc/rc.config.d/netconf-ipv6`.

— Automatic tunneling using the IPv4-compatible address is not supported.

The following sections provides basic examples for configuring an IP6-in-IP tunnel and a “6to4” tunnel. For more information including additional optional tunnel parameters not mentioned in these examples, refer to the `/etc/rc.config.d/netconf-ipv6` file and the `ifconfig(1M)` man page that ship with HP-UX 11i v3.

For more information on the tunneling mechanisms supported in HP-UX 11i v3, refer to the section on “Tunneling” on page 56, in Chapter 5, IPv6 Software and Interface Technology,, later in this Guide.

NOTE SAM has not been enhanced to support the tunneling enhancements available with HP-UX 11i v3.

Creating an IP6-in-IP Point-to-Point Configured Tunnel

If you regularly expect to exchange data between isolated IPv6 networks over an IPv4 network, you may want to create a configured IP6-in-IP tunnel. IP6-in-IP tunnels can be set up as host->host; host->router; router->host or router->router. In HP_UX 11i v3 the HP-UX 11i v3 node can perform the role of a host or router.

A sample `netconf-ipv6` file entry, for configuring the HP-UX 11i v3 IPv6 node is as follows:

```
TUN_INTERFACE_NAME[0]=”iptu0”
TUN_TYPE[0]=”ip6inip”
TUN_LOCAL_ADDRESS[0]=””
TUN_REMOTE_ADDRESS[0]=””
TUN_ENCAP_SRC_ADDRESS[0]=”15.1.1.1”
TUN_ENCAP_DST_ADDRESS[0]=”15.2.2.2”
TUN_INTERFACE_STATE[0]=”up”
```

This example minimizes the number of variables that need to be specified. For example, TUN_LOCAL_ADDRESS[0] was not specified since the IPv6 link-local address for this value can be automatically configured based on the TUN_ENCAP_SRC_ADDRESS[0] value. Similarly, the IPv6 link-local TUN_REMOTE_ADDRESS can be automatically configured based on the TUN_ENCAP_DST_ADDRESS.

Creating a "6to4" Point-to-Multipoint Configured Tunnel

"6to4" offers a point-to-multipoint router-to-router tunneling mechanism for traffic going between IPv6 domains over an IPv4 network. One of the advantages of "6to4" over configured tunneling is that the source router can talk to any other "6to4" router without the need for any manual configuration on the destination router. Thus, "6to4" tunnels do not suffer the scalability problem that configured tunnels do.

A sample `netconf-ipv6` file entry to configure a "6to4" tunnel is as follows:

```
TUN_INTERFACE_NAME[1]="iptu1"
TUN_TYPE[1]="6to4"
TUN_ENCAP_SRC_ADDRESS[1]="15.13.1.2"
TUN_INTERFACE_STATE[1]="up"
```

In this example, the `TUN_LOCAL_ADDRESS[1]` was not specified since the the "6to4" address for this value can be automatically configured based on the `TUN_ENCAP_SRC_ADDRESS[1]` value. For example, if the `TUN_ENCAP_SRC_ADDRESS` is 15.13.1.2, the "6to4" prefix is 2002:0f0d:0102, which can be combined with an interface identifier of "1" to form the "6to4" address 2002:0f0d:0102::1.

The `TUN_REMOTE_ADDRESS[1]` parameter **must not** be specified since"6to4" is an automatic point-to-multipoint tunnel. The remote end-point of the tunnel will be determined based on routing information. Similarly, the `TUN_ENCAP_DST_ADDRESS[1]` parameter, must not be specified since the destination address will be automatically derived from the destination "6to4" address.

Enabling rtradvd (Router Advertiser Daemon)

When `rtradvd` is configured, it sends router advertisement messages to a local LAN periodically, and, when requested, by a node sending a router solicitation message. Refer to the `rtradvd(1M)` man page for more information.

Configuration for `rtradvd` is set, on a per interface basis, by editing the `/etc/rtradvd.conf` file. The `rtradvd.conf` file allows for setting global defaults as well as interface specific settings for both interface options and prefixinfo specific options. Refer to the `rtradvd.conf(4)` man page for more information.

Required Steps: To configure the HP-UX system to run `rtradvd`, and enable the Router Advertisement functionality, the following steps must be taken: the `/etc/rtradvd.conf` file must be edited as needed; the “private” interface flag must be cleared (“-private” for each enabled interface) and the `rtradvd` daemon must be enabled. More specifically:

- Edit the `/etc/rtradvd.conf` file as needed

  The example below shows the minimum configuration needed to send router advertisement packets containing the prefix `2008:65::/64` on `lan0`.

```
#example begins

defaults {
        AdvSendAdvertisement on ;
 };

  interface lan0 {
        prefixinfo 2008:65::/64 {
        };
 };

#example ends
```

  For more examples, refer to the `rtradvd.conf(4)` man page.

- Edit the `/etc/rc.config.d/netconf-ipv6` file to enable `rtradvd` to start up at boot. Also, clear the "private" interface flag (-private), on the appropriate interface(s) to disable stateless address autoconfiguration using prefixes received in router advertisements. The default is "private", and when set to "private" the interface will autoconfigure addresses using prefixes received in router advertisements. For more information, refer to the relevant commented text in the `/etc/rc.config.d/netconf-ipv6` file that is included with HP-UX 11i v3 IPv6.

  A sample `netconf-ipv6` file entry, which clears the private flag and enables `rtradvd`, is as follows:

```
IPV6_INTERFACE[0]=”lan0”
IPV6_INTERFACE_STATE[0]=”up”
IPV6_INTERFACE_FLAG[0]= “-private”
#
#
RTRADVD=1
```

Activating netconf-ipv6 file Configuration

You can activate the `netconf-ipv6` configuration in one of the following ways:

- By rebooting the system.
- Or alternatively, by executing the `ifconfig` and `route` commands, as needed, to make equivalent configuration settings.

NOTE HP recommends rebooting your system to activate any changes you made to your `netconf-ipv6` file. A reboot is the cleanest way to reconfigure an interface because the reboot handles any network initialization dependencies.

HP recognizes that system reboots are disruptive to end users. To delay or schedule the reboot, but still make your configuration changes active, you may execute the `ifconfig` and `route` commands with the appropriate values for your network. These values are ephemeral however, and will not last across reboots. After the reboot, the values in your `netconf-ipv6` file will be used. Refer to the examples that follow and the `ifconfig(1M)`, and `route(1M)` man pages for more information on using these commands.

Example ifconfig and route Commands

HP recommends editing the `/etc/rc.config.d/netconf-ipv6` file to preserve IPv6 interface and address configurations across system reboots. For reference, the commands equivalent to the `netconf-ipv6` edits described earlier are listed below. Refer to the `ifconfig(1M)` and `route(1M)` man pages for more information.

To configure a primary interface, enter:

```
ifconfig lan0 inet6 up
```

To configure a secondary interface, enter:

```
ifconfig lan0:1 inet6 2345::5432 up
```

To add a default IPv6 route, enter:

```
route inet6 add net default 2008:7:6:5:4:3:2:1
```

To create an IP6-in-IP tunnel, enter:

```
ifconfig iptu0 inet6 tunnel ip6inip tsrc 192.1.1.1 tdst
192.2.2.2 up
```

To create a "6to4" tunnel, enter:

```
ifconfig iptu0 inet6 tunnel 6to4 2002:f0e:8cc::1 tsrc
15.13.1.2 up
```

NOTE Remember that configuration using `ifconfig` and `route` is ephemeral, and not maintained after a system reboot.

Host Names and IPv6 Addresses

The following section provides additional information on how addressing works on HP-UX 11i v3 IPv6.

Creating the /etc/hosts File

It is generally recommended to add IPv6 addresses (known as AAAA records) to a DNS Name Server only when the following conditions are true:

- The IPv6 address is assigned to the interface on the node
- The address is configured on the interface
- The interface is on a link which connects to the IPv6 infrastructure

HP recommends beginning with IPv6 addresses and host names in the `/etc/hosts` file on a development network; then adding IPv6 addresses and hosts to a Domain Name Service when moving IPv6 to a production backbone network.

This subsection describes how to edit the `/etc/hosts` file to add an IPv6 address and host name for the network interface you are configuring.

NOTE If using the name service DNS over IPv6, add the IP address and host name to the appropriate databases on the name server system. Refer to BIND v9.2.0 (or later) documentation on `http://www.docs.hp.com` for more information on DNS over IPv6.

The `/etc/hosts` file associates IP host addresses with mnemonic host names and alias names. It contains the names of other nodes in the network with which your system can communicate.

An example `/etc/hosts` file ships with HP-UX 11i v3.

Example Host Name Entry

The example below shows how a system with the name, `host3`, might be referenced in the `/etc/hosts` file:

System name in `swinstall` screen: `host3`

/etc/hosts file:

```
3ffe:ffff:101::230:6eff:fe04:d9ff  host3  host3.site2.region4
192.1.2.34  hpfcrm loghost
```

NOTE HP-UX 11i v2 IPv6 is a dual stack implementation. A single host name can have entries for both an IPv6 address and an IPv4 address in /etc/hosts.

Name and Address Lookup for IPv6

/etc/nsswitch.conf (nsswitch.conf(4)) is a configuration file for the name service switch. The ipnodes entity specifies which name services resolve IPv4 and IPv6 addresses and host names on HP-UX 11i v3 IPv6 transport.

More specifically, on HP-UX 11i v2, the ipnodes keyword specifies the resolver policy for the library functions getnameinfo(3N), getaddrinfo(3N), getipnodebyname(3N) and getipnodebyaddr(3N) for both IPv4 and IPv6 addresses. The existing keyword "hosts" specifies the resolver policy for the library functions gethostbyname() and gethostbyaddr() for IPv4 addresses.

NOTE Internet Services applications (such as telnet, r-commands, etc.) use these library functions to resolve IPv4 and IPv6 addresses.

By default, the /etc/nsswitch.conf is not on the system. The default ipnodes policy (same as default hosts policy) is as follows:

```
dns [NOTFOUND=return] files
```

This policy implies that dns is the authoritative resolver and will only try files if dns is down. If dns is available but returns NOTFOUND, the search stops.

Thus, if DNS has **not** been set up as the definitive source, and files (/etc/hosts) may need to be used for address and host name resolution, HP recommends adding the following entry to /etc/nsswitch.conf:

```
ipnodes: dns [NOTFOUND=continue] files
```

Or if /etc/hosts is to be the primary Name Service, the entry would be set as follows:

```
ipnodes: files [NOTFOUND=continue] dns
```

NOTE You can not specify NIS or NIS+ on the ipnodes entry.

Manually editing nsswitch.conf

If the current system has no nsswitch.conf file, use a text editor to create an /etc/nsswitch.conf file containing one of the following lines, or copy the /etc/nsswitch.defaults file and modify as needed.

If DNS is the primary Name Service, but not necessarily the definitive source, and files (/etc/hosts) may need to be used for address and host name resolution, add:

```
ipnodes: dns [NOTFOUND=continue] files
```

Or if /etc/hosts is to be the primary Name Service, add:

```
ipnodes: files [NOTFOUND=continue] dns
```

Refer to the nsswitch.conf(4) man page for more information.

3 Troubleshooting

This chapter provides guidelines for troubleshooting HP-UX 11i v3 IPv6 transport. It contains a troubleshooting overview and diagnostic flowcharts.

Troubleshooting Overview

Troubleshooting problems on HP-UX 11i v3 IPv6 transport, can involve a variety of hardware and software components. The problem impacting your system might originate in another part of the network.

Because HP-UX 11i v2 IPv6 supports an IPv6/IPv4 Dual Stack, test IPv4 connectivity before testing IPv6 connectivity. Refer to the *HP-UX LAN Administrator's Guide* (available at `http://www.docs.hp.com`) for IPv4 troubleshooting advice.

If you are still unable to identify your problem, proceed to the troubleshooting flowcharts. The troubleshooting flowcharts provide logical steps to follow. Use the diagnostic flowcharts provided in this chapter to verify your assumptions and to try to identify whether the problem is with HP-UX 11i v3 IPv6 transport or router configuration.

Diagnostic Flowcharts

Below is a summary of the types of network tests in the diagnostic flowcharts. To diagnose your problem, first verify the connections and configuration on your system (Flowcharts 1 through 5). If this does not solve your problem, use Flowchart 6 to test and/or verify connectivity with a remote system.

Flowchart 1	Transport Level Test using Internet Services
Flowchart 2	Network Connectivity Test
Flowchart 3	Name Services Test
Flowchart 4	Interface Test
Flowchart 5	Interface Test (continued)
Flowchart 6	Router Remote Loopback Test

Transport Level Loopback Test using Internet Service: Verifies round-trip communication between Transport Layers on the source and target host using telnet.

Network Connectivity Test: Verifies round-trip communication between Network Layers on the source and target host using the `ping(1M)` diagnostic.

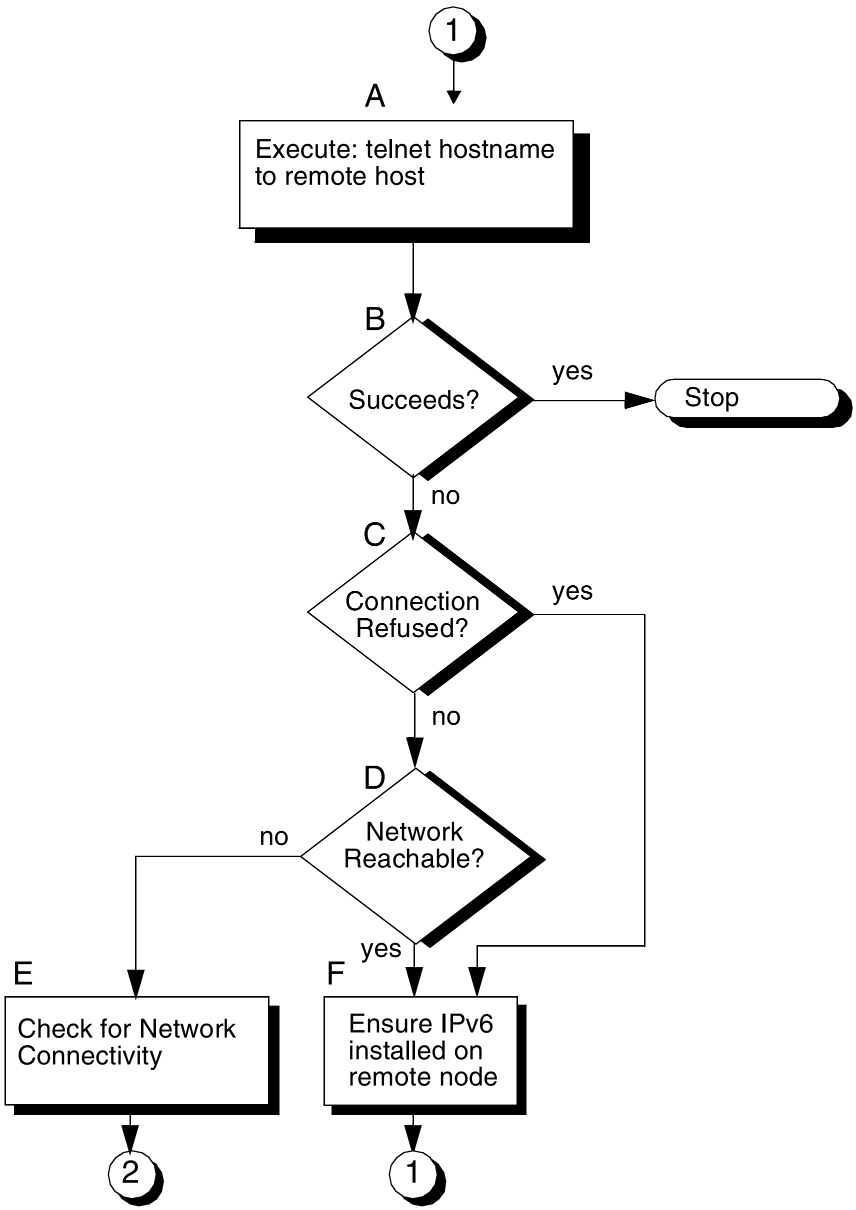
Name Services Test: Verifies host name and IPv6 address resolution.

Interface Test: Verifies the configuration of the network interface on a host using the `lanscan`, and `ifconfig` commands.

Router Remote Loopback Test: Verifies the connection between local and remote nodes through IPv6 routers using the `ping` and `netstat` commands.

Flowchart 1: Transport Level Testing using Internet Services

Figure 3-1 Flowchart 1



Flowchart 1 Procedures

A. *Execute:* `telnet <hostname> to remote host`. Try to connect using `telnet` to a remote host.

B. *Succeeds?* If telnet succeeds, stop. The system connects using TCP over IPv6 through the Transport Layer (OSI Layer 4).

C. *Connection Refused?* Trying to connect to a remote system where HP-UX 11i v2 IPv6 is not installed can cause this message.

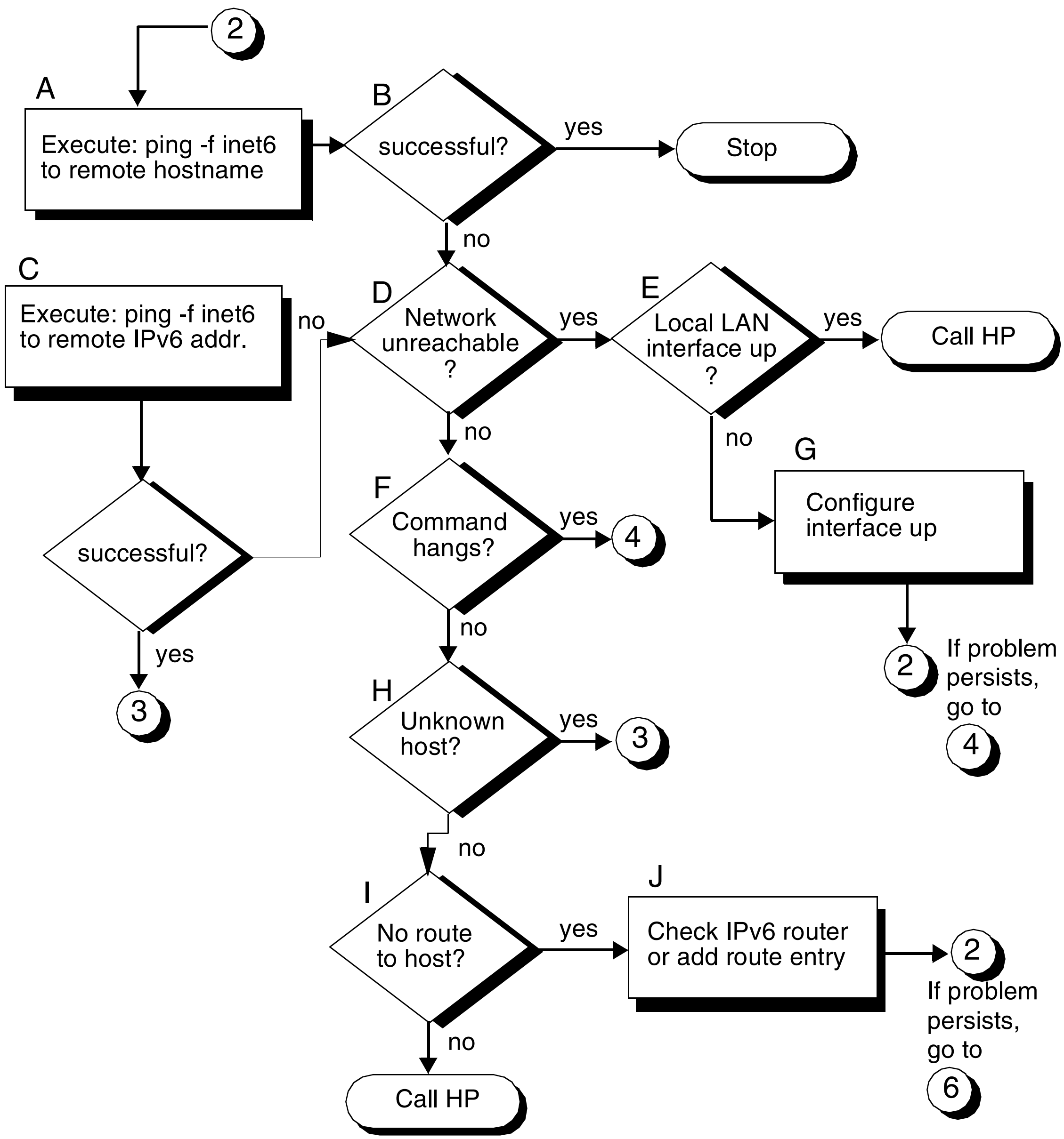
D. *Network Reachable?* If it is, go to F, otherwise continue to E.

E. *Check for Network Connectivity.* Ensure network connectivity by following the steps in Flowchart 2.

F. *Ensure IPv6 installed on remote node.* If `telnet` still fails, examine the `etc/inetd.conf` file on the remote system.

Flowchart 2: Network Connectivity Test

Figure 3-2 **Flowchart 2**



Flowchart 2 Procedures

A. *Execute ping to remote IPv6 "host name".* Using `ping`, send an ICMPv6 message to the remote host with which you are having problems connecting. For example, the remote host name is `hpindon`. Enter:

```
ping -f inet6 hpindon
```

B. *ping successful?* A message is printed on stdout for each ping packet returned by the remote host. If packets are being returned, your system has network level connectivity to the remote host.

C. *Execute ping to remote IPv6 address.* Using `ping`, send a message to the IPv6 address of the remote host. For example,

```
ping -f inet6 8:7:6:5:4:3:2:1
```

D. *Network unreachable?* If so, examine the status of the local LAN interface first. If not, proceed to F.

E. *Local LAN interface up?* Execute `ifconfig` on the local interface to be sure it is configured up. If it is not, go to G. If it is up, call your HP representative for help.

F. *Command hangs?* If a message is not returned after executing ping, go to Flowchart 4, otherwise go to H.

G. *Configure interface up.* If you find the local interface is not up, execute `ifconfig` with the appropriate flags set. Begin Flowchart 2 again. If the problem persists, go to Flowchart 4.

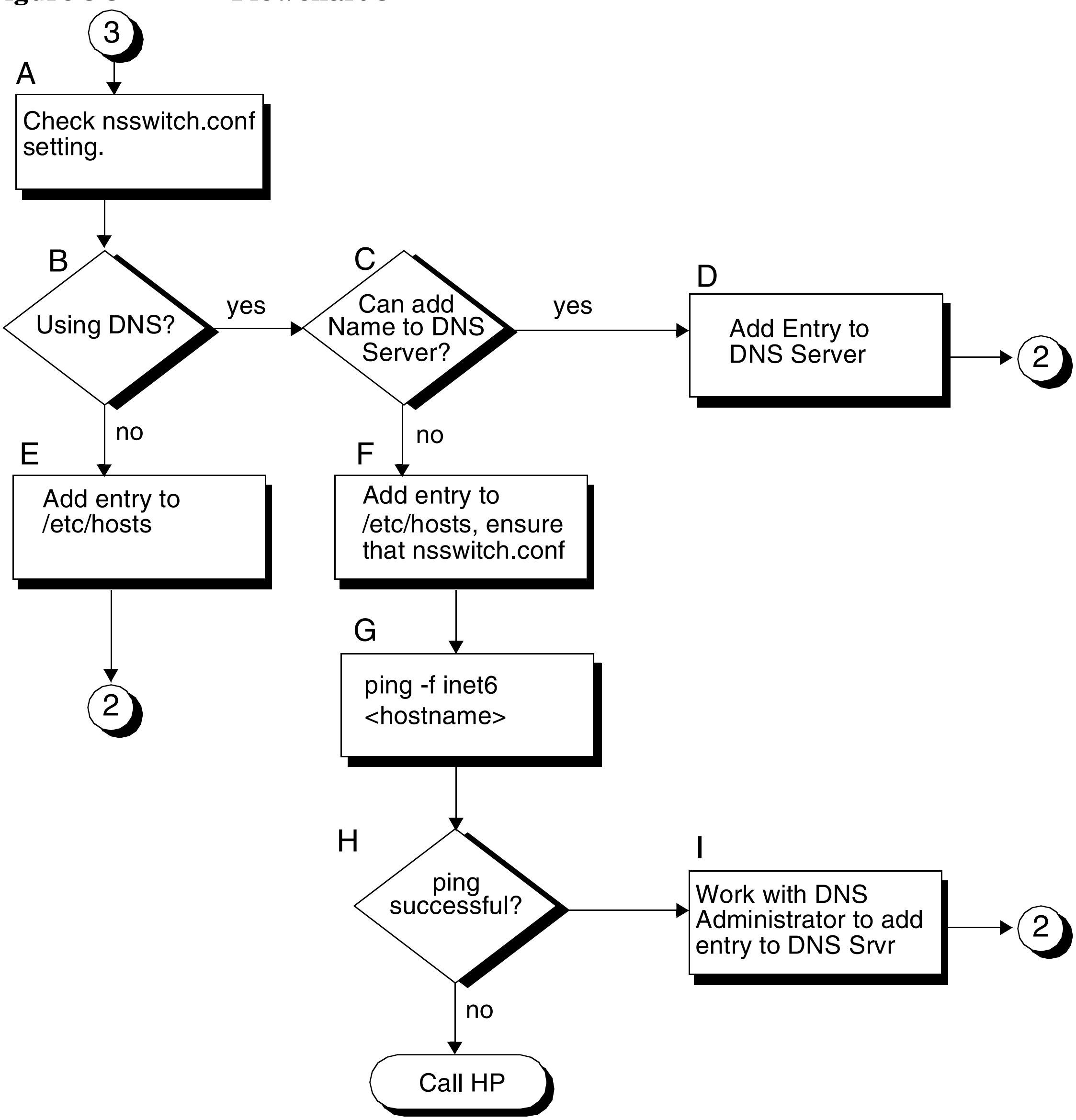
H. *Unknown host?* (Error= Unknown host <hostname>?) If so, there is a problem with the IPv6 address configuration for the host <hostname> in the `/etc/hosts` file or on the name server. Go to Flowchart 3. Otherwise, proceed to I.

I. *No route to host?* (Error= Sendto: No route to host?) Use `netstat -rn` to examine the routing table. If there is no route to host, go to J. Otherwise, call your HP representative for help.

J. *Check IPv6 Router or add route table entry.* Add a route table entry to that host, or ensure that the IPv6 router advertises correct prefixes. Then try Flowchart 2 again. If the problem persists, go to Flowchart 6.

Flowchart 3: Name Service Test

Figure 3-3 **Flowchart 3**



Flowchart 3 Procedures

A. *Check /etc/hosts and /etc/nsswitch.conf files.* If needed, add a missing host name or IPv6 address. If the IPv6 address for the host is in `/etc/hosts`, ensure that you have an `/etc/nsswitch.conf` file entry with an appropriate `ipnodes` policy. For example,

```
ipnodes: DNS [NOTFOUND=continue] files
```

and start again with Flowchart 3.

B. *Using DNS?* If your name and IPv6 address resolution policy use DNS as the primary resolver, go to C. Otherwise, proceed to E.

C. *Can you add a Host Name to the DNS Server?* Are you a DNS administrator? If so, continue on to D, otherwise proceed to F.

D. *Add Entry to DNS Server.* Refer to the BIND 9.2.0 information in the *HP-UX IP Address and Client Management Administrator's Guide* for details (available at `http://docs.hp.com`). Then retry Flowchart 2.

E. *Add entry to /etc/hosts.* If your name and IPv6 address resolution policy uses `/etc/hosts` as the primary resolver, add a correct IPv6 address and host name to the local `/etc/hosts` file. Then retry Flowchart 2.

F. *Add entry to /etc/hosts and ensure that nsswitch.conf is configured properly.* Add a correct IPv6 address and host name to the local `/etc/hosts` file. Ensure that your IPv6 address resolution policy, specified with the `ipnodes` keyword in `/etc/nsswitch.conf` includes using "files" (`/etc/hosts`) in the policy. Then retry Flowchart 2.

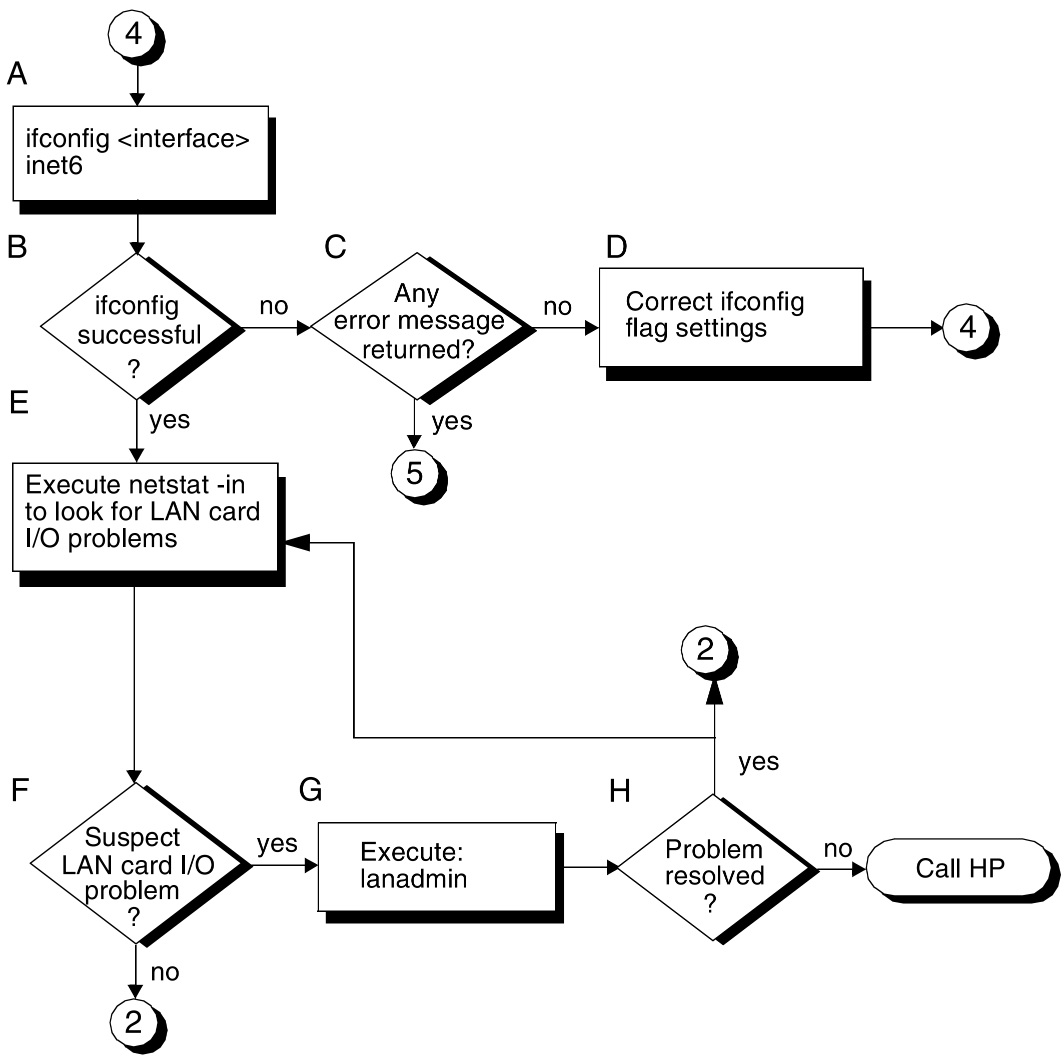
G. *ping -f inet6 hostname.* Test connectivity to the remote host using the `ping` command.

H. *ping successful?* If `ping -f inet6 <hostname>` succeeds using a host name and IPv6 address from `/etc/hosts`, DNS needs updating, proceed to I. If `ping` fails, examine the `/etc/hosts`, `/etc/resolv.conf`, and `/etc/nsswitch.conf` files on both the local and remote hosts. If all look correct, call your HP representative for help.

I. *Work with DNS Administrator to add entry to DNS Server.* When entry is added, retry Flowchart 2 to ensure that DNS correctly resolves host names and IPv6 addresses.

Flowchart 4: Interface Test

Figure 3-4 **Flowchart 4**



Flowchart 4 Procedures

A. *Execute: ifconfig <interface>inet6.* Execute `ifconfig` on the interface you want to test. For example, to view LAN interface `lan0`, enter:

```
ifconfig lan0 inet6
```

B. *ifconfig successful?* ifconfig succeeds when the output shows an Internet address and the flags: UP, RUNNING, MULTICAST, ONLINK. If successful, go to E, if not continue to C.

C. *Any error message returned?* If `ifconfig` fails and displays an error message, go to Flowchart 5. Flowchart 5 shows what to do based on the error message. Otherwise continue to D.

D. *Correct `ifconfig` with non-default flag settings.* If `ifconfig` returns an unexpected flag setting, re-execute the command with the proper setting. For more information, refer to the `ifconfig(1M)` man page. Start again with Flowchart 4.

E. *Execute: netstat -inf -inet6.* If `ifconfig` succeeds, then the network interface is configured correctly. `netstat -i` displays the number of incoming (`Ipkts`) and outgoing (`Opkts`) packets passed through an interface. No increase in the number of incoming or outgoing packets would indicate LAN card I/0 problems.

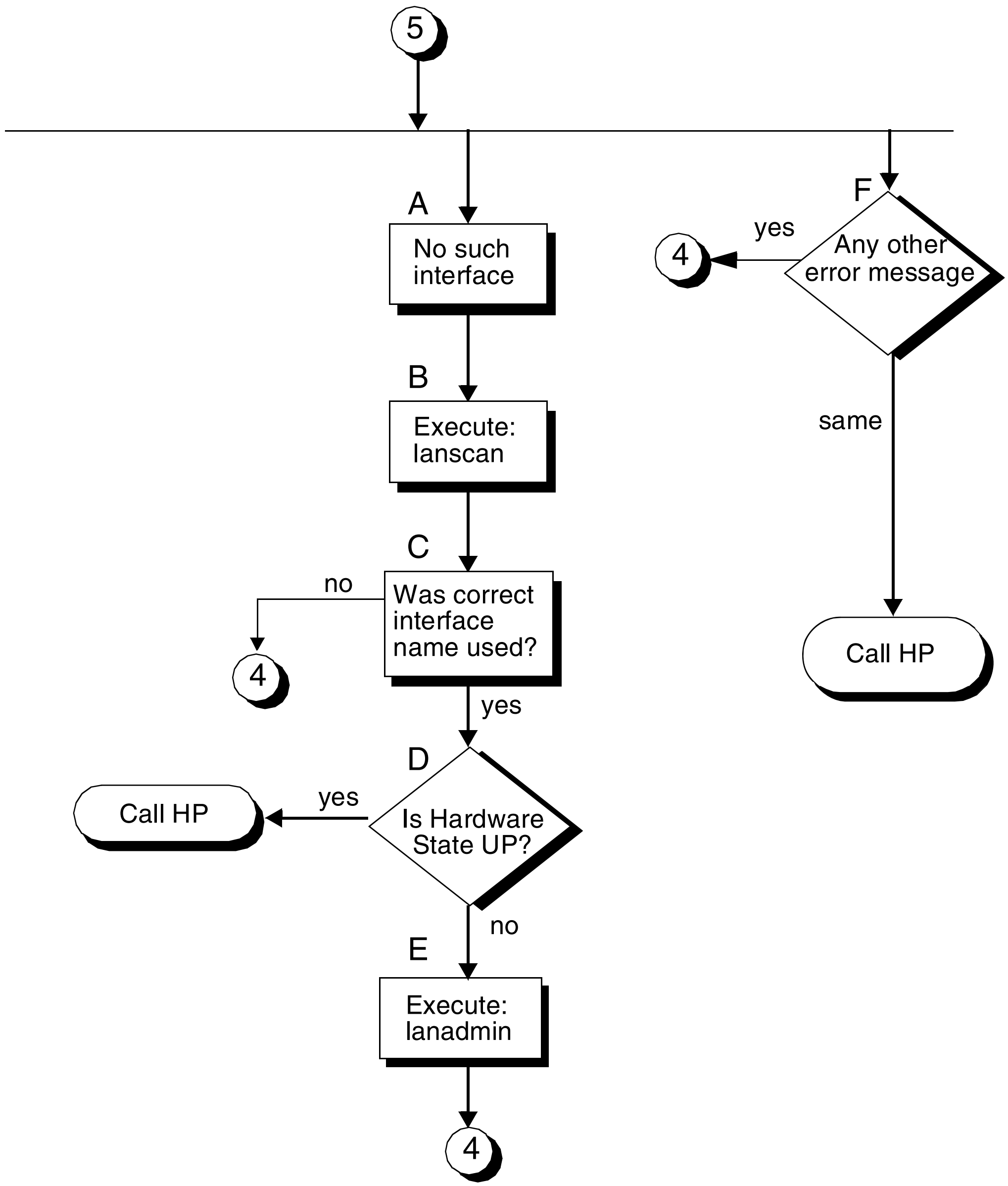
F. *Suspect LAN card I/O problems?* If the statistics indicate possible LAN card problems, go to G, otherwise go to Flowchart 2 to test Network Connectivity.

G. *Execute: lanadmin.* Use `lanadmin` to ensure the LAN card is operational. A substantial increase in the number of the `Ierrs` and `Oerrs` during a file transfer attempt might indicate transmission problems.

H. *Problem resolved?* If you found and corrected the LAN card problem, return to step E to verify the correction. If corrected, re-execute `ifconfig` to bring up the interface, then go to Flowchart 2. If the problem persists, call your HP representative for help.

Flowchart 5: Interface Test continued

Figure 3-5 **Flowchart 5**



Flowchart 5 Procedures

A. *Is error message “No such interface name”?* If not, go to F. If so, the interface name passed to `ifconfig` does not exist on the system. Using `lanscan`, verify the spelling and names of the interfaces on the system.

If the system contains more than one LAN card, make sure the correct number of LAN cards was configured into the kernel and that an `ifconfig` command was executed for each interface.

B. *Execute: lanscan.* Execute `lanscan` to display information about the LAN cards in your system.

C. *Was correct interface name used?* Configure interface using `ifconfig` with the correct interface name. After reconfiguring using the correct interface name, start again with Flowchart 4.

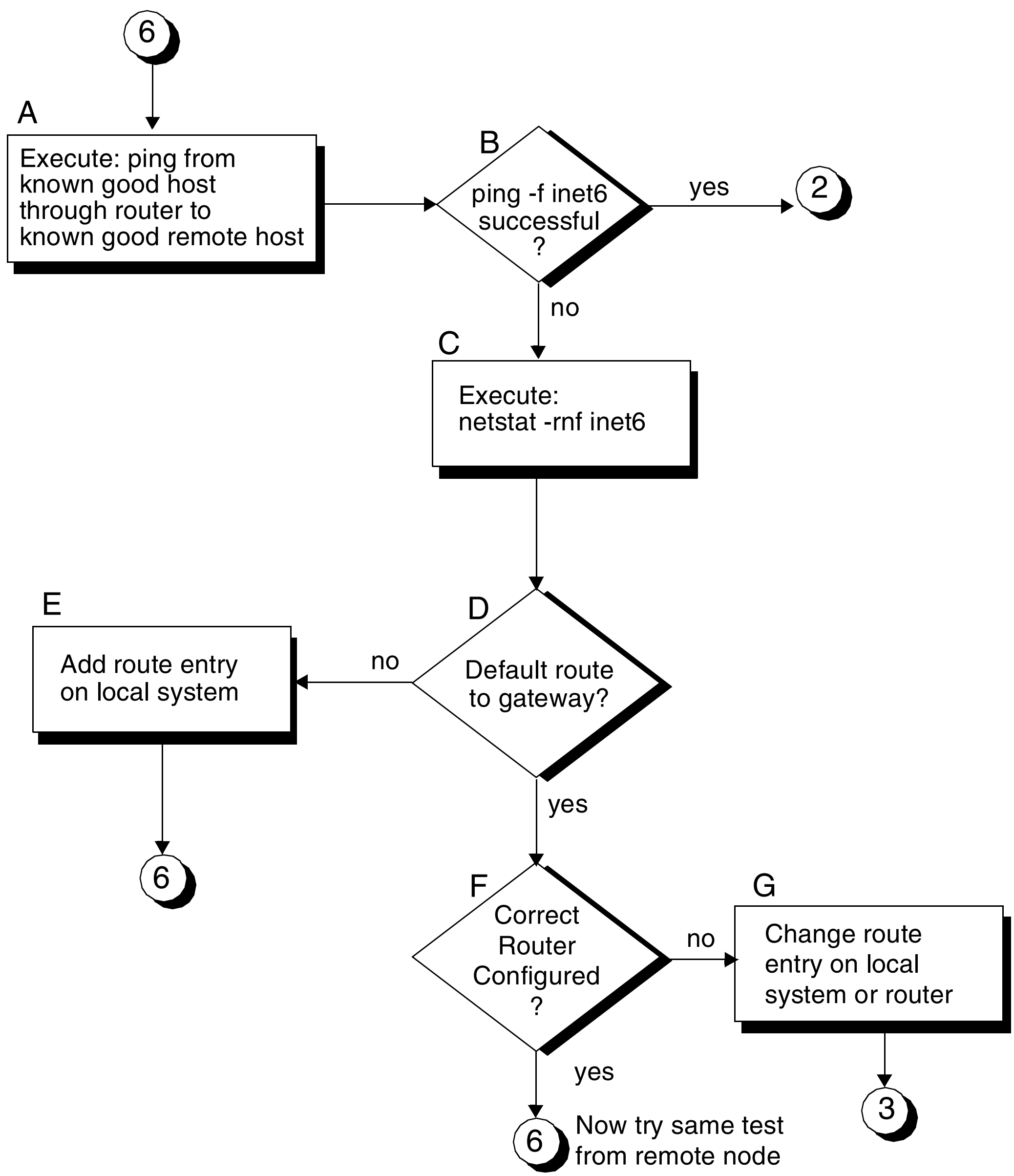
D. *Is Hardware State UP?* Verify the state of the hardware with the output from the `lanscan` command. If the Hardware State is UP call your HP representative for help, otherwise continue to E.

E. *Execute: lanadmin.*e `lanadmin` command to reset the LAN card. Go to Flowchart 4.

F. *Any other error message.* Interpret any other error message and take the appropriate action. Then repeat flowchart 4. If you receive the same error message again, call your HP representative for help.

Flowchart 6: Router Remote Loopback Test

Figure 3-6 Flowchart 6



Flowchart 6 Procedures

A. *Execute: ping from known good host through gateway to known good host on remote network.* This tests router connectivity to the remote network. For more information on `ping`, refer to the `ping(1M)` man page.

B. *ping successful?* If `ping -f inet6` succeeded, return to Flowchart 2. If `ping -f inet6` failed, the problem may exist in the routing table for the problem host. Continue to C.

C. *Execute: netstat -rnf inet6.* To display gateway routing information in numerical form, execute: `netstat -rnf inet6`

D. *Direct route to remote or default route to gateway?* If the route exists, go to F. If not, continue to E to add a new route.

E. *Add route entry on local system.* Use the `route` command to add a route entry to the route table on the local system. Refer to `route(1M)` for a complete description of the command. Or if an IPv6 router on the LAN advertises default routes, wait a few minutes to see if a route advertisement is added to the default router list. Start again with Flowchart 6.

F. *Correct router configured?* If your local host has a route to the correct router, then retry Flowchart 6 from the remote node. If the remote node's routing is configured properly, and both the local and remote nodes can connect to their respective routers, then contact your ISP or network administrator to verify network-to-network connectivity.

G. *Change route entry on local system or router.* If the routing information is incorrect, correct it using `route`, or verify that the IPv6 router is advertising proper subnet prefixes. Then retry Flowchart 2 to test network connectivity.

4 IPv6 Addressing and Concepts

This chapter introduces network addressing concepts for IPv6. It contains sections on Obtaining IPv6 Addresses, IPv6 Address Formats, Neighbor Discovery, Stateless Address Autoconfiguration and some basic general Networking Terminology.

Where to Get IPv6 Addresses

To obtain an IPv6 address, contact a local ISP or the Regional Internet Registries from the following list:

ARIN - American IPv6 registration services
APNIC- Asia Pacific Network Information Center
RIPE - European Regional Internet Registry

The amount of addresses allocated varies according to your network requirements. Small Internet Service Providers (ISPs) or end nodes acquire IPv6 addresses from their upstream provider. Large ISPs, for example can receive from ARIN a minimum prefix of /48 with a second-level allocation of 16 bits for subnets. The remaining 64 bits are for a network interface.

IPv6 Address Formats

IPv6 addresses are 128-bit entities. IPv4 addresses are 32-bit addresses normally written as four decimal numbers (dotted decimal), one for each byte of the address.

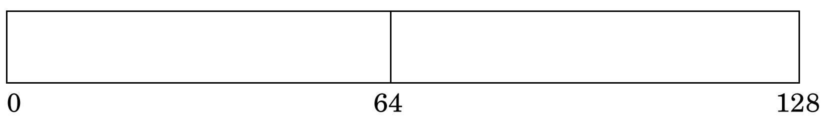
Example: `192.1.2.34`

IPv6 Node Addresses are 128-bit records represented as eight fields of up to four hexadecimal digits. A colon separates each field (:). Example: `3ffe:ffff:101::230:6eff:fe04:d9ff`.

NOTE The symbol "::" is a special syntax that can be used as a shorthand way of representing multiple 16-bit groups of contiguous 0's (zeros). The "::" can appear anywhere in the address; however it can only appear once in the address.

To indicate a subnetwork address, IPv6 uses subnet prefixes similar to IPv4 CIDR format. Figure 4-1shows a 128-bit IPv6 node address with a 64-bit subnet prefix.

Figure 4-1 **IPv6 128-bit Addresses; HP-UX Default Prefix 64**



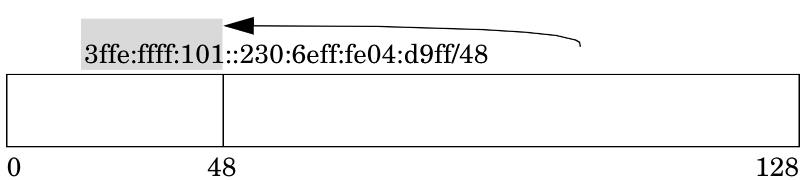
An IPv6 node address and its subnet prefix length can be combined in the following format:

<IPv6-Node-Address>/<Prefix-Length>

Where <*IPv6-Node-Address*> is an IPv6 address and <*Prefix-Length*> is a decimal value specifying how many of the leftmost contiguous bits of the address compose the subnet prefix.

In Figure 4-2, prefix length 48 specifies that the leftmost 48 bits of the IPv6 address compose the subnet prefix.

Figure 4-2 **Example Prefix Length 48**



Address Scope

Link-local	An IPv6 address used on a single link.
Global	An IPv6 address that uniquely identifies a node on the Internet such that packets can be routed to the node from any other node on the Internet.

Address Type

Unicast — Identifies a single interface. Notable unicast addresses are:

Loopback	`::1` Address internal to IPv6 stack
Unspecified	`::` Not a legally defined address

Anycast — Identifies a group of interfaces, possibly belonging to different nodes. A packet sent to an anycast address is delivered to only one of the interfaces in the group. Anycast addresses are currently not supported by HP-UX 11i v2 IPv6.

Multicast — Identifies a group of interfaces, possibly belonging to different nodes. A packet sent to a multicast address is delivered to all the interfaces in this group.

Neighbor Discovery

IPv6 hosts and routers use the IPv6 Neighbor Discovery Protocol to:

- advertise their link-layer address on the local link
- find neighbors' link-layer addresses on the local link
- find neighboring routers able to forward IPv6 packets
- actively track which neighbors are reachable
- search for alternate routers when a path to a router fails

The IPv6 Neighbor Discovery Protocol (ndp) uses ICMPv6. An IPv6-only utility, `ndp` and the Neighbor Discovery Protocol encompass the functionality of the IPv4 Address Resolution Protocol (ARP) and the `arp` utility. ndp also provides some of the address-configuration functionality found in protocols BOOTP and DHCP.

A network device connecting to a network for the first time can learn all parameters necessary to function, solely through Neighbor Discovery information. Both IPv6 hosts and routers advertise their presence using neighbor advertisements and route advertisements, respectively. When an IPv6 host first comes up, it advertises its link-layer address, and solicits neighbor and router information.

For more information, see the `ndp(1m)` and `ndp(7p)` man pages and RFC 2461, "Neighbor Discovery for IP Version 6 (IPv6)."

Stateless Address Autoconfiguration

Stateless address autoconfiguration requires no manual configuration of hosts, minimal configuration of routers, and no additional servers. The primary interface (`lanX:0`) is automatically assigned a link-local address by the system when the interface is configured. This allows each IPv6 interface to have at least one source address that can be used by Neighbor Discovery. Therefore, it is not advisable to assign other addresses to the primary interface besides the link-local address. See RFC 2373 “IP Version 6 Addressing Architecture” for details.

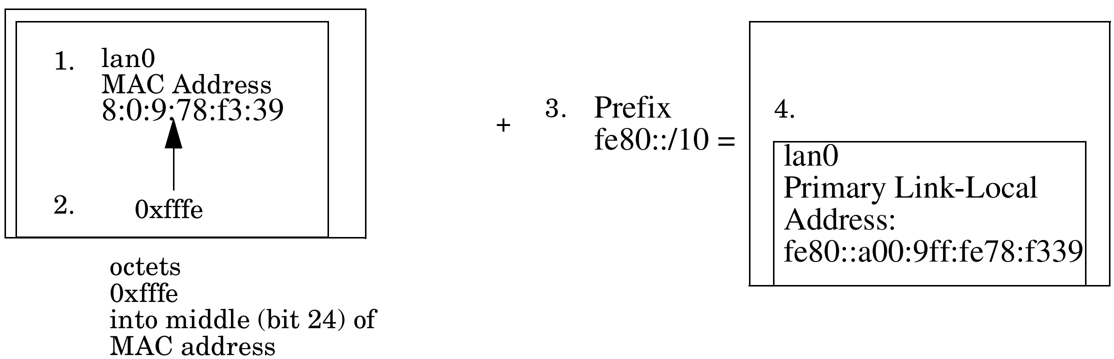
Link-Local Address Assigned Automatically

A link-local address is formed by prepending the well-known link-local prefix `FE80::/10` to the interface identifier which is typically 64 bits long and based on EUI-64 identifiers. Link-local addresses are sufficient for allowing communication among IPv6 hosts attached to the same link.

Figure 4-3 shows the Primary Interface Autoconfiguration steps performed after using the ifconfig command, which is as follows:

```
ifconfig lan0 inet6 up
```

Figure 4-3 Primary Interface Address Autoconfiguration



If you mark an interface "up" without assigning a primary address, the system derives a link-local address by performing the following 4 steps:

1. Taking the LAN card's 48-bit link-level address ("MAC address" `8:0:9:78:f3:39`)

   `0000 1000 0000 0000 0000 1001 0111 1000 1111 0011 0011 1001`

   and putting it into an EUI-64 identifier by:

2. Putting two bytes (`0xffee`) into the middle (bit 24) of the 48-bit link-level address `8:0:9:`**ff:fe**`:78:f3:39`;

0000 10**0**0 0000 0000 0000 1001 **1111 1111 1111 1110** 0111 1000 1111 0011 0011 1001

   and flipping the Universal/local bit (as described in RFC 2373) to form a 64-bit EIU-64 **interface identifier** `a:0:9:ff:fe:78:f3:39`

0000 10**1**0 0000 0000 0000 1001 1111 1111 1111 1110 0111 1000 1111 0011 0011 1001

3. Prepending the well-known prefix `fe80::/10`

4. Forming a 128-bit link-local unicast address for the primary interface `fe80::a00:9ff:fe78:f339`

View the configuration by typing

```
ifconfig lan0 inet6
```

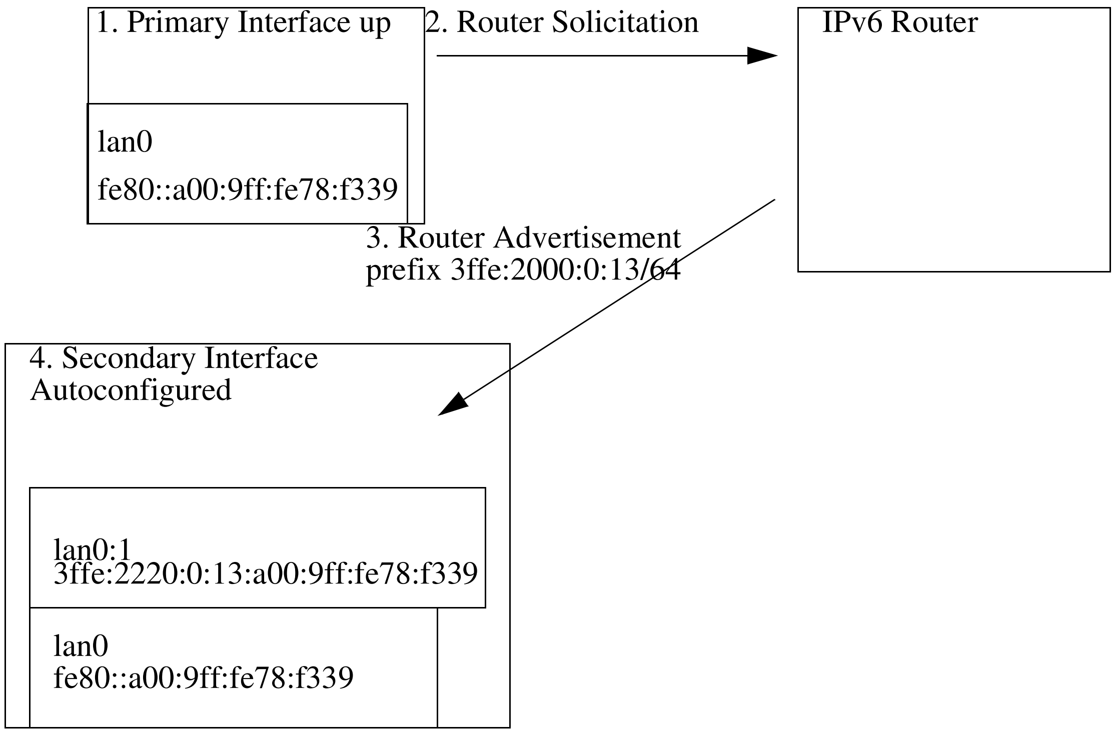
```
lan0: flags=4800841<UP,RUNNING,MULTICAST,ONLINK>
inet6 fe80::a00:9ff:fe78:f339 prefix 10
```

Secondary Interface Autoconfiguration

If an IPv6 router on the network advertises network prefixes in router advertisements, IPv6 derives a second IPv6 address based on the interface identifier. IPv6 assigns this address to a secondary interface for the network interface. The host adds the router as one of its default gateways. In general, there are as many secondary interfaces configured as there are prefixes advertised by the router.

Figure 4-4 shows a general example of Secondary Interface Autoconfiguration.

Figure 4-4 Secondary Interface Autoconfiguration From an IPv6 Router



1. Primary interface comes up with the link-local address autoconfigured.
2. Host multicasts Router Solicitation.
3. IPv6 Router sends Router Advertisement to host.
4. Host autoconfigures secondary interface (`lan0:1`) by prepending prefix (`3ffe:2000:0:13/64`) sent by router to interface identifier (`a00:9ff:fe78:f33`9). Refer to RFC 2461 “Neighbor Discovery for IP Version 6 (IPv6)” for details.

Manual Configuration and Router Advertisements

Note that even if a primary interface is manually configured, if the host receives prefixes from router advertisements, then secondary interfaces are autoconfigured. In this case, the addresses on the secondary interfaces are derived from the interface ID portion of the manually specified primary interface address.

Manual Configuration Overwriting Autoconfiguration

Manual configuration can overwrite autoconfiguration. When a secondary interface is configured with a manually assigned address, and if the user chooses an interface index number that has been used for an already autoconfigured secondary interface, the manual configuration overwrites the autoconfiguration. When this happens, network connectivity through the overwritten autoconfigured IP address is temporarily lost. At a later time, when the host receives the next router advertisement, the host will bring up another secondary interface with a different IP index number, but with the same IP address, and network connectivity through that IP address is restored. Normally, a user can avoid this by checking used IP index numbers. However, there is always a possibility that address autoconfiguration due to router advertisement is happening concurrently while the user manually configures secondary interfaces.

Disabling Specific IPv6 Interfaces

To disable communication through a specific IP address on an autoconfigured secondary interface, that secondary interface must be marked down, not removed or overwritten with a different IP address. If that interface is removed or overwritten, the host will reconfigure another secondary interface with the same IP address when it receives the next router advertisement. Alternatively, the router can be configured to stop advertising the prefix that corresponds to the offending IP address.

Removing An Interface

A primary interface cannot be removed from the system until all secondary interfaces are removed. You can remove secondary interfaces from the system using the `ifconfig inet6` command, as in the following example:

```
ifconfig lan1:1 inet6 ::
```

The primary interface (for example, lan1) can then be removed from the system with the `ifconfig` command, as in the following example:

```
ifconfig lan1 inet6 unplumb
```

A loopback interface does not have a hardware device associated with it. The name of the loopback interface is lo0. A loopback interface is automatically created by the system. You cannot delete it.

Networking Terminology

The following are descriptions of some important IPv6 networking terms.

Node

A node is a device that implements IP on the network. A node can be a host or a router.

A local node (or host) is the computer (or host) where you have logged-in. A remote node is a computer on the IP network where you are not logged in. A remote node does not have to be directly attached to your terminal.

Router

A **router** is a node that forwards IP packets not explicitly addressed to itself. It is a device that can forward packets between two or more IP networks. An IPv6 router can advertise prefixes. IPv6 router guidelines are beyond the scope of this manual. Refer to RFC 2461 for IPv6 router guidelines.

Host

A **host** is any node that is not a router.

Network Interface Name

A **network interface** is a communication device through which messages can be sent and received. An IPv6 address is associated with an interface name. Find the interface name(s) for a network interface by running the `lanscan` command and looking at the "Net-Interface Name PPA" field. For example,

```
lanscan
Hardware Station        Crd Hdw   Net-Interface   NM  MAC       HP-DLPI DLPI
Path     Address        In# State NamePPA         ID  Type      Support Mjr#
2/0/2    0x08000978F339 0   UP    lan0 snap0      1   ETHER     Yes     119
```

The interface name may include a colon (:), followed by an interface index number that denotes the interface number. The interface index number 0 is the first interface number for a card/encapsulation type and is known as the primary interface. The interface name lan0 is equivalent to lan0:0. The syntax is as follows:

`nameX[:interface-index-number]`

In the preceding syntax, *name* is the class of the interface. Valid name is lan (Ethernet LAN). *X* is the Physical Point of Attachment (PPA). *interface-index-number* is the number of the interface.

You must configure the **primary interface** for a LAN card before you can configure subsequent interfaces, known as **secondary interfaces**, for the same card. For example, you must configure lan0 before you configure lan0:1 and lan0:2.

5 IPv6 Software and Interface Technology

The topics discussed in this section concern IPv6 deployment and migration.

Name and Address Lookup for IPv6

It is generally recommended to add IPv6 addresses (known as AAAA records) to a DNS Name Server only when the following conditions are true:

- The IPv6 address is assigned to the interface on the node.
- The address is configured on the interface.
- The interface is on a link which connects to the IPv6 infrastructure.

HP recommends beginning with IPv6 addresses and host names in the `/etc/hosts` file on a development network; then adding IPv6 addresses and hosts to a Domain Name Service when moving IPv6 to a production backbone network.

`/etc/nsswitch.conf` (`nsswitch.conf(4)`) is a configuration file for the name service switch. The `ipnodes` entity specifies which name services resolve IPv4 and IPv6 addresses and host names on HP-UX 11i v2.

More specifically, on HP-UX 11i v2, the keyword `ipnodes` specifies the resolver policy for the library functions `getnameinfo(3N)`, `getaddrinfo(3N)`, `getipnodebyname(3N)` and `getipnodebyaddr(3N)` for both IPv4 and IPv6 addresses. The existing keyword `hosts` specifies the resolver policy for the library functions `gethostbyname()` and `gethostbyaddr()` for IPv4 addresses.

NOTE On HP-UX 11i v2, Internet Services applications (such as `telnet`, r-commands, etc.) use these library functions to resolve IPv4 and IPv6 addresses.

By default, the `/etc/nsswitch.conf` is not on a system. The default `ipnodes` policy (same as default `hosts` policy) is as follows:

```
dns [NOTFOUND=return] files
```

This policy implies that `dns` is the authoritative resolver and will only try `files` if `dns` is down. If `dns` is available but returns `NOTFOUND`, the search stops.

Thus, if DNS has **not** been set up as the definitive source, and files (`/etc/hosts`) may need to be used for address and host name resolution, HP recommends adding the following entry to `/etc/nsswitch.conf`:

```
ipnodes: dns [NOTFOUND=continue] files
```

Or if `/etc/hosts` is to be the primary Name Service

```
ipnodes: files [NOTFOUND=continue] dns
```

NOTE You can not specify `NIS` or `NIS+` on the `ipnodes` entry.

Migrating Name and IPv6 Address Lookup

Most sites test IPv6 on a development subnetwork before deploying it on a larger scale. These sites typically add IPv6 address and host names to the `/etc/hosts` files on IPv6 hosts, then change their hosts lookup policy to search files.

HP recommends that you maintain at least a minimal `/etc/hosts` file that includes important addresses like gateways, root servers, and your host's own IP address. HP also recommends that you include the word `files` in the `hosts` and `ipnodes` lines of `/etc/nsswitch.conf` to help ensure a successful system boot using the `/etc/hosts` file when DNS is not available.

Migrating from IPv4 to IPv6

IPv6 is the next generation Internet protocol and is designed to be a replacement for IPv4. However, it is expected that IPv6 adoption will be gradual and there will be a lengthy transition period during which IPv4 and IPv6 protocols will have to coexist. The IETF (ngtrans working group) has developed a number of transition mechanisms that facilitates IPv6 deployment.

The main goals of these transition mechanisms are to allow newly deployed IPv6 hosts and routers to inter-operate with existing IPv4 hosts and routers and allow isolated IPv6 hosts and routers to communicate with each other using the existing IPv4 infrastructure.

HP-UX 11i v3 IPv6 transport supports the following three transition mechanisms:

- **Dual stack:** This mechanism provides complete concurrent support for both IPv4 and IPv6 protocols in hosts and routers. It allows networks to support both IPv4 applications and IPv6 applications.
- **Tunneling:** Tunneling encapsulates IPv6 packets within IPv4 packets. IPv6 transmission across the IPv4 network is transparent. In HP-UX 11i v3 configured (point-to-point) tunneling is supported. In addition to IP6-in-IP tunneling support, IP6-in-IP6 and IP-in-IP6 tunnels are also supported.
- **"6to4":** Isolated IPv6 nodes and networks can communicate over an IPv4 network, without explicitly configuring tunnels, by using the "6to4" mechanism (RFC 3056). "6to4" effectively treats the IPv4 wide area network as a unicast point-to-point link layer. "6to4" requires no end-node reconfiguration and minimal router configuration.

Tunneling

Tunneling enables IPv6 hosts and routers to connect with other IPv6 hosts and routers over an existing IPv4 network. Dual stack hosts and routers can tunnel IPv6 packets over regions of IPv4 routing topology by encapsulating them within IPv4 packets. The encapsulated packets travel across an IPv4 Internet until they reach their destination host or router. The IPv6-aware host or router decapsulates the IPv6 datagrams, forwarding them as needed. The IPv6 transmission across the IPv4 Internet is transparent. This type of tunneling is referred to as IP6-in-IP.

Tunneling can be used in a variety of ways:

- **Router-to-Router:** IPv6/IPv4 dual stack routers interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. In this case, the tunnel spans one segment of the end-to-end path that the IPv6 packet takes.
- **Host-to-Router:** IPv6/IPv4 dual stack hosts can tunnel IPv6 packets to an intermediary IPv6/IPv4 router that is reachable over an IPv4 infrastructure. This type of tunnel spans the first segment of the packet's end-to-end path.
- **Host-to-Host:** IPv6/IPv4 (dual stack hosts that are interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. In this case, the tunnel spans the entire end-to-end path that the packet takes.
- **Router-to-Host:** IPv6/IPv4 routers can tunnel IPv6 packets to their final destination IPv6/IPv4 host. This tunnel spans only the last segment of the end-to-end path.

The HP-UX 11i v3 IPv6/IPv4 dual stack node can perform the role of the router. It can also continue to perform the role of the host, as it has since the first offering of HP-UX IPv6 transport.

Configured and Automatic Tunneling

Two tunneling techniques are specified in RFC 2893: configured and automatic. The two techniques differ primarily in how the tunnel end-point is determined.

Configured Tunnels are point-to-point tunnels; tunnel configuration must be done on both ends of the tunnel. The tunnel endpoint is determined from the configuration information.

RFC 2893 specifies tunnels as IPv6 interfaces and requires them to be configured with at least (on primary interfaces) link-local addresses. To conform to RFC 2893, tunnels are implemented as IPv6 pseudo-interfaces.

In HP-UX 11i v3 tunnels can be configured (ephemerally) using `ifconfig` and permanently by editing `/etc/rc.config.d/netconf-ipv6`. In general, the following tunnel parameters are relevant in HP-UX 11i v3:

— Tunnel interface name: This is a local identifier name for each tunnel configured. (It need not be the same on both ends of the configured tunnel.) For IP6-in-IP and “6to4” tunnels this would be iptu<#> (e.g. iptu0, iptu1). For IP6-in-IP6 and IP-in-IP6 tunnels it would be ip6tu<#> (e.g. ip6tu0, ipt6u1).

— Tunnel Type: Type of tunnel. Supported tunnels are: “ip6inip”, “6to4”, “ip6inip6”, and “ipinip6”.

— Tunnel entry-point node (local) address: This is the tunnel source address. For tunnel types "ip6inip" and "ip6inip6", it should be a link-local IPv6 address. Example: fe80::1. For tunnel type "ip6inip", if the link-local address is not specified, it will be automatically configured based on the source address in the encapsulating (outer) header. For tunnel type "ipinip6", it should be an IPv4 address. For tunnel type "6to4", it should be a “6to4” address derived from the source address in the encapsulating (outer) header. For example if the source address in the encapsulating (outer) header is 15.13.136.204, the “6to4” prefix should be 2002:0f0d:88cc::, which can be combined to an interface identifier "1" to form the “6to4” address 2002:0f0d:88cc::1.

— Tunnel exit-point node (remote) address: This is the tunnel destination address. For “ip6inip” it will be a link-local IPv6 address configured (automatically if not specified) from the destination address in the encapsulating (outer) header. For “6to4” this value must never be specified, since it will always be automatically determined based on routing information.

— Source address in the encapsulating (outer) header: This must be an address configured on an interface on the tunnel entry-point (local) node. For “ip6inip” and “6to4” it must be an IPv4 address.

— Destination address in the encapsulating (outer) header: This must be an address configured on an interface on the tunnel exit-point (remote) node. For “ip6inip” it must be an IPv4 address. For “6to4” this value must never be specified, since it will automatically be derived from the destination “6to4” address.

— Interface State: Specifies the desired interface state, “up” or “down”. By default it is “up”.

— Interface Flag: Specify interface flag. If set to “-private”, disable stateless address autoconfiguration using prefixes received in router advertisements. Default is “private”, the interface will autoconfigure addresses using prefixes received in router advertisements.

Automatic tunnels are point-to-multipoint tunnels. The IETF is in the process of deprecating automatic tunnels with IPv4-compatible address in favor of "6to4". For more information on "6to4", refer to ""6to4" - Connecting IPv6 Domains over IPv4 Clouds" on page 62 of this guide.

IMPORTANT Automatic tunnels with IPv4-compatible addresses are not supported.

Configured Tunnel IP6-in-IP Tunnel (Host-Host) Example

This section provides an example of how to configure a simple IP6-in-IP configured tunnel between two dual stack hosts both running HP-UX 11i v3.

Figure 5-1 Host-Host Configured Tunnel

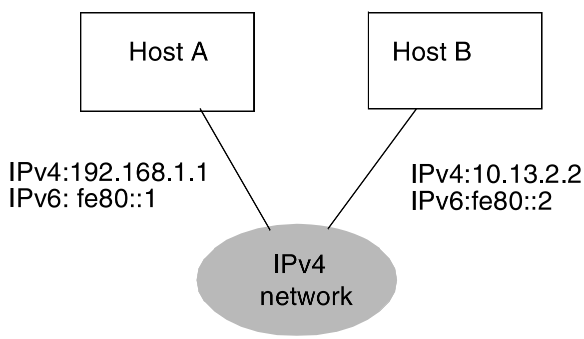


Figure 5-1 illustrates a scenario where you can set up a configured tunnel between Host A and Host B.

On Host A:

— Using `ifconfig` (ephemeral), enter:

```
ifconfig iptu0 inet6 tunnel ip6inip fe80::1 fe80::2 tsrc
192.168.1.1 tdst 10.13.2.2 up
```

— Editing `/etc/rc.config.d/netconf-ipv6` (permanent), add:

```
TUN_INTERFACE_NAME[0]="iptu0"
TUN_TYPE[0]="ip6inip"
TUN_LOCAL_ADDRESS[0]="fe80::1"
```

```
TUN_REMOTE_ADDRESS[0]="fe80::2"
TUN_ENCAP_SRC_ADDRESS[0]="192.168.1.1"
TUN_ENCAP_DST_ADDRESS[0]="10.13.2.2"
TUN_INTERFACE_STATE[0]="up"
```

On Host B:

— Using `ifconfig` (ephemeral), enter:

```
ifconfig iptu0 inet6 tunnel ip6inip fe80::2 fe80::1 tsrc
10.13.2.2 tdst 192.168.1.1 up
```

— Editing `/etc/rc.config.d/netconf-ipv6` (permanent), add:

```
TUN_INTERFACE_NAME[0]="iptu0"
TUN_TYPE[0]="ip6inip"
TUN_LOCAL_ADDRESS[0]="fe80::2"
TUN_REMOTE_ADDRESS[0]="fe80::1"
TUN_ENCAP_SRC_ADDRESS[0]="10.13.2.2"
TUN_ENCAP_DST_ADDRESS[0]="192.168.1.1"
TUN_INTERFACE_STATE[0]="up"
```

Configured IP6-in-IP6 Tunnel (Host-Host) Example

This section provides an example of how to configure a host-host IP6-in-IP6 configured tunnel. IP6-in-IP6 tunnel configuration allows transmission of IPv6 packets encapsulated in an IPv6 header.

On Local Host:

— Using `ifconfig` (ephemeral), enter:

```
ifconfig ip6tu0 inet6 tunnel ip6inip6 fe80::1 fe80::2 tsrc
2ffe::1 tdst 3ffe::1 up
```

— Editing `/etc/rc.config.d/netconf-ipv6` (permanent), add:

```
TUN_INTERFACE_NAME[0]="ip6tu0"
TUN_TYPE[0]="ip6inip6"
TUN_LOCAL_ADDRESS[0]="fe80::1"
TUN_REMOTE_ADDRESS[0]="fe80::2"
TUN_ENCAP_SRC_ADDRESS[0]="2ffe::1"
TUN_ENCAP_DST_ADDRESS[0]="3ffe::1"
TUN_INTERFACE_STATE[0]="up"
```

On Remote Host:

— Using `ifconfig` (ephemeral), enter:

```
ifconfig ip6tu0 inet6 tunnel ip6inip6 fe80::2 fe80::1 tsrc
3ffe::1 tdst 2ffe::1 up
```

— Editing `/etc/rc.config.d/netconf-ipv6` (permanent), add:

```
TUN_INTERFACE_NAME[0]="ipt6u0"
TUN_TYPE[0]="ip6inip6"
TUN_LOCAL_ADDRESS[0]="fe80::2"
TUN_REMOTE_ADDRESS[0]="fe80::1"
TUN_ENCAP_SRC_ADDRESS[0]="3ffe::1"
TUN_ENCAP_DST_ADDRESS[0]="2ffe::1"
TUN_INTERFACE_STATE[0]="up"
```

Configured IP-in-IP6 Tunnel (Host-Host) Example

This section provides an example of how to configure a host-host IP-in-IP6 configured tunnel. IP-in-IP6 tunnel configuration allows transmission of IPv4 packets encapsulated in an IPv6 header.

On Local Host:

— Using `ifconfig` (ephemeral), enter:

```
ifconfig ip6tu0 inet tunnel ipinip6 10.10.10.1 15.15.15.2
tsrc 2ffe::1 tdst 3ffe::1 up
```

— Editing `/etc/rc.config.d/netconf-ipv6` (permanent), add:

```
TUN_INTERFACE_NAME[0]="ip6tu0"
TUN_TYPE[0]="ipinip6"
TUN_LOCAL_ADDRESS[0]="10.10.10.1"
TUN_REMOTE_ADDRESS[0]="15.15.15.2"
TUN_ENCAP_SRC_ADDRESS[0]="2ffe::1"
TUN_ENCAP_DST_ADDRESS[0]="3ffe::1"
TUN_INTERFACE_STATE[0]="up"
```

On Remote Host:

— Using `ifconfig` (ephemeral), enter:

```
ifconfig ip6tu0 inet tunnel ipinip6 15.15.15.2 10.10.10.1
tsrc 3ffe::1 tdst 2ffe::1 up
```

— Editing `/etc/rc.config.d/netconf-ipv6` (permanent), add:

```
TUN_INTERFACE_NAME[0]="ip6tu0"
TUN_TYPE[0]="ipinip6"
TUN_LOCAL_ADDRESS[0]="15.15.15.2"
TUN_REMOTE_ADDRESS[0]="10.10.10.1"
TUN_ENCAP_SRC_ADDRESS[0]="3ffe::1"
TUN_ENCAP_DST_ADDRESS[0]="2ffe::1"
TUN_INTERFACE_STATE[0]="up"
```

NOTE Refer to the `ifconfig(1m)` man page and the `/etc/rc.config.d/netfconf-ipv6` file for more detailed information on tunneling parameters.

"6to4" - Connecting IPv6 Domains over IPv4 Clouds

"6to4" is an automatic tunneling mechanism that can be used to provide connectivity between isolated IPv6 domains or hosts across an IPv4 infrastructure and with native IPv6 domains via relay routers. "6to4" is based on the IP6-in-IP tunneling mechanism defined in RFC 2893 and it falls under the router-to-router tunneling scenario.

"6to4" uses the concept of automatic tunneling where the tunnel end-point is determined from the IPv6 destination address and avoids the complexity of manual tunnel configuration. It does not use the IPv4-compatible address, but instead determines the tunnel endpoint IPv4 address from the special "6to4" prefix of the IPv6 destination address.

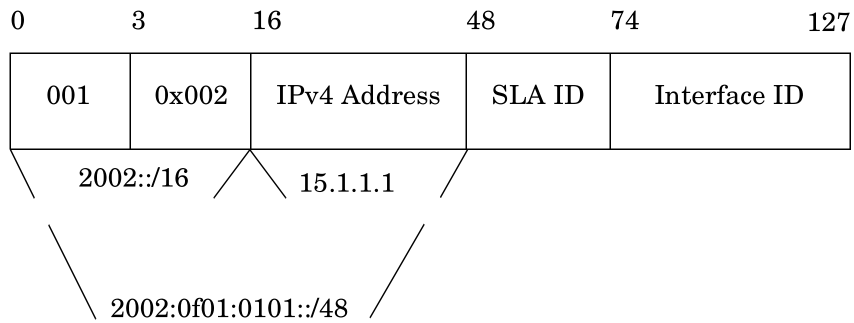
"6to4" Well-Known Prefix

"6to4" defines an address assignment scheme that allows a site to obtain a unique externally routable prefix if the site has at least one globally unique IPv4 address. The Internet Assigned Number Authority (IANA) has assigned the unique IPv6 address prefix of `2002::/16` for "6to4". Each site must have a border dual stack router that has at least one global IPv4 address.

A "6to4" prefix can be generated by concatenating the `2002::` prefix to the global IPv4 address. For example, if the dual stack router has an IPv4 address 15.1.1.1, then its "6to4" prefix will be 2002:0f01:0101::/48.

The "6to4" prefix provides a network prefix for the local IPv6 host or network. The IPv4 address is the endpoint for all external IPv4 connections.

Figure 5-2 "6to4" Prefix



"6to4" Encapsulation

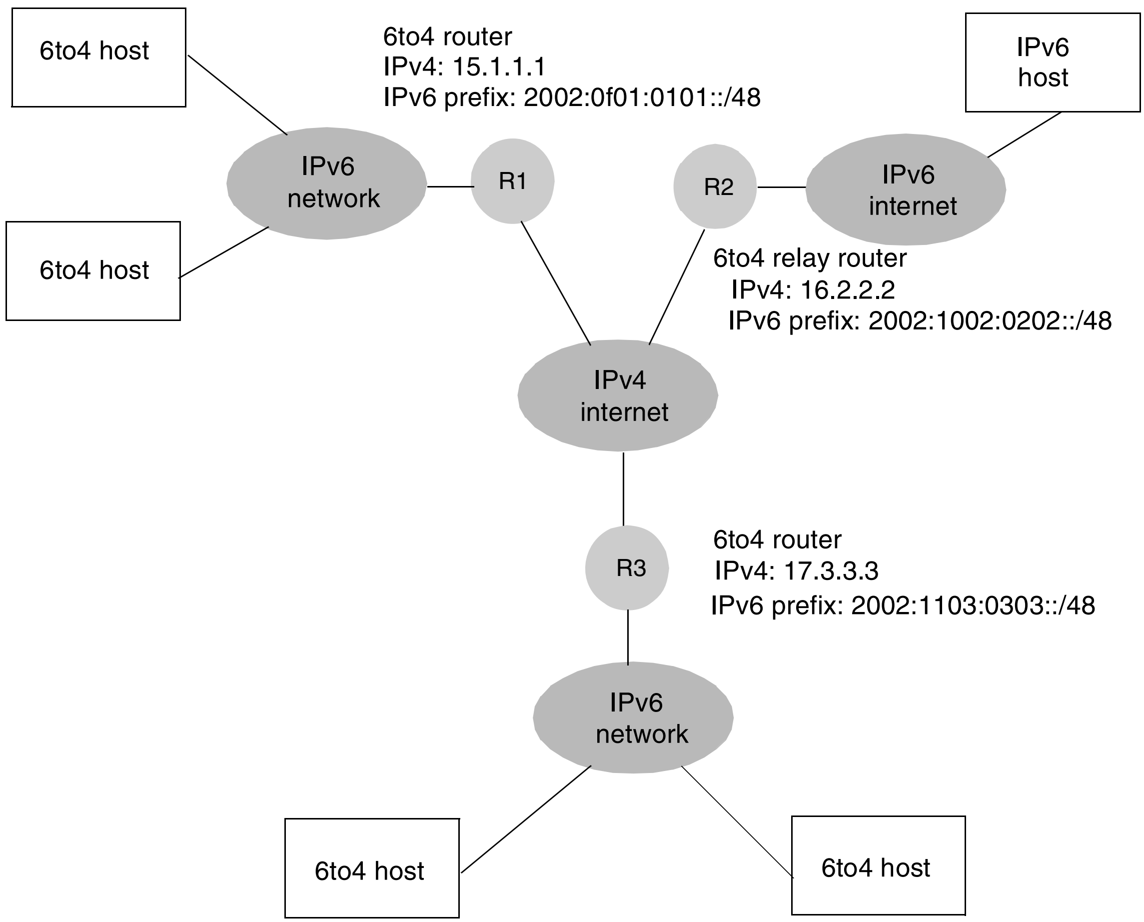
IPv6 packets from a "6to4" site are encapsulated in IPv4 packets when they leave the site over its external IPv4 connection. IPv6 packets are transmitted in IPv4 packets with an IPv4 protocol type of `41`, the same protocol type set when IPv6 packets tunnel inside IPv4 frames.

"6to4" Topology Example

"6to4" topology consists of: one or more "6to4" hosts in a "6to4" domain; at least one "6to4" router in the domain that has at least one IPv4 connection to the Internet; and a "6to4" relay router that is used to connect to a native IPv6 domain.

Nodes can perform the role of the “6to4” router. Host configuration continues to be supported, as was the case in base (default) HP-UX 11i v2.

Figure 5-3 **“6to4” Topology Example**



As shown in the “6to4” topology example of Figure 5-3:

6to4 host: An IPv6 host that has at least one “6to4” address configured. The "6to4" address is autoconfigured using the "6to4" prefix advertised by the "6to4" router. It has a default route to the "6to4" router. All non-local "6to4" addressed packets and native IPv6 packets are sent to the "6to4" router.

"6to4" router: An IPv6/IPv4 border router that forwards "6to4" addressed traffic between "6to4" hosts within a site and other "6to4" routers or to "6to4" relay routers across IPv4 internet. "6to4" routers need to have at least one public IPv4 address and the "6to4" prefix is derived from the public IPv4 address. The "6to4" router advertises the "6to4" prefix on its attached link. It performs the encapsulation and decapsulation functions.

"6to4" relay router: An IPv6/IPv4 router that performs the functions of the "6to4" router and forwards "6to4" addressed traffic between "6to4" routers on the IPv4 internet and IPv6 hosts on the IPv6 internet.

An IPv6 interior routing protocol, such as routing information protocol next generation (RIPng), is used for routing IPv6 in a "6to4" domain. IPv4 exterior routing protocol handles the routing of tunneled IPv4 packets between "6to4" routers and relay routers. In addition, for forwarding native IPv6 addressed packets, a default route from the "6to4" router to the relay router can be setup or IPv6 exterior routing protocol can be used between "6to4" routers and relay routers.

“6to4” Security Considerations By default, “6to4” routers and relay routers accept and decapsulate traffic from any source. This potentially allows malicious parties to get around access controls and spoof addresses, to perform denial of service attacks. Before setting up a tunnel from a “6to4” router to an external “6to4” relay router, review the internet draft *Security Considerations for 6to4* at http://www.ietf.org.

Configuration Example: For the topology example shown in Figure 5-3, the following sample `ifconfig` commands will (ephemerally) configure the HP-UX 11i v2 dual stack routers to handle “6to4”:

On R1: `ifconfig iptu0 inet6 tunnel 6to4 tsrc 15.1.1.1`

On R2: `ifconfig iptu1 inet6 tunnel 6to4 tsrc 16.2.2.2`

On R3: `ifconfig iptu2 inet6 tunnel 6to4 tsrc 17.3.3.3`

In all three cases, you do not need to explicitly specify the address of the tunnel entry point, as this special "6to4" address will be automatically created based on the globally unique IPv4 address that will be the source address in the encapsulating (outer) header.

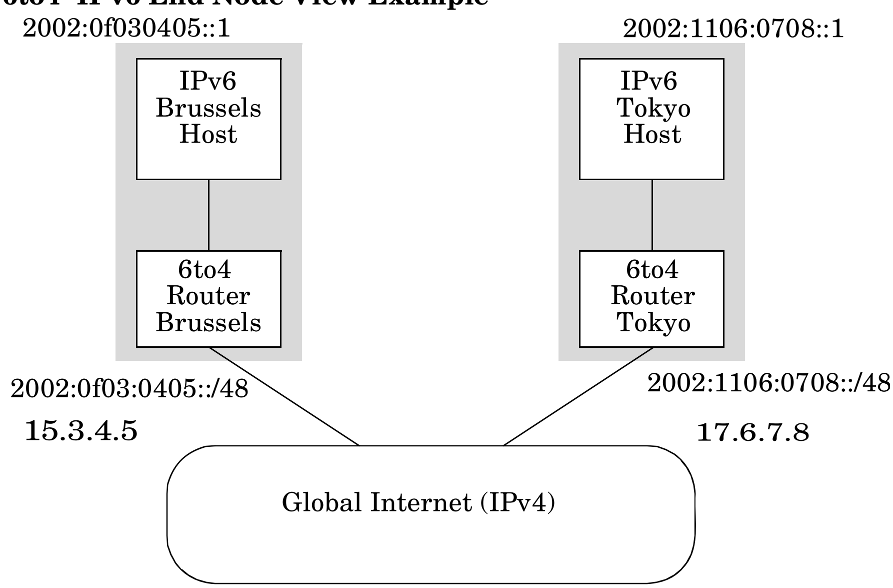
To configure R1 using the `/etc/rc.config.d/netconf-ipv6` file, add the following lines:

```
TUN_INTERFACE_NAME[0]=”iptu0”
TUN_TYPE[0]=”6to4”
TUN_LOCAL_ADDRESS[0]=””
TUN_REMOTE_ADDRESS[0]=””
TUN_ENCAP_SRC_ADDRESS[0]=”15.1.1.1”
TUN_ENCAP_DST_ADDRESS[0]=””
TUN_INTERFACE_STATE[0]=”up”
```

“6to4” End-Node View Example

Figure 5-4 shows two IPv6 subnetworks. The end nodes have their routers’ globally unique IPv4 addresses embedded in their network prefixes. The routers have “6to4” addresses and corresponding globally unique IPv4 addresses. From the IPv6 end-node view, each host’s subnetwork is connected to the other’s through a "6to4" router. All IPv4 tunneling is transparent to the IPv6 end nodes.

Figure 5-4 **"6to4" IPv6 End Node View Example**



Using rtradvd to Advertise "6to4" Routing Prefix

This section provides a simple example to show how to advertise "6to4" routing. In this example, the globally unique IPv4 address of the host is 15.13.1.2.

```
#example beginsdefaults {
        AdvSendAdvertisement on ;
 };

  interface lan0 {
        prefixinfo 2002:f0d:0101::/64 {
        };

 };

#example ends
```

For more examples, refer to the rtradvd.conf(4) man page.

6 Utilities

HP-UX 11i v3 IPv6 transport, for the most part uses IPv6-enhanced IPv4 network utilities. This section summarizes the utilities required for administration of HP-UX 11i v3 IPv6 transport.

Configuration Utilities

This section describes configuration utilities available to configure HP-UX 11i v3 IPv6 transport.

The ifconfig "inet6" Address Family

`Use ifconfig` to assign an IPv6 address to an interface and configure parameters, such as the network prefix. (In IPv6, prefix replaces netmask.)

The `ifconfig keyword` **inet6** required to configure IPv6 interfaces. It is not required to examine IPv6 interfaces. Refer to the `ifconfig(1M)` man page for details.

Neighbor Discovery Protocol Replaces arp in IPv6

The Neighbor Discovery Protocol (`ndp`) replaces `arp` in IPv6. Refer to "Neighbor Discovery" on page 43 in Chapter 4 for details.

The route "inet6" Option

`route(1M)` adds and deletes entries to the network routing table, allowing your system to communicate through a router. In IPv6, routing entries are automatically added when router advertisements are received from an IPv6 router. A configured tunnel route can be added using the `route` utility. Refer to `route(1m)` for more information.

Network Diagnostic Utilities

This section lists network diagnostic utilities available for use as part of the process of administering HP-UX 11i v2 IPv6.

- `lanadmin(1M)` resets or reports status of the LAN card.
- `lanscan(1M)` displays LAN device configuration and status.
- `ndd(1M)` displays and modifies network driver parameters.
- `ndp(1M)` displays and modifies the IPv6 neighbor discovery cache.
- `netstat(1)` provides network statistics and information about network connections.
- `ping(1M)` verifies network connectivity through the Network Layer and reports round-trip time of communication time between hosts.
- `traceroute(1M)` traces the path between hosts at the Network Layer.

IPv6 Additions to Network Tracing and Logging

Use `nettl` to trace traffic through IPv6 Subsystems, or use `nettladm`. Table 6-1 below lists the subsystems available for IPv6 packet tracing.

Table 6-1 Network Trace Subsystems

Description	Subsystem Name
IPv6 Packets	NS_LS_IPV6
ICMPV6 Packets	NS_LS_ICMPV6
IPv6 Loopback packets	NS_LS_LOOPBACK6

Use `netfmt` to format trace records captured by `nettl` from the IPv6 subsystems. `netfmt` can also filter `nettl` output according to the following IPv6 criteria:

Table 6-2 IPv6 Network Filter Criteria

Filter Description	Entry in the netfmt configuration file
IPv6 Packets	NS_LS_IPV6
ICMPV6 Packets	NS_LS_ICMPV6
IPv6 Source Address	`filter ip6_saddr ::abcd` where ::abcd is the source address
IPv6 Destination Address	`filter ip6_daddr ::fedc` where ::fedc is the destination address
Connection per port and IPv6 address	filter connection6 <local_IPv6addr> | <port> <remote_IPv6addr | port

Contacting Your HP Representative

If you do not have a service contract with HP, you may follow the procedure described below, but you will be billed accordingly for time and materials.

If you have a service contract with HP, document the problem as a Service Request (SR) and forward it to your HP representative. Include the following information where applicable:

- A characterization of the problem. Describe the events and symptoms leading up to the problem. Attempt to describe the source of the problem.

  Your characterization should include: HP-UX commands; communication subsystem commands; functionality of user programs; result codes and messages; and data that can reproduce the problem.

- Obtain the version, update, and fix information for all software.

  To determine the version of your HP-UX Operating System, execute the command:

  ```
  uname -a >> /tmp/filename
  ```

  This allows HP to determine if the problem is already known, and if the correct software is installed at your site.

- Illustrate as clearly as possible the context of any message(s). Record all error messages and numbers that appear at the user terminal and the system console.

- Prepare a listing of the HP-UX I/O configuration you are using for your HP representative to further analyze.

- Try to determine the general area within the software where you think the problem exists. Refer to the appropriate reference manual and follow the guidelines on gathering information for that product.

- Document your interim, or “workaround,” solution. The cause of the problem can sometimes be found by comparing the circumstances in which it occurs with the circumstances in which it does not occur.

- Create copies of any Internet Services or HP-UX 11i v3 IPv6 for software link trace files that were active when the problem occurred, for your HP representative to further analyze.
- In the event of a system failure, obtain a full memory dump. If the directory `/var/adm/crash` exists, the HP-UX utility `/usr/sbin/savecore` automatically executes during reboot to save the memory dump. HP recommends that you create the `/var/adm/crash` directory after successfully installing this product. Send the output of your system failure memory dump to your HP representative.
- Prepare copies of the name service files such as `/etc/hosts`, etc/`nsswitch.conf`, `named.conf` and `resolv.conf`. Prepare a copy of the IPv6 configuration file /`etc/rc.config.d/netconf-ipv6`.
- Verify the software: `/usr/sbin/swverify > /tmp/swv-out`
- Execute the `display` command of the `lanadmin` diagnostic on the LAN interface and record the output.
- Record the troubleshooting flowchart number and step number where you are unable to resolve the problem.
- Save all network log files. Make sure that ERROR and DISASTER log classes are enabled when STREAMS subsystem log files are collected in `/var/adm/nettl.LOG000`.
- Execute the following commands and record the output:

```
uname -a >> /tmp/filename
what /stand/vmunix >> /tmp/filename
lanscan >> /tmp/filename
netstat -sf inet6 >> /tmp/filename
netstat -inf inet6 >> /tmp/filename
netstat -rnf inet6 >> /tmp/filename
ndp -an >> /tmp/filename
ndd -get /dev/tcp tcp_status >> /tmp/filename
ndd -get /dev/ip6 ip6_ill_status >> /tmp/filename
ndd -get /dev/ip6 ip6_ipif_status >> /tmp/filename
ndd -get /dev/ip6 ip6_ire_status >> /tmp/filename
ndd -get /dev/ip6 ip6_ill_config_status >> /tmp/filename
```

  Prepare the formatted output (use `netfmt`) and a copy of the log file for your HP representative to further analyze.

A IPv6 ndd Tunable Parameters

The following IPv6 tunable parameters allow advanced fine-tuning of HP-UX 11i v2 IPv6 performance.

Supported IPv6-related ndd parameters

To obtain a list of supported IPv6-related `ndd` parameters, enter:

```
ndd -h supported | grep ip6
```

NOTE

For more information on a specific parameter (if help text is provided for that parameter), enter:

```
ndd -h <parameter>
```

Table A-1 lists the output received from entering,
"ndd -h supported | grep ip6":

Table A-1 Supported IPv6 ndd parameters

Parameter	Description
IPv6:	
ip6_def_hop_limit	- Controls the default Hop Limit in the IPv6 header
ip6_forwarding	- Controls how IPv6 hosts forward packets
ip6_fragment_timeout	- Controls how long IPv6 fragments are kept
ip6_icmp_interval	- Limits the sending rate of ICMPv6 error messages
ip6_ill_status	- Displays a report of all IPv6 physical interfaces
ip6_ipif_status	- Displays a report of all IPv6 logical interfaces
ip6_ire_cleanup_interval	- Timeout interval for purging IPv6 routing entries

Table A-1 Supported IPv6 ndd parameters (Continued)

Parameter	Description
ip6_ire_hash	- Displays all IPv6 routing table entries, in the order searched when resolving an IPv6 address
ip6_ire_pathmtu_interval	- Controls the probe interval for IPv6 PMTU
ip6_ire_redirect_interval	- Controls IPv6 'Redirect' routing table entries
ip6_ire_status	- Displays all IPv6 routing table entries
ip6_raw_status	- Reports IPv6 level RAWIP fanout table
ip6_reass_mem_limit	- Maximum number of bytes for IPv6 reassembly
ip6_send_redirects	- Sends ICMPv6 'Redirect' packets
ip6_tcp_status	- Reports IPv6 level TCP fanout table
ip6_udp_status	- Reports IPv6 level UDP fanout table
IPV6 Neighbor Discovery (ND):	
ip6_ire_reachable_interval	- Controls the ND REACHABLE_TIME
ip6_max_random_factor	- Controls the ND MAX_RANDOM_FACTOR
ip6_min_random_factor	- Controls the ND MIN_RANDOM_FACTOR
ip6_nd_advertise_count	- Controls the ND MAX_NEIGHBOR_ADVERTISEMENT
ip6_nd_anycast_delay	- Controls the ND MAX_ANYCAST_DELAY_TIME
ip6_nd_dad_solicit_count	- Controls the number of duplicate address

Table A-1 Supported IPv6 ndd parameters (Continued)

Parameter	Description
ip6_nd_multicast_solicit_ count	- Controls the ND MAX_MULTICAST_SOLICIT
ip6_nd_probe_delay	- Controls the ND DELAY_FIRST_PROBE_TIME
ip6_nd_transmit_interval	- Controls the ND RETRANS_TIMER
ip6_nd_unicast_solicit_ count	- Controls the ND MAX_UNICAST_SOLICIT
ip6_rd_solicit_count	- Controls the ND MAX_RTR_SOLICITATIONS
ip6_rd_solicit_delay	- Controls the ND MAX_RTR_SOLICITATIONS_DELAY
ip6_rd_transmit_interval	- Controls the ND RTR_SOLICITATION_INTERVAL
RAWIP6:	
rawip6_def_hop_limit	- Controls the default Hop Limit in the IPv6 header

Index