

Software Release 2.2.2

For Rapier Switches, AR300 and AR700 Series Routers, and AR800 Series Modular Switching Routers

Introduction	2
Hardware platforms	2
New Hardware Models	2
Existing Hardware Models	4
Overview of Software Release 2.2.2	5
Rapier Layer 3 Switches and AR800 Modular Switching Routers	5
AR300 Series Routers	6
Rapier 48	6
Accessing the GUI	6
Virtual Local Area Networks (VLANs)	6
Port Trunking	7
Switching	7
Multicast Mode	7
Interaction with the AT-39	8
Port Security	8
MAC Address Learn Limit	8
Spanning Tree Protocol (STP)	8
Layer 3 Filtering	8
Internet Protocol Version 6 (IPv6)	9
IP Multicasting	9
PIM Dense Mode	9
IGMP Traffic and VLAN Ports	11
Pinging	11
DVRMP	11
Firewall	11
Secure Shell Access	11
Multicast Packet Handling	12
Multiple VLAN Interfaces	12
TCP SYN ACK Retransmission	12
Frame Relay	13
Asynchronous Call Control (ACC)	13
Hypertext Transfer Protocol (HTTP)	13
IPsec	13
ISAKMP Signature authentication	13
ISAKMP Aggressive mode	13
ISAKMP XAUTH.	13
Open Shortest Path First (OSPF)	14
Point-to-Point Protocol (PPP)	14
PPPoE	14
Telephony (PBX)	14
Public Key Infrastructure (PKI)	14
Errata: 2.2.1 Software Reference	15
Availability	15
Installation	15

Introduction

Allied Telesyn International announces the release of Software Release 2.2.2 on new and existing models of Rapier Layer 3 managed switches, AR800 Series modular switching routers, AR 300 and AR700 Series routers. This release note describes the new software features and enhancements in Software Release 2.2.2 since Software Release 2.2.1. It should be read in conjunction with the relevant Quick Install Guides, Quick Start Guides, User Guides, Hardware References and Software References for your switch or router, and the release notes for previous releases. These documents can be found on the Documentation and Tools CD-ROM packaged with your switch or router, or on the support site for your switch or router:

- www.alliedtelesyn.co.nz/support/rapier/ or
- www.alliedtelesyn.co.nz/support/ar800/ or
- www.alliedtelesyn.co.nz/documentation/documentation.html

The most significant new features in Software Release 2.2.2 are:

- Support for the new Rapier 48 Layer 3 switch, including a GUI for management
- IPv6 on AR800 modular switching routers and Rapier Layer 3 switches
- PIM Dense Mode on AR800 modular switching routers and Rapier Layer 3 switches.



WARNING: Information in this release note is subject to change without notice and does not represent a commitment on the part of Allied Telesyn International. While every effort has been made to ensure that the information contained within this document and the features and changes described are accurate, Allied Telesyn International can not accept any type of liability for errors in, or omissions arising from the use of this information.

Hardware platforms

This section lists the router and switch hardware platforms supported by Software Release 2.2.2, and the expansion options available.

New Hardware Models

Rapier 48

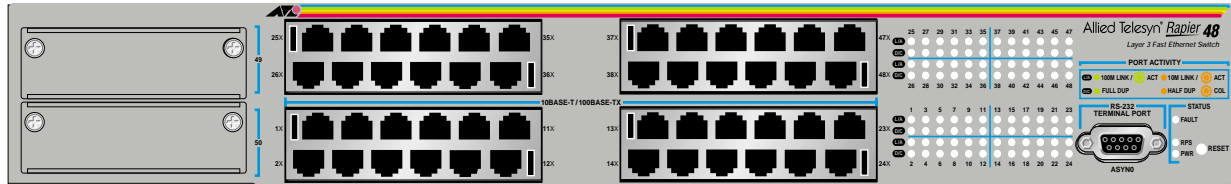
Allied Telesyn International announces the release of the Rapier 48 switch.

Rapier Layer 3 switches deliver wire speed Layer 2 and Layer 3 switching with low-latency high-bandwidth traffic capabilities. Ethernet, fast Ethernet and gigabit Ethernet connectivity make the Rapier and AR800 Series versatile and powerful switching solutions. The range of models allows users to choose the port connectors, expansion options and advanced features required for their networks.

The Rapier 48 provides:

- 48-port 10BaseT/100BaseTX (RJ-45 connectors)
- Two 1000Base Uplink Module bays.

Figure 1: Front panel of the Rapier 48.



Features the Rapier 48 shares with other models in the Rapier Series include:

■ Dimensions

- Height = 66 mm (plus 5.5 mm if the rubber feet are used)
- Width = 440 mm (excluding rack-mounting brackets)
- Depth = 360 mm (including NSM face-plate)
- Weight = Not more than 7 kg, depending on model (excluding NSMs, PICs, and power cord)

■ Mounting System

- 1.5U rack mounting

■ Environmental Conditions

- Operating temperature range: 0 to 40° C (32 to 104° F)
- Storage temperature range: -25 to 70° C (-13 to 158° F)
- Relative humidity range: 5 to 95% non-condensing

■ Regulatory Standards

- EMC: CISPR22 class A, FCC class A, and VCCI class I
- Immunity testing to EN50082 levels 2 (ESD), 3 (susceptibility), 4 (fast transients), 5 (power surge), and 6 (RF immunity)
- Safety: UL1950, CSA22.2, EN60950

■ LEDs

- Ethernet port and System status LEDs.

■ Power Supply Unit

AC models

- Universal 110/240 VAC 50/60 Hz input
- Redundant DC Power connection

DC models

- 48 V DC (39-60 V DC is acceptable)
- Accepts positive or negative earthing (grounding)

■ Switching Core

- Broadcom BCM5600 (Rapier 8/8MT, 8/8SC, 16F/MT, 16F/SC, 24, 48)
- Broadcom BCM5680 (Rapier G6, G6F/LX, G6F/SX, G6F/MT)
- L2 and L3 IP Switching

- Processing Core
 - 200 MHz RISC Processor
 - 32 MBytes Synchronous DRAM
 - 6 MBytes FLASH memory
 - 128 KBytes Non-volatile Storage (battery backed SRAM)
- Asynchronous Serial Port
 - Up to 115 kbps
 - Standard DB9 female RS-232 connector
 - Hardware-flow control
- Uplink Module Bays
 - 2 very high performance bays
 - Support for 1 gigabit Ethernet Uplink Modules

PCI Accelerator Cards (PACs)

Software Release 2.2.2 supports the new PCI Accelerator Cards, the AR060 EPAC and AR061 ECPAC, on AR740 routers, AR800 Series switching routers and Rapier Series switches (excluding the Rapier 48). PACs use PCI (Peripheral Component Interconnect) technology to provide high-performance hardware assisted encryption and/or compression functions. This capability results in higher data throughput and improved switch response times. The AR060 EPAC provides encryption, and the AR061 ECPAC provides both encryption and compression.

Mini Accelerator Card (ECMAC v2)

Software Release 2.2.2 also supports the new version 2 Mini Accelerator Card, the AR013 ECMAC, on AR300 Series and AR700 Series routers. Like the ECPAC, the ECMAC provides high-performance hardware assisted encryption and compression functions.

Existing Hardware Models

The following existing switch hardware platforms are also supported by Software Release 2.2.2, and are described in the corresponding release notes, available from www.alliedtelesyn.co.nz. The Rapier Layer 3 Gigabit switches have an NSM bay for WAN expansion options. The AR800 Series Modular Switching Routers include an NSM bay for WAN expansion options, and software support for advanced features.

- Software Release 2.2.1
 - AT-AR041 — 4 Port BRI Network Service Module
 - AT-AR042 — 8 Port BRI Network Service Module
- Software Release 2.1.5
 - AT-A39/T 1 Gb Port Uplink module
- Software Release 2.1.4
 - Rapier G6 Layer 3 Switch
 - Rapier G6F/LX Layer 3 Gigabit Switch
 - Rapier G6F/SX Layer 3 Gigabit Switch
 - Rapier G6F/MT Layer 3 Gigabit Switch

- Software Release 2.1.3
 - Rapier 16F/SC Layer 3 Gigabit Switch
 - Rapier 16F/MT Layer 3 Gigabit Switch
 - Rapier 8/8SC Layer 3 Gigabit Switch
 - Rapier 8/8MT Layer 3 Gigabit Switch
 - AR824 Modular Switching Router
 - AR816F/SC Modular Switching Router
 - AR816F/MT Modular Switching Router
 - AT-AR040 4 PIC Bay Network Service Module
- Software Release 2.1.2
 - Rapier 24 Layer 3 Gigabit Switch
 - AT-A35/SX Uplink Module
 - AT-A35/LX Uplink Module

The following router models are supported by Software Release 2.2.2 and are described in previous release notes, available from www.alliedtelesyn.co.nz:

- AR740 Router — Software Release 2.0.2
- AR720 Router — Software Release 1.8.1
- AR300 Series routers

Overview of Software Release 2.2.2

Software Release 2.2.2 supports the switch and router models list in “*Hardware platforms*” on page 2 with access via the command line interface and SNMP. All Rapier and AR800 Series models are also supported with a web-based graphical user interface.

Rapier Layer 3 Switches and AR800 Modular Switching Routers

The following features are new on Rapier Layer 3 switches and AR800 modular switching routers:

- IPv6
- PIM Dense Mode

PIM Dense Mode is automatically enabled on AR800 Series switching routers. The Rapier FL3 Upgrade licence AT-RPFL3Upgrade may be purchased to provide full Layer 3 features on Rapier switches. This licence now includes PIM Dense Mode.

IPv6 requires a special feature licence on both AR800 switching routers and Rapier switches.

AR300 Series Routers

The following features are new on AR300 Series routers since Software Release 2.2.1:

- IPv6.

IPv6 requires a special feature licence on both AR300 and AR700 routers.

Rapier 48

Accessing the GUI

To access the Rapier 48 GUI:

1. Connect the switch's cables, as described in the Quick Install Guide, which is available on the CD-ROM shipped with your switch, or from www.alliedtelesyn.co.nz.
2. Connect to the switch via a VT100-compatible terminal or the COM port of a PC, and log in as manager, as described in your switch's Quick Install Guide.
3. Enable IP and assign an IP address (for example, 192.168.1.1) to the switch's default VLAN interface (vlan1), using the commands:

```
ENABLE IP
ADD IP INTERFACE=vlan1 IP=ipaddress
```

where *ipaddress* is a valid IP address in dotted decimal notation.

4. If the PC from which you will access the GUI is on a different subnet to the switch, add a route from the PC to the switch, using the command:

```
ADD IP ROUTE=PC-ipaddress INTERFACE=vlan1
NEXTHOP=switch-ipaddress
```

5. Using Internet Explorer 5.0 or higher, or Netscape 4.7 or higher, connect to the switch at:

```
http://ipaddress
```

Use the buttons on the GUI pages to navigate, not your browser's buttons, to ensure that the configuration settings are saved correctly.

Virtual Local Area Networks (VLANs)

A Virtual LAN is a software-defined broadcast domain. The switch's VLAN feature allows the network to be segmented by software management, improving network performance. Workstations, servers, and other network equipment connected to the switch can be grouped according to similar data and security requirements. Several VLANs can be connected to the same switch. The Rapier 48 supports a maximum of 32 IP interfaces.

For more information about VLANs, see the *Switching* chapter of the Software Reference, which is available on the CD-ROM shipped with your switch or from www.alliedtelesyn.co.nz.

Port Trunking

Port trunking, also known as port bundling or link aggregation, allows a number of ports of equal speed to be configured to join together to make a single logical connection of higher bandwidth. This can be used where a higher performance link is required, and makes links even more reliable.

The Rapier 48 switch supports up to 6 trunk groups, of up to 8 switch ports each. Optimal performance is gained by trunking ports within the sets 1-24 or 25-48, and not mixing ports from the two sets. The two gigabit ports may also be trunked together. Ports in trunk groups do not have to be contiguous.

Port trunking is supported between AR800 Series and Rapier switches, and is compatible with trunking algorithms on third party devices.

For more information about Port Trunking, see the *Switching* chapter of the Software Reference, which is available on the CD-ROM shipped with your switch or from www.alliedtelesyn.co.nz.

Switching

Multicast Mode

A new parameter, MULTICASTMODE, has been added to the SET SWITCH PORT command for Rapier switches and AR800 switching routers. This parameter enables users to specify how the switch handles multicast traffic. More information about multicast group membership can be found in “*Internet Group Management Protocol (IGMP)*” of the *IP Multicasting* chapter of the Software Reference, which is available on the CD-ROM shipped with your router, or from www.alliedtelesyn.co.nz. For VLAN-aware devices, such as the Rapier and AR800 Series, multicast group membership is VLAN-based, but IGMP snooping identifies which ports in the VLAN belong to the multicast group.

The SET SWITCH PORT command syntax is now:

```
SET SWITCH PORT={port-list|ALL} [MULTICASTMODE={A|B|C}]  
[other-options...]
```

The MULTICASTMODE parameter indicates how the switch handles traffic addressed to a multicast group to which the specified port or list of ports belongs. If A is specified, all traffic is flooded on all ports on the VLAN, irrespective of whether or not the ports have joined the multicast group. The effect of this option is to disable IGMP snooping without disabling IGMP. If B is specified, the traffic is only sent to those ports which have joined the multicast group, unless no ports have joined, in which case the traffic is flooded on all ports on the VLAN. If C is specified, the traffic is only sent to those ports which have joined the multicast group; if no ports have joined the group, the traffic is discarded. This option allows the manager more control over who receives the traffic. The default is B.

This feature is also supported by Software Release 2.2.1. In Software Release 2.2.2, the CREATE CONFIG command now correctly sets the multicast mode for each port, instead of setting it for groups of 8 ports.

Interaction with the AT-39

In some circumstances the AT-39 gigabit copper uplink card would take a long time (of the order of minutes) to bring up a link to its link partner. This issue has been corrected and the link up time is now less than 5 seconds.

A patch to correct this issue is available for Software Release 2.2.1 (patch 86221-01), and can be downloaded from the Software Updates area of the Allied Telesyn web site at www.alliedtelesyn.co.nz/support/updates/patches.html. Patches do not require a licence or password.

Port Security

If the state of the switch is saved in a configuration script when a port is in the “locked” state for the Port Security feature, that port will now re-enter the “locked” state when the script is executed.

If two PCs are pinging each other, and are in different VLANs on different ports of the switch, then turning on the Port Security feature on one of the two ports may cause the switch to erroneously receive a packet with the switch’s MAC address in the packet’s source MAC address field. Such packets are now discarded by the CPU.

MAC Address Learn Limit

Rapier and AR800 switches can now correctly generate an SNMP trap when the MAC address learn limit is exceeded.

The learn limit is specified by the LEARN parameter of the SET SWITCH PORT command, and is NONE or a number between 0 and 256. NONE or 0 specify that there is no limit on the number of MAC addresses the switch can learn, and 1-256 specifies that the switch will only learn that number of MAC addresses on this port. Packets received from other MAC addresses after this limit is reached will be treated as intrusions. The action the switch takes with intrusion packets is specified by the INTRUSIONACTION parameter, and INTRUSIONACTION=TRAP causes SNMP traps to be generated.

Spanning Tree Protocol (STP)

A Rapier or AR800 port in the STP Learning or Listening state will now only transmit STP BPDUs. It will now not transmit packets that are forwarded from other ports in the switch.

Layer 3 Filtering

The SHOW SWITCH L3FILTER command now correctly displays the IPORT parameter. The IPORT parameter indicates the ingress port number which traffic filtered by this filter entry matches.

Internet Protocol Version 6 (IPv6)

IPv6 is the next generation of the Internet Protocol (IP). It has primarily been developed to solve the problem of the eventual exhaustion of the IPv4 address space, but also offers other enhancements. IPv6 addresses are 16 bytes long, in contrast to IPv4's 4 byte addresses, providing four times the available address space.

Internet Protocol Version 6 was supported on AR720 and AR740 routers by Software Release 2.2.1. Software Release 2.2.2 adds support for IPv6 on the AR300, Rapier and AR800 Series. A special feature licence is required to enable IPv6.

The following IPv6 features are supported by Release 2.2.2:

- Linking together networks which are running IPv6.
- Allowing address autoconfiguration of hosts connected to the router.
- Enabling IPv6 to operate over current, predominantly IPv4, networks.

IPv6 handling of shortest path routing and multiple equal cost routes has been enhanced in Software Release 2.2.2.

The standards supported for IPv6 are defined in RFC 2080, RFC 2463, RFC 2461, RFC 1886, RFC 2465, RFC 2452, RFC 2454, and RFC 2466.

For full information about IPv6 on Rapier switches and AR800 Series switching routers, see the *Internet Protocol Version 6 (IPv6)* chapter of the Software Reference, which is appended to this Release Note. For information about IPv6 on AR300 Series routers, see the AR router Software Reference, which is available on the CD-ROM shipped with your router, or from www.alliedtelesyn.co.nz.

IP Multicasting

PIM Dense Mode

Software Release 2.2.2 now supports PIM Dense Mode in addition to PIM Sparse Mode on Rapier Series Layer 3 Switches (with the Rapier FL3 Upgrade licence AT-RPFL3Upgrade), and AR800 Series Modular Switching Routers.

Protocol Independent Multicast routing protocols rely on the presence of an existing unicast routing protocol to adapt to topology changes, but are independent of the mechanisms of the specific unicast routing protocol. PIM Dense Mode is most suitable for networks where bandwidth is plentiful, and where the members of a multicast group are densely distributed on the network. PIM Sparse Mode is more suitable when the members of the multicast groups are more sparsely distributed over the network, as there is less duplication of data packets over the network.

Unlike PIM Sparse Mode, PIM Dense Mode (PIM-DM) does not use a bootstrap router or Rendezvous Points.

PIM-DM is similar to DVMRP in that it employs the Reverse Path Multicasting (RPM) algorithm. However, there are differences between PIM-DM and DVMRP:

- PIM-DM relies on the presence of an existing unicast routing protocol to provide routing table information to build up information for the multicast forwarding database, but it is independent of the mechanisms of the specific unicast routing protocol. In contrast, DVMRP contains an integrated routing protocol that makes use of its own RIP-like exchanges to compute the required unicast routing information.
- Unlike DVMRP, PIM-DM simply forwards multicast traffic on all downstream interfaces until explicit prune (un-join) messages are received. PIM-DM is willing to accept the overhead of broadcast-and-prune in the interests of simplicity and flexibility, and of eliminating routing protocol dependencies.

PIM-DM assumes that when a source starts sending, all downstream systems want to receive multicast datagrams. Initially, multicast datagrams are flooded to all areas of the network. If some areas of the network do not have group members, dense-mode PIM will prune off the forwarding branch by setting up prune state. The prune state has an associated timer, which on expiration will turn into forward state, allowing data to go down the branch previously in prune state.

The prune state contains source and group address information. When a new member appears in a pruned area, a router can ``graft'' toward the source for the group, turning the pruned branch into forward state. The forwarding branches form a tree rooted at the source leading to all members of the group. This tree is called a source rooted tree.

The broadcast of datagrams followed by pruning of unwanted branches is often referred to as a broadcast-and-prune cycle, typical of dense mode protocols. The broadcast-and-prune mechanism in dense mode PIM uses a technique called reverse path forwarding (RPF), in which a multicast datagram is forwarded if the receiving interface is the one used to forward unicast datagrams to the source of the datagram.

The PIM mode is set for each interface when the interface is added to PIM, using the command:

```
ADD PIM INTERFACE=interface [DLC=1..1024]
    [DRPRIORITY=1..65535] [MODE={DENSE|SPARSE}]
```

or can be changed for an existing PIM interface using the command:

```
SET PIM INTERFACE=interface MODE={DENSE|SPARSE} [DLC=1..1024]
    [DRPRIORITY=1..65535]
```



All interfaces should have the same mode setting. Do not set the mode to DENSE on some interfaces, and SPARSE on others, because this could cause multicast traffic to be forwarded incorrectly.

Information about PIM Dense Mode can be displayed using the command:

```
SHOW PIM [COUNTERS|DEBUG|INTERFACE|NEIGHBOUR|ROUTE|TIMER]
```

If COUNTERS is specified, then all PIM counters are displayed. If DEBUG is specified, then the list of PIM interface debugging options is displayed. If INTERFACE is specified, then the PIM interface list is displayed. If NEIGHBOUR is specified, then the PIM Neighbour Table is displayed. If TIMER is specified, then timer intervals for PIM operations are displayed. If ROUTE is specified, then the internal PIM routing table is displayed. If TIMER is specified, then timer intervals for PIM operations are displayed. If no parameters are specified, then all of the above PIM information is displayed.

Additional debugging information, including PIM Graft messages, can be displayed for PIM Dense Mode using the command:

```
ENABLE PIM DEBUG={ALL|ASSERT|GRAFT|HELLO|JOIN} [ , . . . ]
```

All other commands for configuring PIM Dense Mode are the same as for configuring PIM Sparse Mode. PIM Sparse Mode is described in the *Multicasting* chapter of the Software Reference, which is available on the CD-ROM shipped with your switch or router, or from www.alliedtelesyn.co.nz.

IGMP Traffic and VLAN Ports

On Rapier switches and AR800 switching routers, IGMP LEAVE and JOIN messages are now only sent out the routing port(s), instead of being sent out all ports on the VLAN including the incoming port.

Pinging

Pinging a broadcast address of a multihome interface now results in Ping reply packets with the correct source address.

DVRMP

If two interfaces on the switch or router are configured with different address classes (for example, one interface has a Class B mask and another a Class C mask), DVRMP now routes correctly.

Firewall

Secure Shell Access

Secure Shell is a method of securely logging into the router from a remote location for configuration and management purposes. It is described fully in the *Secure Shell* chapter of the Software Reference, which is available on the CD-ROM shipped with your switch or router, or from www.alliedtelesyn.co.nz.

A default firewall rule exists which allows Secure Shell traffic to the router without the need for additional configuration, in order to provide Secure Shell access to the router from the public side of the firewall. This default allow rule is removed when the Secure Shell server is disabled on the router or switch. The following criteria must be met for Secure Shell traffic to be allowed:

1. The packet must be destined for the router itself. Packets destined for the private network are not given access by this default rule.
2. The packets must not be explicitly denied by another user-configured rule.
3. The Secure Shell server feature must be enabled. Refer to the *Secure Shell* chapter for information on how to enable and disable the Secure Shell server and how to determine its operational status.

This default allow rule for Secure Shell traffic destined for the router does not weaken the firewall, because Secure Shell is a strongly authenticated and encrypted protocol and the rule does not allow Secure Shell traffic through to

the private network. However, Secure Shell traffic can easily be blocked at the public interface while the Secure Shell server is enabled, by adding a rule. For example, if eth1 is the public interface, to block traffic received via the Secure Shell port, TCP port 22, use the command:

```
ADD FIREWALL POLICY=policy-name RULE=rule-id INT=eth1
ACTION=deny PROTOCOL=tcp PORT=22
```

where:

- *policy-name* is a character string, 1 to 15 characters in length. Valid characters are letters (a-z, A-Z), digits (0-9) and the underscore character (“_”).
- *rule-id* is a number in the range 1 to 299. See the *Firewall* chapter of the Software Reference, which is available on the CD-ROM shipped with your switch or router, or from www.alliedtelesyn.co.nz, for detailed information on interpretation of rule numbers.

Multicast Packet Handling

Multicast packets must be handled specially by the firewall because it does not know on which interface the packet will be forwarded. If several policies use the receiving interface, the packet cannot be associated with only one policy. Since Software Release 2.2.1, the firewall uses the following reasoning to decide if a multicast packet should be allowed or denied:

1. If the interface on which the multicast packet is received is a public interface for one or more policies, the packet will be discarded unless at least one policy has an allow rule for it. Therefore, if one policy allows a particular multicast packet, all other policies implicitly allow the packet. IP multicasting will then decide on which interfaces the packet is forwarded.
2. If the interface on which the multicast packet is received is a private interface, and is not a public interface in any other policies, it will be allowed.
3. Network Address Translation (NAT) of any kind cannot be applied to multicast packets.

Multiple VLAN Interfaces

Traffic on Rapier switches and AR800 switching routers is now correctly sent to the firewall when multiple private or public VLAN interfaces are present.

TCP SYN ACK Retransmission

A TCP SYN ACK retransmission from a device on the public side of the firewall, as part of a session initiated by a device on the private side of the firewall, is now accepted if it arrives within the 6 seconds after the first SYN ACK was received. Previously, such retransmissions were rejected as out-of-sequence packets.

Frame Relay

An optional new parameter, DEFATENCAP, has been added to the CREATE FRAMERELAY and SET FRAMERELAY commands. This parameter enables users to specify the encapsulation to use for Appletalk over Frame Relay.

The CREATE FRAMERELAY command syntax is now:

```
CREATE FRAMERELAY=fr-interface OVER=physical-interface
  [DEFATENCAP={APPLEOUI|IETFOUI}] [other-options...]
```

The SET FRAMERELAY command syntax is now:

```
SET FRAMERELAY=fr-interface [DEFATENCAP={APPLEOUI|IETFOUI}]
  [other-options...]
```

The DEFATENCAP parameter specifies the default AppleTalk packet encapsulation (OUI SNAP encapsulation) that is applied to the packet before it is sent over the Frame Relay interface. APPLEOUI specifies SNAP encapsulation with Apple OUI and IETFOUI specifies SNAP encapsulation with IETF OUI. The default is APPLEOUI.

Asynchronous Call Control (ACC)

The invalid command SHOW ACC CALL=DIALIN CALL (in which the parameter CALL has been entered twice) now results in an accurate error message.

Hypertext Transfer Protocol (HTTP)

The HTTP server will now accept usernames and passwords up to 32 characters in length.

IPsec

ISAKMP Signature authentication

When RSA signatures are used, the SKEYID value is now calculated correctly.

ISAKMP Aggressive mode

If the last message in Aggressive mode exchanges is encrypted, it can now be decrypted successfully.

ISAKMP XAUTH.

Interoperability with other vendors' implementations of XAUTH has been improved.

Open Shortest Path First (OSPF)

The PRIORITY parameter now functions correctly with the ADD/SET OSPF INTERFACE command for Rapier switches and AR800 switching routers.

The PRIORITY parameter is used on multi-access networks to set the OSPF interface priority. When two routing switches attached to a network attempt to become the designated router, the one with the highest OSPF interface priority takes precedence. If the priorities are the same then the routing switch with the highest router identification number takes precedence. A routing switch with a OSPF interface priority of zero is ineligible to become the designated router. Priority can only be configured for routing switches attached to multi-access networks. The default is 1.

Point-to-Point Protocol (PPP)

PPPoE

Rapier switches and AR800 switching routers now process the VLAN parameter of the ADD PPP ACSERVER correctly when this command appears in a router-generated configuration script. This parameter specifies a VLAN on which the PPPoE Access Concentrator service is offered.

When reducing bandwidth on demand, the PPP interface now sends a TERM REQ to close the link, instead of deactivating the lower layer interface directly. This increases the number of PPP implementations with which Allied Telesyn products will interoperate.

If a dial on demand PPP interface is in the REQ SENT state and receives a DOWN event from the lower layer interface, it is now able to bring up the dial on demand PPP interface when the lower layer interface comes back up.

Telephony (PBX)

The AR300(S) and AR310(S) (AR300 series routers with voice ports) now correctly handle either one or two BRI DLCs in both the incoming and outgoing direction.

Public Key Infrastructure (PKI)

Subject Alternative Names are now processed correctly in manual certificate requests.

Some types of certificates, which previously could not be decoded, are now correctly parsed.

Empty CRLs are now correctly parsed.

Interoperability issues with several CAs have now been corrected.

Errata: 2.2.1 Software Reference

The syntax and description of the SET IP ARP TIMEOUT command are incorrect in the Software Reference distributed with Software Release 2.2.1 and more recent software versions.

The correct syntax is:

```
SET IP ARP TIMEOUT=multiplier
```

The correct description is:

This command sets a multiplier value (any integer) used to change the ARP timeout by set increments. By default, the multiplier is 4, which causes ARP timeouts to vary between 1024 and 2048 seconds, depending on when ARP replies are received. Specifying a value of 1 decreases the default timeout to 256-512 seconds.

Availability

Software Release 2.2.2 is available immediately as a FLASH release for upgrading existing routers, switches and switching routers. The release file can be downloaded directly from the Software Updates area of the Allied Telesyn web site at www.alliedtelesyn.co.nz/support/updates/patches.html, or from the support site for your switch: www.alliedtelesyn.co.nz/support/rapier/ or www.alliedtelesyn.co.nz/support/ar800/.

Software releases must be licenced and require a password to activate. To obtain a licence and password, contact your authorised Allied Telesyn distributor or reseller. You will need to specify the software release, switch or router model, and the serial number for each switch or router running the release.

Installation

There are no issues upgrading from Software Releases 2.2.x to Software Release 2.2.2.

Software Release 2.2.1 added a new parameter, LOGIN, to the ADD USER and SET USER commands:

```
ADD USER=login-name PASSWORD=password LOGIN={TRUE|FALSE|ON|
OFF|YES|NO} [other-options...]

SET USER=login-name [LOGIN={TRUE|FALSE|ON|OFF|YES|NO}]
[other-options...]

SET USER [LOGIN={TRUE|FALSE|ON|OFF|YES|NO}] [other-
options...]
```

The LOGIN parameter is used to specify whether or not users with a privilege of “user” will be able to login to the command line interface. Usernames with LOGIN set to TRUE can be used both for PAP and CHAP authentication, and to login and access the command line. Usernames with LOGIN set to FALSE can only be used for PAP and CHAP authentication.

After upgrading from 2.0.x or 2.1.x to 2.2.x, the LOGIN parameter is required. If it is not set in a boot script, its value defaults to TRUE, allowing users with a privilege level of “user” to access the CLI. The manager then has the option of denying access to the CLI for some or all users.

If you are downloading the release to a switch model with a Graphical User Interface, make sure you download the correct resource file as well as the Software Release file.

Software Release 2.2.2

Internet Protocol Version 6 (IPv6)

For Rapier Switches and AR800 Series Modular Switching Routers

Introduction	3
Overview of IPv6	3
The 6bone	4
IPv6 Addresses and Prefixes	4
IPv6 Headers	5
The Internet Control Message Protocol (ICMPv6)	8
IPv6 Routing	10
IPv6 Filtering	11
Integration of IPv4 and IPv6	11
Support for IPv6	11
Enabling IPv6	12
IPv6 Interfaces and Addresses	12
Extension Header Processing	14
Routing Table Processing and RIPv6	14
Neighbour Discovery	15
IPv6 Filtering	15
IPv6 Fragmentation	16
Telnet v6	16
Ping and Trace Route	17
Tunnelling	17
Configuration Examples	18
Basic Routing	18
Dynamic Routing with RIPv6	19
Tunnelling over an IPv4 network	21
IPv6 Filters	23
Commands	27
ADD IPV6 FILTER	27
ADD IPV6 HOST	32
ADD IPV6 INTERFACE	33
ADD IPV6 RIP	34
ADD IPV6 ROUTE	35
ADD IPV6 TUNNEL	36
CREATE IPV6 INTERFACE	37
DELETE IPV6 FILTER	37
DELETE IPV6 HOST	38
DELETE IPV6 INTERFACE	38
DELETE IPV6 RIP	39
DELETE IPV6 ROUTE	39
DELETE IPV6 TUNNEL	40
DESTROY IPV6 INTERFACE	40
DISABLE IPV6	41

DISABLE IPV6 ADVERTISE	41
DISABLE IPV6 DEBUG	42
DISABLE IPV6 RIP	42
ENABLE IPV6	42
ENABLE IPV6 ADVERTISE	43
ENABLE IPV6 DEBUG	43
ENABLE IPV6 RIP	43
SET IPV6 FILTER	44
SET IPV6 INTERFACE	47
SHOW IPV6	48
SHOW IPV6 COUNTER	50
SHOW IPV6 FILTER	53
SHOW IPV6 HOST	55
SHOW IPV6 INTERFACE	56
SHOW IPV6 MULTICAST	57
SHOW IPV6 ND	58
SHOW IPV6 RIP	59
SHOW IPV6 ROUTE	61
SHOW IPV6 TIMER	62
SHOW IPV6 TUNNEL	62

Introduction

This chapter describes the main features of IPv6, the router's implementation of IPv6 and how to configure and operate IPv6 on the router.

In summary, the router supports the following IPv6 features:

- Linking together networks which are running IPv6.
- Allowing address autoconfiguration of hosts connected to the router.
- Enabling IPv6 to operate over current, predominantly IPv4, networks.



IPv6 requires a special feature licence, which can be obtained from any authorised Allied Telesyn distributor or reseller.

Overview of IPv6

IPv6 is the next generation of the Internet Protocol (IP). It has primarily been developed to solve the problem of the eventual exhaustion of the IPv4 address space, but also offers other enhancements. IPv6 addresses are 16 bytes long, in contrast to IPv4's 4 byte addresses. Other features of IPv6 include:

- Address structure improvements:
 - globally unique addresses with more levels of addressing hierarchy, to reduce the size of routing tables
 - autoconfiguration of addresses by hosts
 - improved scalability of multicast routing, by adding a "scope" field to multicast addresses
 - a new type of address, the "anycast address", which is used to send a packet to any one of a group of devices.
- Removal of the need for packet fragmentation en-route, by dynamic determination of the largest packet size that is supported by every link in the path. A link's MTU (Maximum Transmission Unit) must be at least 1280 bytes, compared with 576 bytes for IPv4.
- Traffic Class, which allows a packet to be labelled with an appropriate priority. If the network becomes congested, the router will drop packets of lowest priority.
- Flow labels, which indicate to intermediate routers that packets are part of a flow, and that this flow requires a particular type of service. This feature enables, for example, real-time processing of data streams. It also increases routing speed because the forwarding router need only check the flow label, not the rest of the header. The handling indicated by the flow label can be done by the IPv6 Hop-by-Hop header, or by a separate protocol such as RSVP.
- Mandatory authentication and data integrity protocols, through IPsec. IPsec is optional in IPv4.

An IPv6 network can contain three types of nodes, where *node* is a general term that refers to any IPv6-aware device. A *host* is a device on the network that is not a router. For example, a host may be a printer or a computer. A router may

also act as a host. A *router* is a device on the network that directs the flow of IPv6 packets. For example, a router may be a router or a Layer 3 switch. A *destination* is a host to which packets are specifically sent.

The 6bone

The 6bone is an experimental virtual network of nodes that support IPv6 packets, tunnelled together through the existing IPv4 Internet. Most of the nodes are workstations or similar machines, with IPv6-capable operating systems. The theory of tunnelling IPv6 packets over an IPv4 network is outlined in “*Integration of IPv4 and IPv6*” on page 11.

The 6bone is part of the transition to IPv6. Its purpose is to provide an environment in which IPv6 can be tested and procedures for IPv6 can be developed. When IPv6 is sufficiently developed and being used widely, the 6bone will probably disappear.

IPv6 Addresses and Prefixes

IPv6 addresses are made up of eight 16-bit numbers separated by colons (:). An example of a valid address is FE80:0000:0000:0260:0000:97FF:64AA. In the interests of brevity, addresses can be abbreviated in two ways:

- Leading zeros can be omitted, so this address can be written as FE80:0:0:260:0:97FF:64AA.
- Consecutive zeros can be replaced with a double colon, so this address can be written as FE80::260:0:97FF:64AA. Note that a double colon can replace any number of consecutive zeros, but an address can only contain one double colon.

Prefixes can be specified at the beginning of an address, and provide the equivalent functionality to a subnet mask in IPv4, allowing a subnet to be addressed, rather than a single node. If a prefix is specified, the IPv6 address is followed by a slash to indicate how many bits represent the prefix. For example, 3FFE::<16> indicates that the first 16 bits of the address 3ffe:0:0:0:0:0:0:0 represent the prefix.

Like IPv4 addresses, IPv6 addresses are attached to interfaces.

Unicast Addresses

A unicast address is attached to a single interface, and is used to deliver packets only to that interface.

A number of special addresses have been defined:

- IPv4-compatible and IPv4-mapped addresses. IPv4-compatible addresses are used to tunnel IPv6 packets across an IPv4 network. IPv4-mapped addresses are used by an IPv6 host to communicate with an IPv4 host. The IPv6 host addresses the packet to the mapped address.
- Link-local addresses, which can only be used on the local network that the interface is attached to, and site-local addresses, which can only be used on the networks that are not attached to the Internet.
- The Loopback address, consisting of ::1, which is the equivalent of the IPv4 loopback address, and allows a host to send packets to itself.

- The Unspecified address, consisting of ::, which is the equivalent of the IPv4 unspecified address, and is used as a source address by hosts during the autoconfiguration process.

Multicast Addresses

IPv6 multicast addresses replace IPv4 broadcast addresses. A multicast address identifies a group of interfaces, and packets are sent to all interfaces in that group.

Among the special addresses which have been defined are addresses which allow multicasting to:

- All interfaces on a particular host (FF01::1)
- All nodes on a local network (FF01::2)
- All routers on the local link (FF02::2)
- All routers on the local site (FF05::2).

Anycast Addresses

An *anycast* address is a unicast address that is attached to more than one interface. If a packet is sent to an anycast address it will be delivered to the nearest interface with that address, with the definition of “nearest” depending on the protocol used for routing. If the protocol is RIPv6, the nearest interface will be the one which is the shortest number of hops away.

Anycast addresses can only be assigned to routers, and packets cannot originate from an anycast address. A router must be configured to know if it is using an anycast address, because the address format cannot be distinguished from that of a unicast address.

Only one anycast address has been predefined, the subnet-router address. The subnet-router address is used to send messages to the nearest router on a subnet, and consists of the subnet’s prefix followed by zeros.

IPv6 Headers

The basic unit of data sent through an internet is called a *packet* in IPv6. A packet consists of a *header* followed by the *data* (Figure 1, Table 1). The header contains the information necessary to move the packet across the internet. It must be able to cope with missing and duplicated packets as well as possible fragmentation (and reassembly) of the original packet.

The IPv6 header is similar to the shorter IPv4 header, which is described in Figure 1 and Table 1 of the *Internet Protocol (IP)* chapter in the Software Reference. Although the IPv6 header is only twice as long as the IPv4 header (40 bytes instead of 20 bytes), it contains four times the address space size (128 bits instead of 32 bits).

The Basic IPv6 Header

The IPv6 header no longer contains the header length, identification, flags, fragment offset and header checksum fields. Some of these options have been placed in extension headers. The Time To Live field has been replaced with a hop limit, and the IPv4 Type of Service field is now replaced with a Traffic Class field.

Figure 1: The IPv6 packet.

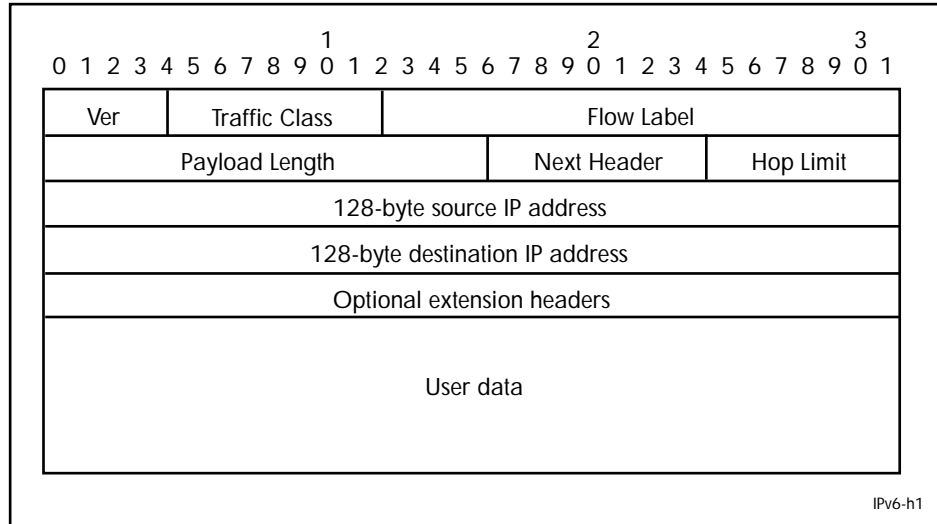


Table 1: A general description of the fields in an IPv6 packet header.

Field	Function
Ver	The version of the IP protocol that created the packet. For IPv6, this field has a value of 6.
Traffic Class	An 8-bit value which indicates the priority that a packet should be given.
Flow Label	A 20-bit value which indicates the data flow that this packet is part of. This flow may be handled in a particular way.
Payload Length	The length of the user data portion of the packet. If the data payload is larger than 64 kB, the length is given in the optional "Jumbo Payload" header and the Payload Length header is given a value of zero.
Next Header	A number which indicates the type of header that immediately follows the basic IP header. This header type may be an optional IPv6 extension header, a relevant IPv4 option header, or another protocol, such as TCP or ICMPv6. The IPv6 extension header values are: 0 (Hop-by-Hop Options Header) 43 (IPv6 Routing Header) 44 (IPv6 Fragment Header) 50 (Encapsulating Security Payload) 51 (IPv6 Authentication Header) 59 (No Next Header) 60 (Destination Options Header).
Hop Limit	A field which is the equivalent of the IPv4 Time To Live field, measured in hops.
Source IP address	The 128-bit IPv6 address of the sender.

Table 1: A general description of the fields in an IPv6 packet header. (Continued)

Field	Function
Destination IP address	The 128-bit IPv6 address of the recipient.
Optional extension headers	The optional headers which give less-frequently used information.

The IPv4 header fields that have been changed in IPv6 are:

■ **Type of Service**

The type of service that a connection should receive is indicated in IPv6 by the Flow Label field in the IPv6 header.

■ **Fragmentation information (the Identification field, the Flags field and the Fragment Offset field)**

In most cases fragmentation will not occur in IPv6, and if it does, packets will only be fragmented at their source, not en route. Therefore, the fragmentation information is now contained in an extension header, to reduce the size of the basic IPv6 header.

■ **Header Checksum**

This option has not been provided in IPv6. This is partly because transport protocols implement checksums and partly because of the availability of the IPsec authentication header (AH) in IPv6.

■ **Options**

Extension headers now handle all the optional values associated with IPv6 packets. The biggest advantage of this scheme is that the size of the basic IP header is a constant.

Extension Headers

In IPv6, many of the less commonly used fields in the IPv4 header (or their equivalents) have become extension headers, which are placed after the basic IPv6 header. The length of each header must be a multiple of 8 bytes.

The first extension header is identified by the Next Header field in the basic IPv6 header. Any extension headers after this first one are identified by an 8-bit "Next Header" value at the beginning of the preceding extension header.

IPv6 nodes that originate packets are required to place extension headers in a specific order:

1. The basic IPv6 header, which must come immediately before the extension headers.
2. The Hop-by-Hop header, which specifies options that must be examined by every node in the routing path.
3. A Destination Options header, to specify options to be processed by the first destination or final destination. The destination options header is the only extension header that may be present more than once in the IPv6 packet.
4. The Routing header, which allows a static path to be specified for the packet, if the dynamically-determined path is undesirable.
5. The Fragment header, which indicates that the source node has fragmented the packet, and contains information about the fragmentation.

6. The Authentication header, which verifies the integrity of the packet and its headers. The AH is an IPsec feature, and is described in “*Overview of IPsec*” in the *IP Security (IPsec)* chapter of the Software Reference.
7. The Encapsulating Security Payload header, which encrypts a packet and verifies the integrity of its contents. Like AH, ESP is an IPsec feature, and is also described in “*Overview of IPsec*” in the *IP Security (IPsec)* chapter of the Software Reference.
8. The Upper Layer Protocol header, which indicates which protocol a higher layer (such as the transport layer) is to process the packet with (for example, TCP).

The Internet Control Message Protocol (ICMPv6)

The Internet Control Message Protocol, ICMPv6, provides a mechanism for error reporting and route discovery and diagnostics. It also conveys information about multicast group membership, a function which is carried out by the Internet Group Management Protocol (IGMP) in IPv4, and performs address resolution, which the Address Resolution Protocol (ARP) performs in IPv4.

Significant aspects of ICMPv6 include neighbour discovery, which allows one device in a network to find out about other nearby devices, and stateless address autoconfiguration, which allows a device to dynamically determine its own IPv6 address.

ICMPv6 is also used to support the Ping v6 (*Packet Internet Groper*) and Trace route v6 functions, which are used to verify the connections between networks and network devices. Ping is used to test the connectivity between two network devices to determine whether or not each network device can “see” the other device. Trace route is used to discover the route used to pass packets between two systems running the IP protocol.

Both of these functions operate almost identically in IPv4 and IPv6. For more information, see “*Ping and Trace Route*” in the *Internet Protocol (IP)* chapter of the Software Reference.

Neighbour Discovery

Neighbour discovery is an ICMPv6 function that allows a router or host to identify other devices on its links. This information is then used in address autoconfiguration, to redirect a node to use a more appropriate router if necessary, and to maintain reachability information with its neighbours.

The IPv6 Neighbour Discovery protocol is similar to a combination of the IPv4 protocols ARP, ICMP Router Discovery and ICMP Redirect.

There are 5 packet types involved with neighbour discovery:

- *Router solicitation*, in which a host sends out a request for routers to generate advertisements.
- *Router advertisement*, which allows routers to advertise their presence and other network parameters. A router will send an advertisement packet in response to a solicitation packet from a host.
- *Neighbour solicitation*, in which a node sends a packet to determine the link layer address of a neighbour or to verify that a neighbour is still active.

- *Neighbour advertisement*, which is a response to a neighbour solicitation packet. These packets are also used to notify neighbours of link layer address changes.
- *Redirect*, which is used to inform hosts of a better first hop.

These packet types are used to provide the following services:

- *Address resolution*

This is a method for carrying out address autoconfiguration, and is achieved using the Neighbour Solicitation Message and the Neighbour Advertisement Message.

- *Router and prefix discovery*

On connection to a link, a node needs to know the address of a router that the node can use to reach the rest of the world. The node also needs to know the prefix (or prefixes) that define the range of IP addresses on its link that it can reach without going through a router.

Routers use ICMP to convey this information to hosts, by means of router advertisements. The message may have an option attached (the *source link address* option), which enables the receiving node to respond directly to the router, without performing a neighbour solicitation.

- *Immediate information*

The configuration of a router includes a defined frequency at which unsolicited advertisements are sent. If a node wants to obtain information about the nearest router immediately, rather than waiting for the next unsolicited advertisement, the node can send a router solicitation message.

Each router that receives the solicitation message sends a router advertisement specifically to the node that sent the solicitation.

- *Redirection*

If a node is aware of more than one router which it can use to connect to wider networks, the router which it sends packets to by default will not always represent the most desirable route. ICMPv6 uses the redirect packet to communicate a more effective path to the node.

- *Neighbour Unreachability Detection (NUD)*

A node may issue solicitation requests to determine whether a path is still viable, or may listen in on acknowledgement packets of higher-layer protocols, such as TCP. If the node determines that a path is no longer viable, it attempts to establish a new link to the neighbour, or to re-establish the previous link. NUD can be used between any two devices in the network, independent of whether the devices are acting as hosts or routers.

Stateless address autoconfiguration

Stateless address autoconfiguration allows an IPv6-aware device to be plugged into a network without manual configuration with an IP address. This plug and play functionality results in networks which are much easier to set up, and simplifies the process of shifting to use a new Internet Service Provider (ISP).

Stateless address autoconfiguration is achieved in a series of steps. The first three steps autoconfigure a link-local address, and the last three a global address. The first three steps are performed by routers, and all six steps are performed by hosts.

On the router or host:

1. During system start-up, the node begins autoconfiguration by generating a link-local address for the interface. A link-local address is formed by adding the interface ID to the link-local prefix FE80::/10.
2. The node then transmits a neighbour solicitation message to this address. If the address is already in use, the node that the address belongs to will reply with a neighbour advertisement message. The autoconfiguration process will stop and manual configuration of the node is then required.
3. If no neighbour advertisement is received, the node will conclude that the address is currently available and will assign it to the chosen interface.

On the host:

4. The node then sends one or more router solicitations to detect if any routers are present. Any routers present will respond with a router advertisement.
If no router advertisement is received, the node then attempts to use DHCP to obtain an address and other configuration information. If no DHCP server responds, the node continues using the link-level address.
If a router advertisement is received, then this message will inform the node how to proceed with the auto configuration process.
5. The prefix from the router advertisement, if received, is then added to the link-level address to form the global unicast IP address.
6. This address is then assigned to the network interface.

If routers are present, the node will continue to receive router advertisements. The node will update its configuration if there are any changes in the router advertisements.

IPv6 Routing

Routing in IPv6 is almost identical to IPv4 routing under CIDR, except that the addresses are 128-bit IPv6 addresses instead of 32-bit IPv4 addresses. More information about routing can be found in “*Routing*” in the *Internet Protocol (IP)* chapter of the Software Reference.

Routing Information Protocol, RIPv6

RIP is a simple distance vector protocol which defines networks based on how many hops they are from the router. Once a network is more than 15 hops away (one hop is one link) it is not included in the routing table.

RIPv6, also referred to as RIPng (for “next generation”) is similar to RIPv2, which is described in “*RIP*” in the *Internet Protocol (IP)* chapter of the Software Reference. Extensions to RIPv2 to support IPv6 are:

- The address field of a routing entry is expanded to 128 bits to allow IPv6 prefixes
- The 32-bit RIPv2 subnet mask field is replaced by an 8-bit prefix length field
- Authentication is removed in RIPv6
- The size of a routing packet is no longer arbitrarily limited
- IPv6 is able to specify the next hop, instead of simply allowing the recipient of the update to set the next hop to the sender of the update.

In RIPv6, each router uses a routing table to keep track of every destination that is reachable throughout the system. Each entry in the routing table contains:

- The IPv6 prefix of the destination
- A metric, which represents the total cost of getting a packet from the router to that destination
- The IPv6 address of the next router along the path to the destination
- A flag to indicate that information about the route has changed recently
- Various timers associated with the route.

IPv6 Filtering

With the increase in connections to the Internet, and the interconnection of networks from different organisations, filtering of data packets is an important mechanism in ensuring that only legitimate connections are allowed. Security can never be perfect while connections to other networks exist, but filters allow network managers to manage the permissible free access, while restricting users who do not have permission.

Like IPv4 filtering, IPv6 filtering is based upon filters. More information can be found in “*Policy-Based Routing*” in the *Internet Protocol (IP)* chapter of the Software Reference.

Integration of IPv4 and IPv6

IPv6 has been designed in such a way that a smooth transition from IPv4 is possible. The most effective way to ensure this is to use a *dual IP stack*. A node which has been configured as a dual stack system has both a 128-bit IPv6 address and a 32-bit IPv4 address, and can communicate with nodes running only IPv4 and nodes running only IPv6.

Another aspect of the transition period is the tunnelling of IPv6 packets across the IPv4 network. IPv6 packets are tunnelled simply by encapsulating the IPv6 packet within an IPv4 datagram, and identifying that this datagram is an encapsulated IPv6 packet by giving the datagram a protocol value of 41.

Support for IPv6

This section describes the router’s support for IPv6, and how to configure IPv6 on the router. Fundamental IPv6 features on the router are:

- IPv6 interfaces and addresses
- Extension header processing
- Routing table processing
- RIPv6 (RIPng)
- Neighbour discovery
- Stateless Address Autoconfiguration
- IPv6 filtering

- IPv6 fragmentation
- Multicasting support

The router also supports the following upper layer protocols:

- UDP, which transports RIPv6 packets
- TCP, which transports Telnet requests
- ICMPv6, which is used for Stateless Address Autoconfiguration, neighbour discovery, Ping and Trace Route requests.

Integration of IPv6 with IPv4 is provided by:

- Tunnelling capabilities.

Many of these features are performed automatically by the router, and most commands operate in a similar manner to their IPv4 equivalents.

Enabling IPv6

The router's implementation of IPv6 is disabled by default. To enable IPv6, use the command:

```
ENABLE IPV6
```

To disable IPv6, use the command:

```
DISABLE IPV6
```

Any IPv6 configuration which the router has performed dynamically will be preserved between disabling and re-enabling IPv6. For example, any addresses which have been configured will still be present.

To display information about IPv6 settings, use the command:

```
SHOW IPV6
```

To display IPv6 counters, use the command:

```
SHOW IPV6 COUNTER
```

To display the IPv6 timers and how long each timer has left to run, use the command:

```
SHOW IPV6 TIMER
```

Because the router implements IPv6 as a dual stack, implementing IPv6 does not affect IPv4 functionality.

IPv6 Interfaces and Addresses

The router supports the adding of IPv6 addresses directly to Virtual Local Area Network (VLAN) and Point-to-Point Protocol (PPP) interfaces and indirectly to virtual interfaces, when the tunnel is created.

IPv6 network addresses on interfaces on the router contain a prefix, the length of which can be indicated in one of two ways. The IPv6 address can be followed either by a slash (/) and the length of the prefix as a number, or the prefix length can be given as a separate PLEN parameter. Therefore, the syntax for an IPv6 address is:

```
IPADDRESS=ipv6address{/prefixlength|PLEN=prefixlength}
```

To create an IPv6 logical interface, and associate it with a VLAN interface, use the command:

```
CREATE IPV6 INTERFACE=interface
```

As part of the creation process, the router will perform stateless address autoconfiguration to assign it an IPv6 address. In stateless address autoconfiguration, the router will automatically determine the IPv6 addresses of its interfaces, by adding the interface's MAC address after the reserved IPv6 prefix FE80::. These addresses are only link-local addresses, which are sufficient for communication among devices on the same link.

If the router is used as a host, it will still be able to use stateless address autoconfiguration to generate a link-local address. However, it will be unable to receive router advertisements, and therefore unable to autoconfigure a global address. If it is necessary to give the router a global IPv6 address, it must be added manually.

To add an IPv6 address manually to the VLAN interface, or to create a VLAN or PPP interface and manually add the IPv6 address to it at the same time, use the command:

```
ADD IPV6 INTERFACE=interface
    IPADDRESS=ipv6address{/prefixlength|PLEN=prefixlength}
    [FILTER=filtNum] [PREFERRED=seconds|INFINITE]
    [PRIORITYFILTER=priNum] [VALID=seconds|INFINITE]
```

To change the address or other parameters of the interface, use the command:

```
SET IPV6 INTERFACE=interface
    IPADDRESS=ipv6address{/prefixlength|PLEN=prefixlength}
    [FILTER=filtNum] [PREFERRED=seconds|INFINITE]
    [PRIORITYFILTER=priNum] [VALID=seconds|INFINITE]
```

To destroy an IPv6 interface, use the command:

```
DESTROY IPV6 INTERFACE=interface
```

To delete an address from an interface, use the command:

```
DELETE IPV6 INTERFACE=interface IPADDRESS=ipv6address
```

To display information about the configured interfaces, use the command:

```
SHOW IPV6 INTERFACE[=interface]
```

To display information about IPv6 multicast addresses, use the command:

```
SHOW IPV6 MULTICAST
```

To associate an IPv6 address with the name of a host, use the command:

```
ADD IPV6 HOST=name IPADDRESS=ipv6address
```

This functionality is similar to the ADD IP HOST command, described in the *Internet Protocol (IP)* chapter of the Software Reference.

To disassociate the IPv6 address and the host name, use the command:

```
DELETE IPV6 HOST=name
```

To display information about the host name table, use the command:

```
SHOW IPV6 HOST
```

Extension Header Processing

All routers in the path to the final destination will process the routing header and the hop by hop header, to route the packet to the specified path. The final node will process the fragment header. If the router is the source or final destination of the packet, it will process these extension headers as required.

Routing Table Processing and RIPv6

The router maintains and processes a routing table for IPv6 addresses, in a similar manner to the IPv4 routing table described in “*Routing*” in the *Internet Protocol (IP)* chapter of the Software Reference.

The Router Information Protocol (RIPv6 or RIPng, as described in RFC 2080, “*RIPng for IPv6*”) has been implemented, to allow routers to share information from their routing tables with routers that connect other networks. RIPv6 passes routing table information from neighbour to neighbour along a line of routers.

RIPv6 packets are transported over UDP. When IPv6 is enabled, the router can route UDP packets through an IPv6 network or tunnel.

RIPv6 routing is disabled by default. To enable it, use the command:

```
ENABLE IPv6 RIP
```

To enable receiving or sending of RIP packets on an interface, use the command:

```
ADD IPV6 RIP INTERFACE=interface
```

To stop an interface from sending or receiving RIP packets, use the command:

```
DELETE IPV6 RIP INTERFACE=interface
```

To disable RIP, use the command:

```
DISABLE IPv6 RIP
```

To display information about RIP counters or timers, use the command:

```
SHOW IPV6 RIP [COUNTER|TIMER]
```

Under some conditions, the route that is dynamically determined by RIPv6 may not be the most desirable one. For example, certain packets may need to be sent over a more secure route. The desired route can be added statically to an interface, using the command:

```
ADD IPV6 ROUTE=ipv6address{/prefixlength|PLEN=prefixlength}  
INTERFACE=interface NEXTHOP=ipv6address [METRIC=1..16]  
[PREFERENCE=0..65535]
```

The ROUTE parameter identifies final destination network to which the packets will be sent.

This command can also be used to add a default route to which packets will be sent if no other route is found, for example the gateway between the LAN and the wider network. To add a default route, use the command:

```
ADD IPV6 ROUTE::/0 INTERFACE=interface  
NEXTHOP=ipv6address-to-send-packets-to  
[METRIC=1..16] [PREFERENCE=0..65535]
```

To delete a route from an interface, use the command:

```
DELETE IPV6 ROUTE=ipv6address INTERFACE=interface
NEXTHOP=ipv6address
```

To display information about the IPv6 routes on the router, use the command:

```
SHOW IPV6 ROUTE
```

Neighbour Discovery

The router will issue router advertisement messages in response to a router solicitation message from a host, to enable the host to determine the router's identity and availability. The router will also send neighbour solicitation messages to neighbouring nodes, and respond to neighbour solicitation messages with a neighbour advertisement message. The address resolution mechanism and queue structure is similar to that used in ARP in IPv4. More information about neighbour discovery can be found in RFC 2461, "*Neighbour Discovery for IPv6*".

Router advertisement is enabled by default. To disable it, use the command:

```
DISABLE IPV6 ADVERTISEERTISE
```

To re-enable router advertisements, use the command:

```
ENABLE IPV6 ADVERTISE
```

Neighbour discovery and solicitation are part of the IPv6 protocol and cannot be disabled.

To display information about the neighbours determined by neighbour discovery, use the command:

```
SHOW IPV6 ND
```

IPv6 Filtering

IPv6 packets can be filtered on arrival at the router, with a traffic filter, or on transmission from the router, with a priority filter. Traffic filters can be used to determine whether an incoming packet is accepted or rejected. Priority filters apply a particular priority to the packets. Priorities range from 0 to 7, with 0 having the highest priority.

To add an IPv6 filter, use the command:

```
ADD IPV6 FILTER=filter-number
SOURCE=ipv6address{/prefixlength|SLEN=prefixlength}
[ACTION={INCLUDE|EXCLUDE}|PRIORITY=P0..P7]
[DESTINATION=ipv6address{/prefixlength|
DPLEN=prefixlength}] [DPORT={port-name|port-id|ANY}]
[ENTRY=entry-number] [ICMPCODE={icmp-code-name|icmp-code-
id|ANY}] [ICMPTYPE={icmp-type-name|icmp-type-id|ANY}]
[LOG={4..1950|DUMP|HEADER|NONE}] [OPTIONS={YES|NO}]
[PROTOCOL={protocol|ANY|EGP|ICMP|OSPF|TCP|UDP}]
[SESSION={ANY|ESTABLISHED|START}] [SIZE={size|ANY}]
[SPORT={port-name|port-id|ANY}]
```

The ADD IPV6 FILTER command can be used to add another entry to an existing filter, by identifying the filter by its filter-number and using the

ENTRY parameter to indicate the desired entry number for the new entry. Entries with the lowest entry numbers will be higher in the filter, and therefore will be applied to traffic before higher-numbered entries.

To modify an existing filter entry, use the command:

```
SET IPV6 FILTER=filter-number ENTRY=entry-number
[ACTION={INCLUDE|EXCLUDE}|PRIORITY=P0..P7]
[DESTINATION=ipv6address{/prefixlength|
DPLEN=prefixlength}] [DPORT={port-name|port-id|ANY}]
[ICMPCODE={icmp-code-name|icmp-code-id|ANY}]
[ICMPATYPE={icmp-type-name|icmp-type-id|ANY}]
[LOG={4..1950|DUMP|HEADER|NONE}] [OPTIONS={YES|NO}]
[PROTOCOL={protocol|ANY|EGP|ICMP|OSPF|TCP|UDP}]
[SESSION={ANY|ESTABLISHED|START}][SIZE={size|ANY}]
[SOURCE=ipv6address{/prefixlength|SPLEN=prefixlength}]
[SPORT={port-name|port-id|ANY}]
```

To delete a filter or one of its entries, use the command:

```
DELETE IPV6 FILTER=filter-number ENTRY={entry-number|ALL}
```

To display information about existing filters, sorted by filter and entry numbers, use the command:

```
SHOW IPV6 FILTER[=filter-number]
```

IPv6 Fragmentation

Routers along a path send an “ICMP packet too big” reply when they receive a packet that is larger than the MTU for the link. It is then up to the host to fragment the packets.

As a host, the router will perform fragmentation as needed. If the packet size is greater than the MTU the packet is fragmented into packets that are sized in multiples of 8 bytes. Each of the new packets carries a fragmentation header. A packet with a fragmentation header is identified with the value 44 in the previous header’s Next Header field.

Telnet v6

The TELNET command allows remote access to a device’s command-line interface. It operates similarly in IPv4 and IPv6. For more information, see “Telnet” in the *Terminal Server* chapter of the Software Reference.

To Telnet to an IPv6 interface with the address FE80::260:0:97FF:64AA, use the command:

```
TELNET FE80::260:0:97FF:64AA
```

Instead of entering the IPv6 address of the host’s interface, an easy-to-remember name can be associated with the host, using the command:

```
ADD IPV6 HOST=name IPADDRESS=ipv6address
```

Telnet messages are transported over TCP. When IPv6 is enabled, the router can route TCP packets through an IPv6 network or tunnel.

Ping and Trace Route

The router supports an extended PING command, described in the *Internet Protocol (IP)* chapter of the Software Reference, which allows the user to attempt to contact an IPv6 interface and record whether packets are received and the response time if they are. Default values, including a default address, can be entered with the SET PING command, described in the *Internet Protocol (IP)* chapter of the Software Reference.

The TRACE command, described in the *Internet Protocol (IP)* chapter of the Software Reference, records the path to an IPv6 node.

To view the route to a node, use the command:

```
TRACE ipv6address
```

Tunnelling

The router supports the tunnelling of IPv6 packets over an IPv4 tunnel. This allows two networks running IPv6 to be linked by an IPv4 network. “*Tunnelling over an IPv4 network*” on page 21 is an example of this configuration.

To link two IPv6 networks through an IPv4 tunnel, first create the tunnel on each router, using the command:

```
ADD IPV6 TUNNEL IPADDRESS=ipv6address LOCAL=ipv4address  
TARGET=ipv4address
```

Then add a route on both IPv6 routers participating in the tunnel that tells the router to use the tunnel.

To delete a tunnel from an interface, use the command:

```
DELETE IPV6 TUNNEL=ipv6address
```

The IPv6 address in the TUNNEL parameter of the DELETE command is the address that was added to the tunnel with the IPADDRESS parameter of the ADD IPV6 TUNNEL command.

To display information about the tunnels that are configured on the router, use the command:

```
SHOW IPV6 TUNNEL
```

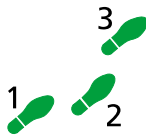
Configuration Examples

The following examples illustrate the steps required to configure IPv6 on the router. The first example shows how to configure basic IPv6 routing, the second shows how to configure dynamic routing with RIP and the third example demonstrates tunnelling IPv6 packets through an IPv4 network. The final example shows how to configure IPv6 filtering on the router.

Before an IPv6 address can be added to a VLAN, the VLAN must be created, using the `CREATE VLAN` command. For more information about VLANs or the `CREATE VLAN` command, see the *Switching* chapter of the Software Reference.

Basic Routing

This example demonstrates configuring a router's VLAN interface with an IPv6 address, and enabling it to access local hosts and external hosts through a gateway device. The gateway device and any other routers which the router needs to communicate with must also be configured with appropriate IPv6 interfaces, addresses and routes.



To configure basic IPv6 routing on the router:

1. Enable the IPv6 module.

Enable IPv6, using the command:

```
ENABLE IPV6
```

2. Add a global IPv6 address to an interface.

To manually add the IPv6 address `3FFE::1/32` to the interface, use the command:

```
ADD IPV6 INTERFACE=vlan1 IPADDRESS=3FFE::1/32
```

3. Check the automatically-created route.

When an interface is added, an interface route is automatically created. Check this route by displaying the routing information, using the command:

```
SHOW IPV6 ROUTE
```

This should produce a display like that shown in Figure 2.

Figure 2: Example output from the `SHOW IPV6 ROUTE` command for a basic IPv6 network.

```

Destination prefix ---> Next Hop
Int.  Age Policy Protocol  Metric Pref Tunnel DLCI Flags
-----
3ffe::/32 ---> ::
vlan1 no 0      interface 1      0    no    -
-----
Codes: P=publish, D=default, A=addrconf, S=stale, L=onlink
N=nonexthop, C=cache, F=flow, U=unknown

```

The router should now be able to communicate with other hosts on the network with the same prefix. To test that this route is functional, use the PING command to access another node in the local network:

```
PING 3FFE::2
```

A stream of replies from the node will be echoed on screen.

4. Add a default route to the local gateway.

The local gateway is the device connecting the local network to other networks such as the Internet. The default route is the route that a packet will be sent to if the router cannot find a route for it. To add a default route, set the ROUTE parameter to `::/0` and the NEXTHOP parameter to the address the packets should be sent to.

To add a default route, when the router is connected to the local LAN via the `vlan1` interface and the local gateway's IPv6 address is `3FFE::2`, use the command:

```
ADD IPV6 ROUTE=::/0 NEXT=3FFE::2 INTERFACE=vlan1
```

5. Test this route with PING.

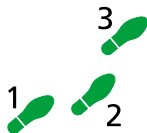
Once a return route has been created from the network `5FFE::/32` to `3FFE::/32`, communications between the networks can be tested, using the command:

```
PING 5FFE::1
```

The router should now be able to access all local hosts with the same prefix, and any host that the gateway device has a route to.

Dynamic Routing with RIPv6

RIP works between routers connected to multiple networks. This example demonstrates configuring three routers' VLAN interfaces with IPv6 addresses, and enabling RIPv6 routing on them so the routers at each extreme end can route to networks to which the others have access.



To configure dynamic routing with RIPv6:

1. Enable the IPv6 module.

Enable IPv6, using the following command on each router:

```
ENABLE IPV6
```

2. Add global IPv6 addresses to each required interface.

Manually add IPv6 addresses to VLAN interfaces on each router. On router A, use the command:

```
ADD IPV6 INTERFACE=vlan2 IPADDRESS=4FFE::1/32
```

On router B, use the commands:

```
ADD IPV6 INTERFACE=vlan2 IPADDRESS=4FFE::2/32
```

```
ADD IPV6 INTERFACE=vlan3 IPADDRESS=5FFE::2/32
```

On router C, use the command:

```
ADD IPV6 INTERFACE=vlan2 IPADDRESS=5FFE::1/32
```

3. Check the routes on each router.

When the interfaces are added, interface routes are automatically created. Check these routes by displaying the routing information, using the following command on each router:

```
SHOW IPV6 ROUTE
```

This should produce a display like that shown in Figure 2 on page 18.

4. Enable RIP on each router and add a RIP interface.

To enable RIP, use the following command on each router:

```
ENABLE IPV6 RIP
```

To create an RIP interface on the network between the routers, use the following commands:

On router A:

```
ADD IPV6 RIP INTERFACE=VLAN2
```

On router B:

```
ADD IPV6 RIP INTERFACE=VLAN2
```

```
ADD IPV6 RIP INTERFACE=VLAN3
```

On router C:

```
ADD IPV6 RIP INTERFACE=VLAN2
```

The RIP updates take 30 seconds to propagate between the routers. After this time, router A should contain a route to router C, and router C a route to router A.

5. Check the routes.

On each router, display the routes to check that the routes appear on the opposite router, using the command:

```
SHOW IPV6 ROUTE
```

The output on router A should be like that shown in Figure 3.

Figure 3: The output of the SHOW IPV6 ROUTE command on router A.

```
IPV6 Routing Table Entries
Destination prefix ---> Next Hop
Int.  Age Policy Protocol  Metric Pref Tunnel DLCI Flags
-----
5ffe::/32 ---> fe80::0200:cdff:fe00:a14d
vlan2 yes 0 ripng 2 100 no -
4ffe::/32 ---> ::
vlan2 no 0 interface 1 0 no -
-----
Codes: P=publish, D=default, A=addrconf, S=stale, L=onlink
N=nonexthop, C=cache, F=flow, U=unknown
```

The output on router C should be like that shown in Figure 4.

Figure 4: The output of the SHOW IPV6 ROUTE command on router C.

```

IPV6 Routing Table Entries
Destination prefix ---> Next Hop
Int.  Age  Policy Protocol  Metric Pref Tunnel DLCI Flags
-----
5ffe::/32 ---> ::
vlan2 no 0 interface 1 0 no -
4ffe::/32 ---> fe80::0200:cdff:fe00:a148
vlan2 yes 0 ripng 2 100 no -
-----
Codes: P=publish, D=default, A=addrconf, S=stale, L=onlink
N=nonexthop, C=cache, F=flow, U=unknown

```

Note that the next hop for the RIPv6 routes is the link-local address for the other router, not the 4ffe::/32 address. This allows routers to share routes even if they aren't on the same logical network.

6. Test these routes with PING.

On router A, use the command:

```
PING 5FFE::1
```

On router C, use the command:

```
PING 4FFE::1
```

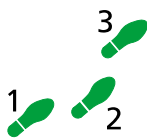
Each router should now be able to communicate with the networks they previously did not have a route to.

Tunnelling over an IPv4 network

A tunnel allows IPv6 packets to be routed between IPv6-aware nodes that are only connected by an IPv4 network. The addresses used in this example are shown in Table 2.

Table 2: IPv4 and IPv6 interfaces and addresses used in this example.

Router	IPv4 address	IPv6 address of tunnel
A	192.168.1.1	3FFE::1
B	192.168.1.2	4FFE::1



To configure a tunnel:

1. Enable IP and add an interface to each router.

Details on configuring basic IPv4 routing can be found in “A Basic TCP/IP Setup” in the *Internet Protocol (IP)* chapter of the Software Reference.

On router A, use the commands:

```

ENABLE IP
ADD IP INTERFACE=VLAN1 IPADDRESS=192.168.1.1

```

On router B, use the commands:

```

ENABLE IP
ADD IP INTERFACE=VLAN1 IPADDRESS=192.168.1.2

```

2. Enable IPv6 on both routers.

Once the IPv4 connection is working correctly, enable IPv6 on each router, using the command:

```
ENABLE IPV6
```

3. Create a tunnel between the two routers.

The LOCAL parameter is the IPv4 address of the router at that end of the tunnel, so for router A the LOCAL parameter is A's IPv4 address and the TARGET parameter is B's IPv4 address. The IPADDRESS parameter is the IPv6 address that will be associated with that end of the tunnel. In order for the PING, TELNET and TRACE commands to function through the tunnel, this address should be on the same network as the router, so for the A end of the tunnel the IPADDRESS parameter should have the same prefix as router A's IPv6 address, and for the B end the same prefix as for router B.

On router A, use the command:

```
ADD IPV6 TUNNEL IPADDRESS=3FFE::1 LOCAL=192.168.1.1
TARGET=192.168.1.2
```

On router B, use the command:

```
ADD IPV6 TUNNEL IPADDRESS=4FFE::1 LOCAL=192.168.1.2
TARGET=192.168.1.1
```

4. Add normal IPv6 interfaces on each router.

To allow testing of the tunnel with the PING command, add at least one interface to each router.

On router A, use the command:

```
ADD IPV6 INTERFACE=VLAN1 IPADDRESS=3FFE::2/32
```

On router B, use the command:

```
ADD IPV6 INTERFACE=VLAN1 IPADDRESS=4FFE::2/32
```

5. Add a route to the other network using the tunnel.

On each router, add an IPv6 route which goes to the tunnel.

On router A, use the command:

```
ADD IPV6 ROUTE=4FFE::/32 NEXT=3FFE::1 INTERFACE=VIRT0
```

On router B, use the command:

```
ADD IPV6 ROUTE=3FFE::/32 NEXT=4FFE::1 INTERFACE=VIRT0
```

Two things are different about this route compared to others that have been added to the router in other examples:

- the NEXTHOP parameter is the IPv6 address of an interface on the router, instead of being the IPv6 address of an interface on another router.
- the INTERFACE parameter is "virt0". This is the virtual interface for all tunnels on the router, so the interface is always "virt0", independent of which tunnel a packet is sent to. The router determines which tunnel to use by the tunnel's IPv6 address.

6. Test the routes.

On router A, use the command:

```
PING 4FFE::2
```

On router B, use the command:

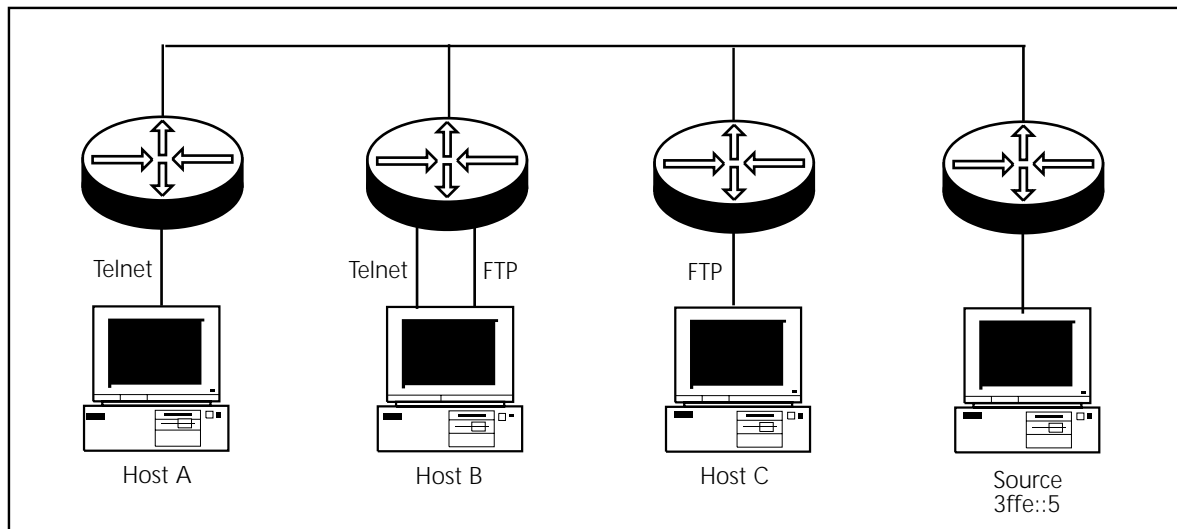
```
PING 3FFE::2
```

IPv6 Filters

The following example illustrates how to configure IPv6 filters on the router.

In this example, there are three hosts. Each host has its own router. Host A can be connected to via Telnet, Host B via Telnet and FTP, and Host C can be connected to via FTP. Only traffic from source address 3ffe::5 will reach the host. This is shown in Figure 5.

Figure 5: Example showing Host and Router connections.



To add an entry to an IPv6 traffic filter or priority filter, use the command:

```
ADD IPV6 FILTER
```

To add a filter to block all Telnet traffic from IP address 3ffe::3 to any IP address whose first 64 contiguous bits match 3ffe::4, and log the header details, use the command:

```
ADD IPV6 FILTER=2 SOURCE=3ffe::3/128 DESTINATION=3ffe::4/64
SIZE=120 PROTOCOL=TCP ACTION=EXCLUDE SPORT=ANY
DPORT=TELNET LOG=HEADER
```

To configure the router with IPv6 filtering, an IPv6 interface must be added. For example, to add the previously-created IPv6 traffic (referred to as FILTER), and priority filters to the interface assigned vlan2, use the command:

```
ADD IPV6 INTERFACE=vlan2 IPADDRESS=3ffe::4/128 FILTER=2
PRIORITYFILTER=201 VALID=10
```

This command adds an IPv6 address to the interface with a specified lifetime (10 seconds). If no time limit is specified, the lifetime is infinite. If the interface has not already been created this command creates one. Valid interfaces are VLAN and PPP.

Filters can be deleted by using the command:

```
DELETE IPV6 FILTER=filter-number [ENTRY=entry-number|ALL]
```

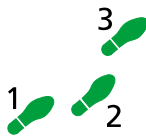
This command deletes an entry from an IP traffic filter or priority filter. The exact entry must already exist in the filter.

To change an entry in an IP traffic filter or priority filter, use the SET command:

```
SET IPV6 FILTER=filter-number ENTRY=entry-number
```

To set a filter to monitor and log all Telnet sessions from IP address `3ffe::3`, to IP address `3ffe::4`, use the command:

```
SET IPV6 FILTER=2 SOURCE=3ffe::3/128 DESTINATION=3ffe::4/128
SIZE=120 PROTOCOL=TCP ACTION=INCLUDE SPORT=ANY
DPORT=TELNET LOG=HEADER
```



To configure IP filters:

1. Create a filter to control the traffic to hosts A, B and C.

Create filters on each of the routers, for the `vlan2` interface in this example, to control the traffic to hosts A, B and C.

To enable Telnet connections to host A (IP address `3FFE::4`) from IPv6 address `3FFE::5`, use the commands:

```
ENABLE IPV6
ADD IPV6 FILTER=1 SOURCE=3ffe::5 SLEN=128
DESTINATION=3ffe::4 DLEN=128 DPORT=TELNET SPORT=ANY
PROTOCOL=TCP SESSION=ANY ACTION=INCLUDE
```

To enable Telnet and FTP access to host B (IP address `3FFE::6`) from IPv6 address `3FFE::5`, use the commands:

```
ADD IPV6 FILTER=1 SOURCE=3ffe::5 SLEN=128
DESTINATION=3ffe::6 DLEN=128 DPORT=FTPDATA SPORT=ANY
PROTOCOL=TCP SESSION=ESTABLISHED ACTION=INCLUDE
ADD IPV6 FILTER=1 SOURCE=3ffe::5 SLEN=128
DESTINATION=3ffe::6 DLEN=128 DPORT=FTP SPORT=ANY
PROTOCOL=TCP SESSION=ANY ACTION=INCLUDE
ADD IPV6 FILTER=1 SOURCE=3ffe::5 SLEN=128
DESTINATION=3ffe::6 DLEN=128 DPORT=TELNET SPORT=ANY
PROTOCOL=TCP SESSION=ANY ACTION=INCLUDE
```

To enable FTP access to host C (IP address `3FFE::7`) from IPv6 address `3FFE::5`, use the commands:

```
ADD IPV6 FILTER=1 SOURCE=3ffe::5 SLEN=128
DESTINATION=3ffe::7 DLEN=128 DPORT=FTP SPORT=ANY
PROTOCOL=TCP SESSION=ESTABLISHED ACTION=INCLUDE
ADD IPV6 FILTER=1 SOURCE=3ffe::5 SLEN=128
DESTINATION=3ffe::7 DLEN=128 DPORT=FTPDATA SPORT=ANY
PROTOCOL=TCP SESSION=ESTABLISHED ACTION=INCLUDE
```

The last entry in a filter is always an implicit entry (one which you do not have to enter) to exclude all sources, destinations and ports. It is equivalent to the command:

```
ADD IPV6 FILTER=1 SOURCE=:: SLEN=128 DESTINATION=::
DLEN=128 PROTOCOL=ANY ACTION=EXCLUDE
```

2. Add the filters to the interfaces

On router A, use the command:

```
ADD IPV6 INTERFACE=VLAN2 IPADDRESS=3ffe::4 PLEN=36 FIL=1
```

On router B, use the command:

```
ADD IPV6 INTERFACE=VLAN2 IPADDRESS=3ffe::6 PLEN=36 FIL=1
```

On router C, use the command:

```
ADD IPV6 INTERFACE=VLAN2 IPADDRESS=3ffe::7 PLEN=36 FIL=1
```

3. Check the configuration

The definitions of the filters on router A can be checked with the command:

```
SHOW IPV6 FILTER
```

This produces an output screen like the one shown in Figure 6.

To see the filter definitions for router B and router C the same command would be used, and a similar output would be shown.

The command:

```
SHOW IPV6 INTERFACE
```

displays details of the IP interfaces defined, including the filter assigned to each interface, as shown in Figure 10 on page 54.

Figure 6: Output of the SHOW IPV6 FILTER command for the Host A IPv6 Filters.

No.	Ent.	SourceAddress	/plen
		Source Port	
		Dest.Address	/plen
		Dest. Port	
		Size	Prot(C/T)
		Options	Session
		Logging	
		Matches	Act/Pri
1	1	3ffe::0005	/128
		Any	
		3ffe::0004	/128
		23	
		Any	TCP
		No	Any
		None	
		0	Include
		Passes: 0	Fails: 0

Figure 7: Output of SHOW IPV6 INTERFACE for Host A.

```

IPV6 Interface Configuration
-----
Physical Interface .....loopback
Ipv6 Interface Index ..... N/A
Link-layer address ..... N/A
EUI-64 Interface Identifier ..... N/A
True MTU/Link MTU ..... 1500/1500
Multicast status ..... Enabled
Send Router Advertisements? ..... No
Ipv6 Interface Addresses:
Int Ipv6 Interface Addresses          plen valid/pref      Sc/St/E/
-----
 0 ::0001                             infinite/infinite loop/pref/Y
-----

IPV6 Interface Configuration
-----
Physical Interface .....vlan2
Ipv6 Interface Index ..... 1
Link-layer address ..... 00-00-cd-00-a0-20
EUI-64 Interface Identifier .....0200CDFFFE00A020
True MTU/Link MTU ..... 1280/1280
Multicast status ..... Enabled
Send Router Advertisements? ..... No
Filter ..... 1
Ipv6 Interface Addresses:
Int Ipv6 Interface Addresses          plen valid/pref      Sc/St/E/
-----
 0 fe80::0200:cdff:fe00:a020          /10 infinite/infinite link/pref/Y
 1 3ffe::0004                          /36 infinite/infinite global/pref/Y

```

Commands

This section describes the commands available to configure and manage the IPv6 functions on the router.

See “*Conventions*” in the *Preface* in the front of the Software Reference for details of the conventions used to describe command syntax. See *Appendix A, Messages* for a complete list of messages and their meanings.

ADD IPV6 FILTER

Syntax `ADD IPV6 FILTER=filter-number`
`SOURCE=ipv6address {/prefixlength | SLEN=prefixlength}`
`[ACTION={ INCLUDE | EXCLUDE } | PRIORITY=P0..P7]`
`[DESTINATION=ipv6address {/prefixlength |`
`DPLEN=prefixlength}] [DPORT={ port-name | port-id | ANY }]`
`[ENTRY=entry-number] [ICMPCODE={ icmp-code-name | icmp-`
`code-id | ANY }] [ICMPATYPE={ icmp-type-name | icmp-type-id |`
`ANY }] [LOG={ 4..1950 | DUMP | HEADER | NONE }] [OPTIONS={ YES |`
`NO }] [PROTOCOL={ protocol | ANY | EGP | ICMP | OSPF | TCP | UDP }]`
`[SESSION={ ANY | ESTABLISHED | START }] [SIZE={ size | ANY }]`
`[SPORT={ port-name | port-id | ANY }]`

where:

- *filter-number* is a number in the range of 0 to 299.
- *ipv6address* is a valid IPv6 address, with its prefix length optionally indicated by slash notation (see “*IPv6 Interfaces and Addresses*” on page 12).
- *port-name* is the predefined name for a TCP or UDP port (see Table 3).
- *port-id* is an IP port number or a range thereof in the format *low:high*.
- *prefixlength* is the decimal value specifying how many of the leftmost contiguous bits of the address comprise the prefix.
- *entry-number* is the position of this entry in the filter.
- *icmp-code-name* is the predefined name for an ICMP reason code (see Table 4 on page -30).
- *icmp-code-id* is the number of an ICMP reason code (see Table 4 on page -30).
- *icmp-type-name* is the predefined name for an ICMP reason type (see Table 5 on page -30).
- *icmp-type-id* is the number of an ICMP reason type (see Table 5 on page -30).
- *protocol* is an IPv6 protocol number.
- *size* is a number in the range 0 to 65535.

Description This command adds an entry to an IPv6 traffic filter or priority filter. The exact pattern within that entry should not already exist in the filter.

The **FILTER** parameter specifies the number of the filter to which the entry is to be added. Filters with numbers in the range 0 to 99 are treated as traffic filters, and use the **ACTION** parameter to specify the action to take with a packet that matches the entry. Filters with numbers in the range 200 to 299 are treated as priority filters, and use the **PRIORITY** parameter to specify the priority to assign to a packet that matches the entry. An interface may have a maximum of one traffic filter and one priority filter, but the same traffic or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface, whereas priority filters are applied to packets as they are transmitted.

The **SOURCE** parameter specifies the source IP address, in IPv6 notation, for the entry. A prefix length must be specified, using either slash notation (e.g. 3FFE::0/16) or the **SPLEN** parameter. If both are used, the two prefix lengths must be the same.

The **ACTION** parameter specifies, for traffic filters, the action to take when the entry is matched. If **INCLUDE** is specified, the IP packet will be processed and forwarded. If **EXCLUDE** is specified, the IP packet will be discarded. The **ACTION** and **PRIORITY** parameters are mutually exclusive—only one may be specified. The default is **INCLUDE**.

The **DESTINATION** parameter specifies the destination IP address, in IPv6 notation, for the entry. A prefix length must be specified, using either slash notation (e.g. 3FFE::0/16) or the **DPLEN** parameter. If both are used, the two prefix lengths must be the same.

The **DPORT** parameter specifies the port to check against the destination port for this entry, as the recognised name of a well-known UDP or TCP port, (see Table 3), a decimal value in the range 0 to 65535, or a range of numbers formatted as *low:high*. If *low* is omitted, 0 is assumed. If *high* is omitted, the maximum port number is assumed. If a port other than **ANY** is specified, the **PROTOCOL** parameter must also be specified, and must be one of **TCP** or **UDP**. The default is **ANY**.

Table 3: Well-known TCP/UDP Ports.

Name	Port	Protocol	Description
ANY	-	-	Any port
BOOTPC	68	UDP	Bootstrap Protocol Client
BOOTPS	67	UDP	Bootstrap Protocol Server
DOMAIN	53	TCP/UDP	Domain Name Server
FINGER	79	TCP	Finger
FTP	21	TCP	File Transfer [Control]
FTPDATA	20	TCP	File Transfer [Default Data]
GOPHER	70	TCP	Gopher
HOSTNAME	101	TCP/UDP	NIC Host Name Server
IPX	213	TCP/UDP	IPX
KERBEROS	88	UDP	Kerberos
LOGIN	49	UDP	Login Host Protocol
MSGICP	29	TCP/UDP	MSG ICP
NAMESERVER	42	UDP	Host Name Server
NEWS	144	TCP	News

Table 3: Well-known TCP/UDP Ports. (Continued)

Name	Port	Protocol	Description
NNTP	119	TCP	Network News Transfer Protocol
NTP	123	TCP	Network Time Protocol
RTELNET	107	TCP/UDP	Remote Telnet Service
SFTP	115	TCP/UDP	Simple File Transfer Protocol
SMTP	25	TCP	Simple Mail Transfer
SNMP	161	UDP	SNMP
SNMPTRAP	162	UDP	SNMPTRAP
SYSTAT	11	TCP	Active Users
TELNET	23	TCP	Telnet
TFTP	69	UDP	Trivial File Transfer
TIME	37	TCP/UDP	Time
UUCP	540	TCP	uucpd
UUCPRLOGIN	541	TCP/UDP	uucp-rlogin
XNSTIME	52	TCP/UDP	XNS Time Protocol

The DPLEN parameter specifies the length of the IPv6 prefix to apply to the destination address for this entry. This number indicates the number of contiguous bits that represent the network address of the destination. If slash notation is used on the DESTINATION parameter, the DPLEN parameter is not required. If the slash notation is not used, the DPLEN parameter is required. The prefix is used to determine the portion of the destination IPv6 address in the IPv6 packet that is significant for comparison with this entry. If DPLEN is specified, DESTINATION must also be specified. The default value is 128.

The ENTRY parameter specifies the entry number which this new entry occupies in the filter. Existing entries with the same or higher entry numbers are pushed down the filter. The default is to add the new entry to the end of the filter.

The ICMPATYPE and ICMPCODE parameters specify the ICMP message type and ICMP message reason code (see Table 4) to match against the ICMP type and code fields in an ICMP packet. The ICMPATYPE parameter specifies the ICMP message type to match, as a decimal value in the range 0 to 65535, or the recognised name of an ICMP type (see Table 5). A packet is only matched against type and code when the PROTOCOL parameter is set to ICMP. The default is ANY.

Table 4: Predefined ICMP Code Names Used by the IPV6 Filtering Process.

ICMP code name	ICMP code value	Applies to ICMP type value	Command Line Syntax
ICMPv6_ANY	-	-	ANY
ICMPv6_NO_ROUTE_TO_DESTINATION	0	1	NOROUTETODEST
ICMPv6_COMMUNICATION_PROHIBITED	1	1	COMMSPROHIBITED
ICMPv6_SCOPE_MISMATCH	2	1	SCOPEMISMATCH
ICMPv6_ADDRESS_UNREACHABLE	3	1	ADDRUNREACHABLE
ICMPv6_PORT_UNREACHABLE	4	1	PORTUNREACHABLE
ICMPv6_HOP_LIMIT_EXCEEDED	0	3	HOPLIMITEXCD
ICMPv6_REASSEMBLY_TIME_EXCEEDED	1	3	REASMBTIMEEXC
ICMPv6_ERRONEOUS_HEADER_FIELD	0	4	ERRONEOUSHEADER
ICMPv6_UNRECOGNISED_NEXT_HEADER	1	4	URCNXTHEADER
ICMPv6_UNRECOGNISED_OPTION	2	4	URCOPTION

Table 5: Predefined ICMP Type Names Used by the IPV6 Filtering Process.

ICMP Type Name	ICMP Type Value	ICMP Codes Supported	Command Line Syntax
ICMPv6_ANY	-	-	ANY
ICMPv6_DESTINATION_UNREACHABLE	1	Yes	DESTUNREACH
ICMPv6_PACKET_TOO_BIG	2	Yes	PKTTOOBIG
ICMPv6_TIME_EXCEEDED	3	Yes	TIMEEXCEEDED
ICMPv6_PARAMETER_PROBLEM	4	Yes	PARAMPROB
ICMPv6_ECHO_REQUEST	128	No	ECHORQ
ICMPv6_ECHO_REPLY	129	No	ECHORP
ICMPv6_MULTICAST_LISTENER_QUERY	130	No	MLQUERY
ICMPv6_MULTICAST_LISTENER_REPORT	131	No	MLREP
ICMPv6_MULTICAST_LISTENER_DONE	132	No	MLDONE
ICMPv6_ROUTER_SOLICIT	133	No	RTSOLICIT
ICMPv6_ROUTER_ADVERT	134	No	RTADVERT
ICMPv6_NEIGHBOR_SOLICIT	135	No	NBR SOLICIT
ICMPv6_NEIGHBOR_ADVERT	136	No	NBRADVERT
ICMPv6_REDIRECT	137	No	REDIRECT
ICMPv6_ROUTER_RENUMBERING	138	No	RTRENUMBER

The LOG parameter specifies whether or not any matches to a filter entry result in a log message being sent to the router's logging facility, and the content of the log messages. This parameter enables logging of the IP packet filtering process down to the level of an individual filter entry.

- If a number in the range 4 to 1950 is specified, the first 4 to 1950 octets of the data portion of TCP, UDP and ICMP packets or the first 4 to 1950 octets after the IP header of other protocol packets are logged with a message type/subtype of IPFIL/DUMP. The filter number, entry

number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) are also logged with a message type/subtype of IPFIL/PASS (for entries with an INCLUDE action) or IPFIL/FAIL (for entries with an EXCLUDE action).

- If DUMP is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) are logged with a message type/subtype of IPFIL/PASS (for entries with an INCLUDE action) or IPFIL/FAIL (for entries with an EXCLUDE action). In addition, the first 40 octets of the data portion of TCP, UDP and ICMP packets, or the first 40 octets after the IP header of other protocol packets are logged with a message type/subtype of IPFIL/DUMP.
- If HEADER is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) are logged with a message type/subtype of IPFIL/PASS (for entries with an INCLUDE action) or IPFIL/FAIL (for entries with an EXCLUDE action).
- If NONE is specified, matches to the filter entry are not logged.
- The default is NONE.

The OPTIONS parameter specifies the presence or absence of any *time-length-variable* (TLV) encoded “Options” to check against for this entry. TLV encoded options can be found in the Hop-by-Hop and Destination Options extension headers. If YES is specified, the entry matches IP packets with any extension header TLV options set. If NO is specified, the entry matches IP packets without any extension header TLV options set. The default is NO.

The PRIORITY parameter specifies, for priority filters, the priority to apply to forwarding packets when the entry is matched. A low value (P0) assigns a high priority to the packet. A high value (P7) assigns a low priority to the packet. The priority number is employed during forwarding (transmission). The default is P7. The ACTION and PRIORITY parameters are mutually exclusive—only one may be specified.

The PROTOCOL parameter specifies a protocol to check against the protocol for this entry, as a decimal value in the range 0 to 65535, or the recognised name of an IP protocol type. If either the SPORT or DPORT parameters are used, PROTOCOL must be defined as TCP, UDP or ANY. Specifying TCP or UDP will filter out packets from companion protocols, such as ICMP and OSPF, that do not use TCP or UDP as a transport mechanism. The default is ANY.

The SESSION parameter specifies the type of TCP packet to match, and is only used as a basis for packet filtering when the PROTOCOL parameter specifies TCP. If START is specified, the entry matches TCP packets with the SYN bit set and the ACK bit clear. If ESTABLISHED is specified, the entry matches TCP packets with either the SYN bit clear or the ACK bit set. If ANY is specified, the entry matches any TCP packet. The default is ANY.

The SIZE parameter specifies the maximum reassembled size to match against, for each IP fragment. If the fragment’s offset plus size is greater than the value specified, the fragment is discarded. The default is ANY, which indicates that size is not required as a matching category.

The SPORT parameter specifies the port to check against the source port for this entry, as the recognised name of a well-known UDP or TCP port, (see Table 3), a decimal value in the range 0 to 65535, or a range of numbers in the

form *low:high*. If *low* is omitted, 0 is assumed. If *high* is omitted, the maximum port number is assumed. If a port other than ANY is specified, the PROTOCOL parameter must also be specified, and must be one of TCP or UDP. The default is ANY.

The SLEN parameter specifies the length of the IPv6 prefix to apply to the source address for this entry. This number indicates the number of contiguous bits that represent the network address of the source. If slash notation is used on the SOURCE parameter, the SLEN parameter is not required. If the slash notation is not used, the SLEN parameter is required. The prefix is used to determine the portion of the source IPv6 address in the IPv6 packet that is significant for comparison with this entry. If SLEN is specified, SOURCE must also be specified. The default is 128.

Examples To add a filter to block all Telnet traffic, from IP address *3ffe::3* to any IP address whose first 64 contiguous bits match *3ffe::4*, and log the header details, use the command:

```
ADD IPV6 FILTER=2 SOURCE=3ffe::3/128 DESTINATION=3ffe::4/64
    SIZE=ANY PROTOCOL=TCP ACTION=EXCLUDE SPORT=ANY
    DPORT=TELNET LOG=HEADER
```

See Also ADD IPV6 INTERFACE
DELETE IPV6 FILTER
SET IPV6 FILTER
SHOW IPV6 FILTER

ADD IPV6 HOST

Syntax ADD IPV6 HOST=*name* IPADDRESS=*ipv6address*

where:

- *name* is a character string up to 60 characters in length. Valid characters are any printable characters. If the string contains spaces it must be enclosed in double quotes.
- *ipv6address* is a valid IPv6 address.

Description This command adds a user-defined name for an IPv6 host to the router's host name table. The host name table makes it easier to Telnet to commonly accessed hosts, by enabling the user to enter a shorter, easier to remember name for the host rather than the host's full IPv6 address or domain name.

The HOST parameter specifies the user-defined name for the IPv6 host. A host with the same name must not already exist in the host name table. When a host name is specified in the TELNET command, (see the TELNET command, described in the *Terminal Server* chapter of the Software Reference), the entire name will be used to match a name in the host name table. All characters are used in the comparison, including non-alphabetic characters if they are present. The HOST parameter is case insensitive, so the names *MyName* and *myname* are identical.

The IPADDRESS parameter specifies the IPv6 address of the host.

Examples To add a new host called *foobar* with an IPv6 address of 3FFE::1, use the command:

```
ADD IPV6 HOST=foobar IPADDRESS=3FFE::1
```

See also DELETE IPV6 HOST
SHOW IPV6 HOST

ADD IPV6 INTERFACE

Syntax ADD IPV6 INTERFACE=*interface*
IPADDRESS=*ipv6address*{/*prefixlength*|PLEN=*prefixlength*}
[FILTER=*filtNum*] [PREFERRED=*seconds*|INFINITE]
[PRIORITYFILTER=*priNum*] [PUBLISH={YES|NO}]
[VALID=*seconds*|INFINITE]

where:

- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. vlan0). Valid interface types are VLAN and PPP.
- *ipv6address* is a valid IPv6 address, with its prefix length optionally indicated by slash notation (see “IPv6 Interfaces and Addresses” on page 12).
- *prefixlength* is an integer between 1 and 128.
- *filtNum* is an integer between 0 and 99.
- *seconds* is a time in seconds up to 4294967295.
- *priNum* is an integer between 200 and 299.

Description This command adds an IPv6 address to an interface, with a specified lifetime. If no lifetime is specified then the lifetime is infinite. If the interface has not already been created, this command will also perform the creation.

The INTERFACE parameter is the interface to add the IPv6 address to. Valid interface types are VLAN and PPP.

The IPADDRESS parameter specifies the IPv6 address to assign to the interface, in IPv6 notation. A prefix length must be specified, using either slash notation (e.g. 3FFE::0/16) or the PLEN parameter. If both are used, the two prefix values must be the same.

The PLEN parameter is the length of the IPv6 prefix for this interface. This number is the number of contiguous bits that represent the network address of the network to which the interface is attached. If slash notation is used on the IPADDRESS parameter, the PLEN parameter is not required. If slash notation is not used, the PLEN parameter is required. If both are used, the value must be the same.

The FILTER parameter is the number of the traffic filter that is to be applied to this interface.

The PREFERRED parameter specifies the time in seconds for which this IPv6 address will be the preferred address for this interface. The default is INFINITE. The value of this parameter cannot be greater than that of the VALID parameter.

The PRIORITYFILTER parameter is the number of the priority filter that is to be attached to this interface.

The PUBLISH parameter determines whether or not to include this prefix in router advertisement packets. The default is NO.

The VALID parameter is the time in seconds that the IPv6 address exists on the interface. After the time specified expires, the IPv6 address is deleted. The default is INFINITE. The time value must be the same as or greater than that of the PREFERRED parameter.

Examples To add an IPv6 address of 3FFE::1/32 to *vlan2*, use one of the commands:

```
ADD IPV6 IPV6ADDRESS=3FFE::/32 INTERFACE=VLAN2
ADD IPV6 IPV6ADDRESS=3FFE:: INTERFACE=VLAN2 PLEN=32
```

See Also CREATE IPV6 INTERFACE
DELETE IPV6 INTERFACE
DESTROY IPV6 INTERFACE
SET IPV6 INTERFACE
SHOW IPV6
SHOW IPV6 INTERFACE

ADD IPV6 RIP

Syntax ADD IPV6 RIP INTERFACE=*interface*

where:

- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. *vlan0*). Valid interface types are VLAN, PPP and VIRT. VIRT is the interface type of a tunnel.

Description This command enables listening for RIPv6 packets on the specified interface.

The INTERFACE parameter is the physical interface to listen for RIP packets on.

Example To enable listening for RIP packets on *vlan2*, use the command:

```
ADD IPV6 RIP INTERFACE=VLAN2
```

See Also DELETE IPV6 RIP
DISABLE IPV6 RIP
ENABLE IPV6 RIP
SHOW IPV6 RIP

ADD IPV6 ROUTE

Syntax `ADD IPV6 ROUTE=ipv6address{/prefixlength|
PLEN=prefixlength} INTERFACE=interface
NEXTHOP=ipv6address [METRIC=1..16]
[PREFERENCE=0..65535]`

where:

- *ipv6address* is a valid IPv6 address, with its prefix length optionally indicated by slash notation (see “IPv6 Interfaces and Addresses” on page 12).
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. vlan0). Valid interface types are VLAN, PPP and VIRT. VIRT is the interface type of a tunnel.
- *prefixlength* is an integer between 1 and 128.

Description This command adds a static route to the IPv6 route table. The static route must not already exist. However, if the route exists as a dynamic (e.g. RIP-derived) route, the static route may still be added. To specify that the destination node is on the network to which traffic is being directed, the NEXTHOP parameter should be 0:0:0:0:0:0:0 (or ::).

Static routes can be used to define default routes to external routers or networks. A default route is one with a network address of 0:0:0:0:0:0:0 (or ::). If the router receives data and cannot find a route for it, the data will be sent to the default route. To define a default route, set ROUTE to :: and set NEXTHOP to point to the network (router) to which default packets are to be directed.

The ROUTE parameter is the IPv6 address of a host or network to which packets are to be routed. Slash notation can be used on this parameter.

The INTERFACE parameter is the physical interface out of which the router will forward packets.

The NEXTHOP parameter is the IPv6 address of the next router along the path to the destination.

The METRIC parameter specifies the cost of the route. The cost is used in RIP entries to determine the best path to a node. The default is 1.

The PLEN parameter is the number of contiguous leftmost bits of an IPv6 address that represents the prefix for all IPv6 addresses on this network, similar to a subnet mask in IPv4. Using this parameter allows static routing of traffic to a specific network. If slash notation is used in the ROUTER parameter, this parameter is not necessary. If the PLEN parameter and the number after a slash on the ROUTE are different, the route will not be added and a warning message will be printed.

The PREFERENCE parameter specifies the preference for the route. When more than one route in the route table matches the destination address in an IPv6 packet, the route with the lowest preference value will be used to route the packet. If two or more candidate routes have the same preference, the route with the longest prefix will be used. Interface routes have a preference of 0 and RIP routes have a preference of 100. The default preference for static routes other than 0:0:0:0:0:0:0 is 60. The default for the default static route 0:0:0:0:0:0:0 is 360.

Example To add a route to the network 3FFE::/16, using a gateway with the IPv6 address 4FFE::1 that is connected to the router on vlan2, use the command:

```
ADD IPV6 ROUTE=3FFE:: PLEN=16 NEXT=4FFE::1 INTERFACE=VLAN2
```

See Also DELETE IPV6 ROUTE
SHOW IPV6
SHOW IPV6 ROUTE

ADD IPV6 TUNNEL

Syntax ADD IPV6 TUNNEL IPADDRESS=*ipv6address* LOCAL=*ipv4address*
TARGET=*ipv4address*

where:

- *ipv6address* is a valid IPv6 address.
- *ipv4address* is a valid IPv4 address in dotted decimal notation.

Description This command allows two IPv6 networks to be linked over an IPv4 network. A tunnel must be created on both of the IPv6 routers which link the two networks, and a route that uses the tunnel must also be added on each router.

The IPADDRESS parameter is the IPv6 address of the tunnel, to be used when adding a route over the tunnel. This address is also the source address for TELNET, and for PING and TRACE when no other source is specified.

The LOCAL parameter is the IPv4 address of the interface on the router which will be used in the tunnel.

The TARGET parameter is the IPv4 address of the last node in the tunnel. The end node must be capable of forwarding an IPv6 packet. A tunnel must also be configured from the end node back to the router.

Example To add a tunnel between two networks, 3ffe::/64 (router A) and 4ffe::/64 (router B), use the commands:

```
A> add ip interface=vlan2 ipaddress=192.168.1.1
A> add ipv6 tunnel ip=3ffe::1 local=192.168.1.1
    target=192.168.1.2
A> add ipv6 route=4ffe::/64 next=3ffe::1 interface=virt0
B> add ip interface=vlan2 ipaddress=192.168.1.2
B> add ipv6 tunnel ipaddress=4ffe::1 local=192.168.1.2
    target=192.168.1.1
B> add ipv6 route=3ffe::/64 next=4ffe::1 interface=virt0
```

See Also ADD IPV6 ROUTE
DELETE IPV6 TUNNEL
SHOW IPV6 TIMER

CREATE IPV6 INTERFACE

Syntax `CREATE IPV6 INTERFACE=interface`

where:

- *interface* is an interface name formed by concatenating VLAN and an interface instance (e.g. vlan0).

Description This command creates an IPv6 VLAN interface and uses Stateless Address autoconfiguration to assign it a link-local address. The address is formed from the link-layer address and the prefix FE80::.

To create a PPP interface and assign it an IPv6 address, use the `ADD IPV6 INTERFACE` command on page -33. To create a VIRT interface, use the `ADD IPV6 TUNNEL` command on page -36.

The `INTERFACE` parameter specifies the VLAN interface that is to be created for IPv6. Routing using this interface will then be possible.

Example To create an IPv6 interface, use the command:

```
CREATE IPV6 INTERFACE=VLAN2
```

See Also `ADD IPV6 INTERFACE`
`ADD IPV6 TUNNEL`
`DELETE IPV6 INTERFACE`
`DESTROY IPV6 INTERFACE`
`SET IPV6 INTERFACE`
`SHOW IPV6`
`SHOW IPV6 INTERFACE`

DELETE IPV6 FILTER

Syntax `DELETE IPV6 FILTER=filter-number ENTRY={entry-number|ALL}`

where:

- *filter-number* is a number between 0 and 299.
- *entry-number* is the position of this entry in the filter.

Description This command deletes an entry from an IP traffic filter or priority filter. The entry must already exist in the filter.

The `FILTER` parameter specifies the number of the filter from which the entry is to be deleted. Filters with numbers in the range 0 to 99 are traffic filters and filters with numbers in the range 200 to 299 are priority filters.

The `ENTRY` parameter specifies the entry number in the filter that is to be deleted. Existing entries with the same or higher entry numbers will be pushed up the filter to occupy the vacant entry. If `ALL` is specified, the filter is deleted.

Examples To delete *entry 2* from *filter 2*, use the command:

```
DELETE IPV6 FILTER=2 ENTRY=2
```

To delete all entries from *filter 2*, use the command:

```
DELETE IPV6 FILTER=2 ENTRY=ALL
```

See Also ADD IPV6 FILTER
SET IPV6 FILTER
SHOW IPV6 FILTER

DELETE IPV6 HOST

Syntax DELETE IPV6 HOST=*name*

where:

- *name* is a character string, 1 to 60 characters in length.

Description This command removes a host entry.

The HOST parameter is a name that has been associated with an IPv6 address, using the command ADD IPV6 HOST.

Examples To remove the *foobar* host entry, use the command:

```
DELETE IPV6 HOST=FOOBAR
```

See Also ADD IPV6 HOST
SHOW IPV6 HOST

DELETE IPV6 INTERFACE

Syntax DELETE IPV6 INTERFACE=*interface* IPADDRESS=*ipv6address*

where:

- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. vlan0). Valid interface types are VLAN, PPP and VIRT. VIRT is the interface type of a tunnel.
- *ipv6address* is a valid IPv6 address.

Description This command removes an IPv6 address from an interface.

The INTERFACE parameter specifies the interface on the router to remove the IPv6 address from.

The IPADDRESS parameter specifies the IPv6 address to remove from the interface.

Examples To remove the address *3ffe::1* from the *vlan2* interface, use the command:

```
DELETE IPV6 INTERFACE=VLAN2 IPADDRESS=3FFE::1
```

See Also ADD IPV6 INTERFACE
CREATE IPV6 INTERFACE
DESTROY IPV6 INTERFACE
SET IPV6 INTERFACE
SHOW IPV6

DELETE IPV6 RIP

Syntax DELETE IPV6 RIP INTERFACE=*interface*

where:

- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. *vlan0*). Valid interface types are VLAN and PPP.

Description This command stops an interface from listening for or sending RIP packets.

The INTERFACE parameter specifies the physical interface that the router was receiving RIP packets on.

Examples To stop an interface listening for or sending RIP packets, use the command:

```
DELETE IPV6 RIP INTERFACE=VLAN2
```

See Also ADD IPV6 RIP
DISABLE IPV6 RIP
ENABLE IPV6 RIP
SHOW IPV6 INTERFACE

DELETE IPV6 ROUTE

Syntax DELETE IPV6 ROUTE=*ipv6address* INTERFACE=*interface*
NEXTTHOP=*ipv6address*

where:

- *ipv6address* is a valid IPv6 address.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. *vlan0*). Valid interface types are VLAN and PPP.

Description This command removes a route from the IPv6 route table.

The ROUTE parameter specifies the IPv6 address of a host or network to which packets were being routed.

The INTERFACE parameter specifies the physical interface out of which the router was forwarding packets.

The NEXTHOP parameter specifies the IPv6 address of the router that was next on the path to the destination.

Examples To delete the route to 3ffe:: with a next hop of 3ffe::1 on vlan2, use the command:

```
DELETE IPV6 ROUTE=3FFE:: NEXT=3FFE::1 INTERFACE=VLAN2
```

See Also ADD IPV6 ROUTE
SHOW IPV6 ROUTE

DELETE IPV6 TUNNEL

Syntax DELETE IPV6 TUNNEL=*ipv6address*

where:

- *ipv6address* is a valid IPv6 address.

Description This command removes an IPv6 tunnel.

The TUNNEL parameter specifies the IPv6 address of the tunnel to be removed. The tunnel's IPv6 address is the address that was specified with the IPADDRESS parameter of the ADD IPV6 TUNNEL command, when the tunnel was created.

Example To delete the tunnel with an IPv6 address of 3FEE::1, use the command:

```
DELETE IPV6 TUNNEL=3FEE::1
```

See Also ADD IPV6 TUNNEL
SHOW IPV6 TIMER

DESTROY IPV6 INTERFACE

Syntax DESTROY IPV6 INTERFACE=*interface*

where:

- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. vlan0). Valid interface types are VLAN and PPP.

Description This command removes an IPv6 interface.

The INTERFACE parameter specifies the physical interface that is to be removed from IPv6. Routing using this interface will no longer be possible.

Examples To remove the *vlan2* interface, use the command:

```
DESTROY IPV6 INTERFACE=VLAN2
```

See Also ADD IPV6 INTERFACE
CREATE IPV6 INTERFACE
DELETE IPV6 INTERFACE
SET IPV6 INTERFACE
SHOW IPV6 INTERFACE

DISABLE IPV6

Syntax DISABLE IPV6

Description This command disables the IPv6 module. Any dynamic configuration, such as address autoconfiguration, is preserved between disabling and re-enabling the IPv6 module.

The IPv6 module is disabled by default.

Examples To disable the IPv6 module, use the command:

```
DISABLE IPV6
```

See Also ENABLE IPV6
SHOW IPV6

DISABLE IPV6 ADVERTISE

Syntax DISABLE IPV6 ADVERTISE

Description This command disables sending of router advertisement packets. Router advertisements are enabled by default.

Examples To disable router advertisements, use the command:

```
DISABLE IPV6 ADVERTISE
```

See Also ENABLE IPV6 ADVERTISE
SHOW IPV6

DISABLE IPV6 DEBUG

Syntax DISABLE IPV6 DEBUG

Description This command disables debugging of IPv6 packets.

Debugging is disabled by default.

Examples To disable IPv6 debugging, use the command:

```
DISABLE IPV6 DEBUG
```

See Also SHOW IPV6
ENABLE IPV6 DEBUG

DISABLE IPV6 RIP

Syntax DISABLE IPV6 RIP

Description This command disables routing of IPv6 packets using RIPng. By default RIP is disabled.

Examples To disable the sending and receiving of RIP packets, use the command:

```
DISABLE IPV6 RIP
```

See Also ADD IPV6 RIP
DELETE IPV6 RIP
ENABLE IPV6 RIP
SHOW IPV6 RIP

ENABLE IPV6

Syntax ENABLE IPV6

Description This command enables the IPv6 module. Any dynamic configuration is preserved between disabling and re-enabling the IPv6 module.

By default the IPv6 module is disabled.

Examples To enable IPv6, use the command:

```
ENABLE IPV6
```

See Also DISABLE IPV6
SHOW IPV6

ENABLE IPV6 ADVERTISE

Syntax `ENABLE IPV6 ADVERTISE`

Description This command enables sending of router advertisement packets.

By default router advertisements are enabled.

Examples To enable the sending of router advertisements, use the command:

```
ENABLE IPV6 ADVERTISE
```

See Also `DISABLE IPV6 ADVERTISE`
`SHOW IPV6`

ENABLE IPV6 DEBUG

Syntax `ENABLE IPV6 DEBUG`

Description This command enables debugging of IPv6 packets.

Debugging is disabled by default.

Examples To enable debugging, use the command:

```
ENABLE IPV6 DEBUG
```

See Also `DISABLE IPV6 DEBUG`
`SHOW IPV6`

ENABLE IPV6 RIP

Syntax `ENABLE IPV6 RIP`

Description This command enables routing of IPv6 packets using RIPv6.

RIPv6 is disabled by default.

Examples To enable the sending and receiving of RIPv6 packets, use the command:

```
ENABLE IPV6 RIP
```

See Also `ADD IPV6 RIP`
`DELETE IPV6 RIP`
`DISABLE IPV6 RIP`
`SHOW IPV6 RIP`

SET IPV6 FILTER

Syntax SET IPV6 FILTER=*filter-number* ENTRY=*entry-number*
 [ACTION={INCLUDE|EXCLUDE}|PRIORITY=P0..P7]
 [DESTINATION=*ipv6address*{/*prefixlength*|
 DPLEN=*prefixlength*} [DPORT={*port-name*|*port-id*|ANY}]
 [ICMPCODE={*icmp-code-name*|*icmp-code-id*|ANY}]
 [ICMPTYPE={*icmp-type-name*|*icmp-type-id*|ANY}]
 [LOG={4..1950|DUMP|HEADER|NONE}] [OPTIONS={YES|NO}]
 [PROTOCOL={*protocol*|ANY|EGP|ICMP|OSPF|TCP|UDP}]
 [SESSION={ANY|ESTABLISHED|START}][SIZE={*size*|ANY}]
 [SOURCE=*ipv6address*{/*prefixlength*|SPLEN=*prefixlength*}]
 [SPORT={*port-name*|*port-id*|ANY}]

where:

- *filter-number* is an integer between 0 and 99 or 200 and 299.
- *entry-number* is the position of this entry in the filter.
- *ipv6address* is a valid IPv6 address, with its prefix length optionally indicated by slash notation (see “IPv6 Interfaces and Addresses” on page 12).
- *port-name* is the predefined name of a TCP or UDP port.
- *port-id* is an IP port number or a range thereof in the format *low:high*.
- *prefixlength* is an integer between 1 and 128.
- *icmp-code-name* is the predefined name for an ICMP reason code (see Table 4).
- *icmp-code-id* is the number of an ICMP reason code (see Table 4).
- *icmp-type-name* is the predefined name of an ICMP message type (see Table 5).
- *icmp-type-id* is the number of an ICMP message type (see Table 5).
- *protocol* is an IPv6 protocol number.
- *size* is an integer between 0 and 65535.

Description This command changes an entry in an IP traffic filter or priority filter.

The FILTER parameter specifies the number of the filter in which the entry is to be changed. Filters with numbers in the range 0 to 99 are treated as traffic filters, and use the ACTION parameter to specify the action to take with a packet that matches the entry. Filters with numbers in the range 200 to 299 are treated as priority filters, and use the PRIORITY parameter to specify the priority to assign to a packet that matches the entry. An interface may have a maximum of one traffic filter and one priority filter. The same traffic or priority filter can be assigned to more than one interface. Traffic filters are applied to packets received via the interface, whereas priority filters are applied to packets as they are transmitted.

The ENTRY parameter specifies the entry number to be changed.

The ACTION parameter specifies, for traffic filters, the action to take when the entry is matched. If INCLUDE is specified, the IP packet will be processed and forwarded. If EXCLUDE is specified, the IP packet will be discarded. The ACTION and PRIORITY parameters are mutually exclusive—only one may be specified. The default is INCLUDE.

The **DESTINATION** parameter specifies the destination IP address, in IPv6 notation, for the entry. A prefix length must be specified, using either slash notation (e.g. 3FFE::0/16) or the **DPLEN** parameter. If both are used, the two prefix lengths must be the same.

The **DPLEN** parameter specifies the length of the IPv6 prefix to apply to the destination address for this entry. This number indicates the number of contiguous bits that represent the network address of the destination. If slash notation is used on the **DESTINATION** parameter, the **DPLEN** parameter is not required. If the slash notation is not used, the **DPLEN** parameter is required. The prefix is used to determine the portion of the destination IPv6 address in the IPv6 packet that is significant for comparison with this entry. If **DPLEN** is specified, **DESTINATION** must also be specified. The default is 128.

The **DPORT** parameter specifies the port to check against the destination port for this entry, as the recognised name of a well-known UDP or TCP port, (see Table 3), a decimal value in the range 0 to 65535, or a range of numbers in the form *low:high*. If *low* is omitted, 0 is assumed. If *high* is omitted, the maximum port number is assumed. If a port other than ANY is specified, the **PROTOCOL** parameter must also be specified, and must be one of TCP or UDP. The default is ANY.

The **ICMP CODE** parameter specifies the ICMP message reason code to match against the ICMP code field in an ICMP packet, as a decimal value in the range 0 to 65535, or the recognised name of an ICMP reason code (see Table 4 on page -30). This parameter is only used as a basis for packet matching when the **PROTOCOL** parameter is set to ICMP. The default is ANY.

The **ICMP TYPE** parameter specifies the ICMP message type to match against the ICMP type field in an ICMP packet header, as a decimal value in the range 0 to 65535 or the recognised name of an ICMP type (see Table 5 on page -30). This parameter is only used as a basis for packet matching when the **PROTOCOL** parameter is set to ICMP. The default is ANY.

The **LOG** parameter specifies whether or not any matches to a filter entry result in a log message being sent to the router's logging facility, and the content of the log messages. This parameter enables logging of the IP packet filtering process down to the level of an individual filter entry.

- If a number in the range 4 to 1950 is specified, the first 4 to 1950 octets of the data portion of TCP, UDP and ICMP packets or the first 4 to 1950 octets after the IP header of other protocol packets are logged with a message type/subtype of IPFIL/DUMP. The filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) are also logged with a message type/subtype of IPFIL/PASS (for entries with an INCLUDE action) or IPFIL/FAIL (for entries with an EXCLUDE action).
- If DUMP is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) are logged with a message type/subtype of IPFIL/PASS (for entries with an INCLUDE action) or IPFIL/FAIL (for entries with an EXCLUDE action). In addition, the first 40 octets of the data portion of TCP, UDP and ICMP packets, or the first 40 octets after the IP header of other protocol packets are logged with a message type/subtype of IPFIL/DUMP.

- If HEADER is specified, the filter number, entry number and IP header information (source and destination IP addresses, protocol, source and destination ports, and size) are logged with a message type/subtype of IPFIL/ PASS (for entries with an INCLUDE action) or IPFIL/FAIL (for entries with an EXCLUDE action).
- If NONE is specified, matches to the filter entry are not logged.
- The default is NONE.

The OPTIONS parameter specifies the presence or absence of any *time-length-variable* (TLV) encoded “Options” to check against for this entry. TLV encoded options can be found in the Hop-by-Hop and Destination Options extension headers. If YES is specified, the entry matches IP packets with any extension header TLV options set. If NO is specified, the entry matches IP packets without any extension header TLV options set. The default is NO.

The PRIORITY parameter specifies, for priority filters, the priority to apply to forwarding packets when the entry is matched. A low value (P0) assigns a high priority to the packet. A high value (P7) assigns a low priority to the packet. The priority number is employed during forwarding (transmission). The default is P7. The ACTION and PRIORITY parameters are mutually exclusive—only one may be specified.

The PROTOCOL parameter specifies a protocol to check against the protocol for this entry, as a decimal value in the range 0 to 65535, or the recognised name of an IP protocol type. If either the SPORT or DPORT parameters are used, PROTOCOL must be defined as TCP, UDP or ANY. Specifying TCP or UDP will filter out packets from companion protocols, such as ICMP and OSPF, that do not use TCP or UDP as a transport mechanism. The default is ANY.

The SESSION parameter specifies the type of TCP packet to match, and is only used as a basis for packet filtering when the PROTOCOL parameter specifies TCP. If START is specified, the entry matches TCP packets with the SYN bit set and the ACK bit clear. If ESTABLISHED is specified, the entry matches TCP packets with either the SYN bit clear or the ACK bit set. If ANY is specified, the entry matches any TCP packet. The default is ANY.

The SIZE parameter specifies the maximum reassembled size to match against, for each IP fragment. If the fragment’s offset plus size is greater than the value specified, the fragment is discarded. The default is ANY, which indicates that size is not required as a matching category.

The SOURCE parameter specifies the source IP address, in IPv6 notation, for the entry. A prefix length must be specified, using either slash notation (e.g. 3FFE::0/16) or the SPLLEN parameter. If both are used, the two prefix values must be the same.

The SPLLEN parameter specifies the length of the IPv6 prefix to apply to the source address for this entry. This number indicates the number of contiguous bits that represent the network address of the source. If slash notation is used on the SOURCE parameter, the SPLLEN parameter is not required. If the slash notation is not used, the SPLLEN parameter is required. The prefix is used to determine the portion of the source IPv6 address in the IPv6 packet that is significant for comparison with this entry. If SPLLEN is specified, SOURCE must also be specified. The default is 128.

The SPORT parameter specifies the port to check against the source port for this entry, as the recognised name of a well-known UDP or TCP port, (see Table 3), a decimal value in the range 0 to 65535, or a range of numbers in the

form *low:high*. If *low* is omitted, 0 is assumed. If *high* is omitted, the maximum port number is assumed. If a port other than ANY is specified, the PROTOCOL parameter must also be specified, and must be one of TCP or UDP. The default is ANY.

Examples To set a filter to monitor and log all Telnet sessions from IP address *3ffe::3*, to IP address *3ffe::4*, use the command:

```
SET IPV6 FILTER=2 SOURCE=3ffe::3/128 DESTINATION=3ffe::4/64
  SIZE=ANY PROTOCOL=TCP ACTION=INCLUDE SPORT=ANY DPORT=23
  LOG=HEADER
```

See Also ADD IPV6 FILTER
DELETE IPV6 FILTER
SHOW IPV6 FILTER

SET IPV6 INTERFACE

Syntax SET IPV6 INTERFACE=*interface*
IPADDRESS=*ipv6address*{/*prefixlength*|PLEN=*prefixlength*}
[FILTER=*filtNum*] [PREFERRED=*seconds*]
[PRIORITYFILTER=*priNum*] [VALID=*seconds*]

where:

- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. vlan0). Valid interface types are VLAN and PPP.
- *ipv6address* is a valid IPv6 address, with its prefix length optionally indicated by slash notation (see “IPv6 Interfaces and Addresses” on page 12).
- *prefixlength* is an integer between 1 and 128.
- *filtNum* is an integer between 0 and 99.
- *seconds* is a time in seconds up to 4294967295.
- *priNum* is an integer between 200 and 299.

Description This command modifies various values associated with an interface that was previously created by a call to either the CREATE IPV6 INTERFACE command, or the ADD IPV6 INTERFACE command.

The INTERFACE parameter specifies the physical interface to set the properties of.

The IPADDRESS parameter specifies the IPv6 address to associate with this interface, in IPv6 notation. A prefix length must be specified, using either slash notation (e.g. 3FFE::0/16) or the PLEN parameter. If both are used, the two prefix values must be the same.

The PLEN parameter is the length of the IPv6 prefix for this interface. This number represents the number of contiguous bits that represent the network address of the network to which the interface is attached. If slash notation is used on the IPADDRESS parameter, the PLEN parameter is not required. If the slash notation is not used, then the PLEN parameter is required.

The FILTER parameter is the number of the filter that is to be applied to this interface.

The PREFERRED parameter specifies the time in seconds for which this IPv6 address will be the preferred address for this interface. The default is INFINITE. The value of this parameter cannot be greater than that of the VALID parameter.

The PRIORITYFILTER parameter is the number of the priority filter attached to this interface.

The VALID parameter is the time in seconds that the IPv6 address exists on the interface. After the time specified expires, the IPv6 address is deleted. The default is INFINITE. The time value must be the same as or greater than that of the PREFERRED parameter.

Examples To set the interface address *3ffe::1* to be the preferred address for 300 seconds, use the command:

```
SET IPV6 INTERFACE=VLAN2 IPADDRESS=3FFE::1 PREF=300
```

See Also ADD IPV6 INTERFACE
CREATE IPV6 INTERFACE
DELETE IPV6 INTERFACE
DESTROY IPV6 INTERFACE
SHOW IPV6
SHOW IPV6 INTERFACE

SHOW IPV6

Syntax SHOW IPV6

Description This command displays information on the settings for the various parts of the IPv6 module.

Examples To show the general settings for the IPv6 module, use the command:

```
SHOW IPV6
```

Figure 8: Example output from the SHOW IPV6 command.

```

IPV6 Module Configuration
-----

Module Status ..... Enabled
IPV6 Packet Forwarding ..... Enabled
IPV6 RIP ..... Disabled
IPV6 Echo Reply ..... Enabled
Name Server ..... 202.49.72.50
Secondary Name Server ..... Not Set
Source-Routed Packets ..... Discarded

Routing Protocols

RIP Neighbours ..... 0

Active Routes:

Static ..... 5
Interface ..... 1
Neighbour Discovery..... 0
RIP ..... 0
Other ..... 0
-----
Total Number of routes..... 6
Discarded routes ..... 0
    
```

Table 6: Parameters displayed in the output of the SHOW IPV6 command.

Parameter	Meaning
Module Status	Whether or not the IPv6 module has been enabled; one of " Enabled" or " Disabled" .
IPV6 Packet Forwarding	Whether or not the router is currently capable of forwarding packets; one of " Enabled" or " Disabled" .
IPV6 RIP	Whether or not the router has been configured to send and receive RIPv6 packets; one of " Enabled" or " Disabled" .
IPV6 Echo Reply	Whether or not the router has been configured to reply to echo request (ping) packets; one of " Enabled" or " Disabled" .
Source-Routed Packets	What the router has been configured to do with source routed packets.
RIP Neighbours	The number of nodes on the network known by the router to be running RIPv6.
Static	The number of static routes that have been added to the router through manual configuration.
Interface	The number of routes that have been generated through an interface being added.
Neighbour Discovery	The number of routes gathered from neighbour discovery.
RIP	The number of routes gathered from RIPv6 packets.
Other	The number of routes gathered through other protocols, for example, OSPF.
Discarded Routes	The number of routes that have been discarded due to finding a better route through neighbour discovery. Static routes are not discarded.

SHOW IPV6 COUNTER

Syntax SHOW IPV6 COUNTER

Description This command uses the COUNTER parameter to display the IPv6 counters.

Examples To show the IPv6 counters, use the command:

```
SHOW IPV6 COUNTER
```

Figure 9: Example output from the SHOW IPV6 COUNTER command.

```

IPV6 MIB Counters
-----

Interface Counters

Interface: vlan1
  InReceives ..... 0          OutForwDatagrams ..... 5
  InNoRoutes ..... 0          OutRequests ..... 5
  InDiscards ..... 0          OutDiscards ..... 0
  InAddrErrors ..... 0        OutFragOKs ..... 0
  InUnknownProtos ..... 0     OutFragFails ..... 0
  InTruncatedPkts ..... 0     OutFragCreates ..... 0
  InMcastPkts ..... 0         OutMcastPkts ..... 0
  ReasmReqds ..... 0          ReasmOKs ..... 0
  ReasmFails ..... 0
  InDelivers ..... 0
  InHdrErrors ..... 0
  InTooBigErrors ..... 0

ICMP counters

  inMsgs ..... 0              OutMsgs ..... 5
  InErrors ..... 0            OutErrors ..... 0
  InDestUnreachs ..... 0     OutDestUnreachs ..... 0
  InAdminProhibs ..... 0     OutAdminProhibs ..... 0
  InTimeExcds ..... 0        OutTimeExcds ..... 0
  InParmProblems ..... 0     OutParmProblems ..... 0
  InPktTooBigs ..... 0       OutPktTooBigs ..... 0
  InEchos ..... 0             OutEchos ..... 5
  InEchoReplies ..... 0     OutEchoReplies ..... 0
  InRouterSolicits ..... 0   OutRouterSolicits ..... 0
  InRouterAdvert ..... 0     OutRouterAdvert ..... 0
  InNeighborSolicits ..... 0 OutNeighborSolicits ..... 0
  InNeighborAdvert ..... 0   OutNeighborAdvert ..... 0
  InRedirects ..... 0        OutRedirects ..... 0
  InGroupMembQueries ..... 0  OutGroupMembQueries ..... 0
  InGroupMembResp ..... 0    OutGroupMembResp ..... 0
  InGroupMembReduct ..... 0  OutGroupMembReduct ..... 0

```

Table 7: Parameters displayed in the output of the SHOW IPV6 COUNTER command.

Parameter	Meaning
InReceives	The number of packets received.
InNoRoutes	The number of input packets discarded because no route could be found to transmit them to their destination.
InDiscards	The number of discarded packets.
InAddrErrors	The number of packets received with invalid addresses.
InUnknownProtos	The number of packets received with unknown next headers.
InTruncatedPkts	The number of truncated packets received.
InMcastPkts	The number of multicast packets received.
ReasmReqds	The number of packets requiring reassembling received.
ReasmFails	The number of packets reassembly has failed for.
InDelivers	The number of packets that have successfully been delivered.
InHdrErrors	The number of packets received with invalid headers.
InTooBigErrors	The number of packets that have been received and discarded because they were too big.
OutForwDatagrams	The number of packets that have been forwarded out.
OutRequests	The number of echo requests sent.
OutDiscards	The number of packet discarded messages sent.
OutFragOKs	The number of fragmentation success messages sent.
OutFragFails	The number of fragmentation failed messages sent.
OutFragCreates	The number of output packet fragments that have been generated as a result of fragmentation at this output interface.
OutMcastPkts	The number of multicast packets sent.
ReasmOKs	The number of IPv6 packets successfully reassembled. Note that this counter is incremented at the interface to which these packets were addressed, which will not necessarily be the input interface for some of the fragments.
inMsgs	The total number of ICMP messages received by the interface which includes all those counted by InErrors. Note that this interface is the interface to which the ICMP messages were addressed. This may not necessarily be the input interface for the messages.
InErrors	The number of ICMP messages which the interface received but determined as having ICMP-specific errors (bad ICMP checksums, bad length etc).
InDestUnreachs	The number of ICMP Destination Unreachable messages received by the interface.
InAdminProhibs	The number of ICMP Destination Unreachable/Communication Administratively Prohibited messages received by the interface.
InTimeExcds	The number of ICMP Time Exceeded messages received by the interface.

Table 7: Parameters displayed in the output of the SHOW IPV6 COUNTER command. (Continued)

InParmProblems	The number of ICMP Parameter Problem messages received by the interface.
InPktTooBigs	The number of ICMP Packet Too Big messages received by the interface.
InEchos	The number of ICMP Echo (request) messages received by the interface.
InEchoReplies	The number of ICMP Echo Reply messages received by the interface.
InRouterSolicits	The number of ICMP Router Solicit messages received by the interface.
InRouterAdvert	The number of ICMP Router Advertisement messages received by the interface.
InNeighborSolicits	The number of ICMP Neighbor Solicitation messages received by the interface.
InNeighbor Advert	The number of ICMP Neighbor Advertisement messages received by the interface.
InRedirects	The number of Redirect messages received by the interface.
InGroupMembQueries	The number of ICMPv6 Group Membership Query messages received by the interface.
InGroupMembResp	The number of ICMPv6 Group Membership Response messages received by the interface.
InGroupMembReduct	The number of ICMPv6 Group Membership Reduction messages received by the interface.
OutMsgs	The total number of ICMP messages which the interface attempted to send. Note that this counter includes all those counted by OutErrors.
OutErrors	The number of ICMP messages which this interface did not send due to problems discovered within ICMP, such as a lack of buffers. This value does not include errors discovered outside the ICMP layer, such as the inability of IPv6 to route the resultant packet.
OutDestUnreachs	The number of ICMP Destination Unreachable messages sent by the interface.
OutAdminProhibs	The number of ICMP destination Unreachable/Communication Administratively Prohibited messages sent.
OutTimeExcds	The number of ICMP Time Exceeded messages sent by the interface.
OutParmProblems	The number of ICMP Parameter Problem messages sent by the interface.
OutPktTooBigs	The number of ICMP Packet Too Big messages sent by the interface.
OutEchos	The number of ICMP Echo (request) messages sent by the interface.
OutEchoREplies	The number of ICMP Echo Reply messages sent by the interface.
OutRouterSolicits	The number of ICMP Router Solicit messages sent by the interface.

Table 7: Parameters displayed in the output of the SHOW IPV6 COUNTER command. (Continued)

OutRouterAdvert	The number of ICMP Router Advertisement messages sent by the interface.
OutNeighborSolicits	The number of ICMP Neighbor Solicit messages sent by the interface.
OutNeighborAdvert	The number of ICMP Router Advertisement messages sent by the interface.
OutRedirects	The number of Redirect messages sent by the interface. For a host, this object will always be zero, since hosts do not send redirects.
OutGroupMembQueries	The number of ICMPv6 Group Membership Query messages sent.
OutGroupMembResp	The number of ICMPv6 Group Membership Response messages sent.
OutGroupMembReduct	The number of ICMPv6 Group Membership Reduction messages sent.

SHOW IPV6 FILTER

Syntax `SHOW IPV6 FILTER[=filter-number]`

where:

- *filter-number* is a number in the range 0 to 299.

Description This command displays information about filters. If a filter is specified, the entries in the filter are displayed. If a filter is not specified, the entries in all filters are displayed (see Figure 10).

Figure 10: Example output from the SHOW IPv6 FILTER command.

```

IP Filters
-----
No.   Ent.   Source Address                               /Plen
      Source Port
      Destination Address                       /Plen
      Destination Port
      Size                                     Prot. (C/T)
      Options                                 Session
      Logging
      Matches                                 Act/Pri
-----
1     1     FE80:0000:0000:0000:0260:97FF:FE8F:64AA   /16
      Any
      FE80:0000:0000:0000:0324:96BB:FE8F:64AA /32
      Any
      Any                                     Any
      No                                     Any
      None
      0                                       Exclude
-----
      2     FE80:0000:0000:0000:0260:97FF:FE8F:63CC   /16
      Any
      ::                                       /16
      Any
      Any                                     ICMP Any Any
      No                                     Any
      Header
      155                                    Include
      Passes: 0                            Fails: 636
-----
2     1     FE80:0000:0000:0000:0333:97FF:FE8F:64AA   /32
      Any
      FE80:0000:0000:0000:0444:96BB:FE8F:64AA /32
      Any
      Any                                     ICMP Any Any
      No                                     Any
      Header
      132                                    Exclude
      Passes: 0                            Fails: 0

```

Table 8: Parameters displayed in the output of the SHOW IPv6 FILTER command.

Parameter	Meaning
Source Address	The source IPv6 address.
Source Port	The source IPv6 port.
/Plen	The number of leftmost contiguous bits of the address to match against.
Destination Address	The destination IPv6 address.
Destination Port	The destination IPv6 port.
Size	The size of the packet to match against.
Prot. (C/T)	The protocol to match against. If the protocol is ICMP, the ICMP type and code will be shown.
Options	Whether either the 'Hop by Hop' or 'destination Option' extension headers are present.

Table 8: Parameters displayed in the output of the SHOW IPV6 FILTER command. (Continued)

Parameter	Meaning
Session	The status of the TCP session establishment process.
Logging	Log details of the packet.
Matches	The number of packets that matched the entry.
Act/Pri	The action to take and priority to give if packets match the entry.

Examples To show all filters, use the command:

```
SHOW IPV6 FILTER
```

To show all entries in *filter 2*, use the command:

```
SHOW IPV6 FILTER=2
```

See Also ADD IPV6 FILTER
DELETE IPV6 FILTER
SET IPV6 FILTER

SHOW IPV6 HOST

Syntax SHOW IPV6 HOST

Description This command uses the HOST parameter to display information on the host names associated with IPv6 addresses on the router.

Examples To show the host names associated with the IPv6 addresses on the router, use the command:

```
SHOW IPV6 HOST
```

Figure 11: Example output from the SHOW IPV6 HOST command.

IPv6 Address	Host Name
3ffe::0002	foobar
3ffe::0004	mainserver
3ffe::0006	bob

Table 9: Parameters displayed in the output of the SHOW IPV6 HOST command.

Parameter	Meaning
IPv6 Address	The IPv6 address of the host.
Host Name	The alias that has been assigned to that host.

SHOW IPV6 INTERFACE

Syntax SHOW IPV6 INTERFACE[=*interface*]

where:

- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. vlan0). Valid interface types are VLAN, PPP and VIRT. VIRT is the interface type of a tunnel.

Description This command uses the INTERFACE parameter to display information on the various interfaces configured with IPv6. If an interface is specified, information on the router's routes is displayed.

Examples To display information on the vlan0 interface, use the command:

```
SHOW IPV6 INTERFACE=VLAN0
```

Figure 12: Example output from the SHOW IPV6 INTERFACE command.

```
IPV6 Interface Configuration
-----
Physical Interface ..... loopback
Ipv6 Interface Index ..... N/A
Link-layer address ..... N/A
EUI-64 Interface Identifier ..... N/A
True MTU/Link MTU ..... 1500/1500
Multicast status ..... Enabled
Send Router Advertisements? ..... No
Ipv6 Interface Addresses:
Int Ipv6 Interface Addresses          plen valid/pref      Sc/St/E/
-----
  0 ::0001                            infinite/infinite loop/pref/Y
-----

IPV6 Interface Configuration
-----
Physical Interface ..... virt0
Ipv6 Interface Index ..... 1
Link-layer address ..... ipv4 tunnel
EUI-64 Interface Identifier ..... ipv4 tunnel
True MTU/Link MTU ..... 1280/576
Multicast status ..... Enabled
Send Router Advertisements? ..... No
filter..... 6
priFilter..... 206
Ipv6 Interface Addresses:
Int Ipv6 Interface Addresses          plen valid/pref      Sc/St/E/
-----
  1 4ffe::0001                        /128infinite/infinite global/pref/Y
-----
```

Table 10: Parameters displayed in the output of the SHOW IPV6 INTERFACE command.

Parameter	Meaning
Physical Interface	The name of the physical interface.
IPv6 Interface Index	The index of the interface in the IPv6 interface table.
Link-layer address	The link layer address of the interface.
EUI-64 Interface Identifier	Interface identifier in IEEE EUI-64 format.
True MTU/Link MTU	The Maximum Transmission Unit of the interface. The " True MTU" is the true link MTU of the interface. The " Link MTU" is the MTU set by the user.
Multicast Status	Whether or not receiving of multicast packets is enabled on the interface.
Send Router Advertisements?	Whether or not the interface will send router advertisements.
Int	Index in the IPv6 address table.
IPv6 Interface Address	The IPv6 address(es) configured on the router.
plen	The prefix length of the address(es) on the interface.
valid/pref	The values of the valid and preferred timers.
Sc/St/E	The current scope (Sc), which indicates the extent of the address; one of " link" , " site" or " global" . The status (St) of the address according to duplicate address detection; one of " pref" (preferred) if the address is unique or " depr" (deprecated) if the address is a duplicate or has timed out. Whether or not the address is enabled (E); one of " Y" or " N" .

SHOW IPV6 MULTICAST

Syntax SHOW IPV6 MULTICAST

Description This command uses the MULTICAST parameter to display information on the multicast settings.

Examples To show the multicast settings, use the command:

```
SHOW IPV6 MULTICAST
```

Figure 13: Example output from the SHOW IPV6 MULTICAST command.

```

Ipv6 Multicast Memberships:
Multicast Address                Interface
-----
ff02::0001:ff01:f9f5             vlan0
ff02::0002                       vlan0
ff02::0001                       vlan0
ff02::0001:ff00:0006             vlan0
-----

```

Table 11: Parameters displayed in the output of the SHOW IPV6 MULTICAST command.

Parameter	Meaning
Multicast Address	The address of the multicast group.
Interface	The interface that belongs to the multicast group.

SHOW IPV6 ND

Syntax SHOW IPV6 ND

Description This command displays information gathered through neighbour discovery.

Examples To show information gathered through neighbour discovery, use the command:

```
SHOW IPV6 ND
```

Figure 14: Example output from SHOW IPV6 ND command.

```

Ipv6 Neighbour Cache:
Ipv6 Address          Link-layer address
Interface      State      LastReachble IsRouter
-----
3ffe:0c00:8017:0120:0220:35ff:feb1:7065 00-20-35-b1-70-65
vlan1          stale      0 msec      no
fe80::0220:35ff:feb1:7065 00-20-35-b1-70-65
vlan1          stale      0 msec      no
-----

```

Table 12: Parameters displayed in the output of the SHOW IPV6 ND command.

Parameter	Meaning
IPv6 Address	The IPv6 address of the neighbour.
Link-layer address	The link-layer address of the neighbour.
Interface	The name of the interface the neighbour is located on.F
State	The state of the neighbour discovery entry; one of "reachable", "unreachable" and "stale" .
LastReachable	The number of milliseconds the neighbour was last reachable in.
IsRouter	Whether or not the neighbour is a router.

See Also DISABLE IPV6
ENABLE IPV6

SHOW IPV6 RIP

Syntax `SHOW IPV6 RIP [COUNTER|TIMER]`

Description This command displays information about RIP interfaces.

The COUNTER parameter displays information on RIP counters.

The TIMER parameter displays information on the RIP timers.

Example To display a list of RIP interfaces, use the command:

```
SHOW IPV6 RIP
```

Figure 15: Example output from the SHOW IPV6 RIP command.

Interface	Ipv6 Address
virt0	3ffe::0002

Table 13: Parameters displayed in the output of the SHOW IPV6 RIP command.

Parameter	Meaning
Interface	The logical interface that RIP is running over.
IPv6 Address	The IPv6 address of that interface.

To display a list of RIP timers, use the command:

```
SHOW IPV6 RIP TIMER
```

Figure 16: Example output from the SHOW IPV6 RIP TIMER command.

RIPng route timers:					
Destination	int.	met.	val	hold	flush
2fff::/32	1	2	18	0	28
2ffa::/32	1	2	18	0	28
2ffb::/32	1	2	18	0	28

Table 16-1: Parameters displayed in the output of the SHOW IPV6 RIP TIMER command.

Parameter	Meaning
Destination	The destination network of the route.
Int	The logical interface that RIP is running over.
Met	The RIP metric associated with this route.
Val	The valid lifetime of the route, in seconds.
Hold	The time interval (in seconds) after a route has become invalid, during which the router will ignore updates for the route which would normally make the route valid again.
Flush	The time interval (in seconds), from the last update of a route, until the route is flushed from the route table.

To display a list of RIP counters, use the command:

```
SHOW IPV6 RIP COUNTER
```

Figure 17: Example output from the SHOW IPV6 RIP COUNTER command.

```

-----
IPV6 RIPng Counter Summary:
  Input:
    inResponses ..... 3
    inDiscards ..... 0
  Output:
    outResponses..... 0
-----

```

Table 14: Parameters displayed in the output of the SHOW IPV6 RIP COUNTER command.

Parameter	Meaning
inResponses	The number of RIP responses received.
inDiscards	The number of RIP packets discarded.
outResponses	The number of RIP responses sent.

See Also ADD IPV6 RIP
 DELETE IPV6 RIP
 DISABLE IPV6 RIP
 ENABLE IPV6 RIP
 SHOW IPV6

SHOW IPV6 ROUTE

Syntax SHOW IPV6 ROUTE

Description This command uses the ROUTE parameter to display information on the routes that the router has.

Examples To show the routes a router has, use the command:

```
SHOW IPV6 ROUTE
```

Figure 18: Example output from the SHOW IPV6 ROUTE command.

```

Destination prefix  --->  Next Hop
Int.      Age  Policy Protocol      Metric  Pref   Tunnel  Flags
-----
2ffe::/32 ---> 3ffe::0002
virt0     no   0      static           2       60     yes     -
2fff::/32 ---> 3ffe::0001
virt0     yes  0      ripng            2       100    yes     -
2ffa::/32 ---> 3ffe::0001
virt0     yes  0      ripng            2       100    yes     -
2ffb::/32 ---> 3ffe::0001
virt0     yes  0      ripng            2       100    yes     -
-----
Codes: P=publish, D=default, A=addrconf, S=stale, L=onlink
N=nonexthop, C=cache, F=flow, U=unknown

```

Table 15: Parameters displayed in the output of the SHOW IPV6 ROUTE command.

Parameter	Meaning
Destination prefix	The destination network of the route.
Next Hop	The next node along the path to the destination.
Int	The interface that packets will be sent out.
Age	Whether or not the route will age out.
Policy	Reserved for future use.
Protocol	The protocol the route was gathered from. " Static" indicates that the route was manually entered onto the router. " Interface" indicates that the route was automatically created through adding an interface.
Metric	The RIP metric that is being used. This usually indicates how far away the destination is, in hops.
Pref	The preference of the route over others to the same destination.
Tunnel	Whether or not the interface is a tunnel.
Flags	The various flags that have been set or implied on the route.

SHOW IPV6 TIMER

Syntax SHOW IPV6 TIMER

Description This command uses the TIMER parameter to show information on the internal timers the IPv6 module is keeping.

Examples To show the internal timers on the module, use the command:

```
SHOW IPV6 TIMER
```

Figure 19: Example output from the SHOW TIMER command.

Timer Name	Time Left
Adv. Router Advertisement(1)	0 secs
Adv. Router Advertisement(2)	0 secs
ND Wait Queue	1 secs

Table 16: Parameters displayed in the output of the SHOW IPV6 TIMER command.

Parameter	Meaning
Timer Name	The name of the timer.
Time Left	The amount of time left until the timer finishes.

SHOW IPV6 TUNNEL

Syntax SHOW IPV6 TUNNEL

Description This command uses the TUNNEL parameter to display information on the tunnels that are configured on the router.

Examples To show the tunnels configured on the router, use the command:

```
SHOW IPV6 TUNNEL
```

Figure 20: Example output from the SHOW IPV6 TUNNEL command.

Ipv6 Tunnels:		ipv6 Interface
Ipv6 Tunnel Address	Tunnel start	Tunnel end
1 4ffe::0001	192.168.1.2	192.168.1.1

Table 17: Parameters displayed in the output of the SHOW IPV6 TUNNEL command.

Parameter	Meaning
IPv6 Tunnel Address	The IPv6 address of the tunnel.
Tunnel Start	The IPv4 address of the start of the tunnel.
Tunnel End	The IPv4 address of the end of the tunnel.