AX Series
IPv6 Configuration Guide

for
the
Guaranteed
Network

Edition 2

Preface

This guide is an IPv6 network deployment guide for helping engineers easily configure IPv6-based systems.

This guide also describes settings for the AX series from ALAXALA Networks Corporation and how to configure various types of servers so that the minimum requirements for an IPv6 system can be met.

Notes on using this guide

This guide offers information about basic operability and connectivity as confirmed by ALAXALA Networks Corporation under specific conditions and does not guarantee the validity of the Switch functionality, performance, and reliability in every environment. Use this guide as a general guideline for setting up systems supported by ALAXALA Networks Corporation products.

Export restrictions

If you export this guide, you must check and comply with all applicable laws, rules and restrictions of Japan and any other countries, such as Japan's Foreign Exchange and Foreign Trade Law and U.S. export control laws and regulations.

Conventions: The terms "Switch" and "switch"

The term Switch (upper-case "S") is an abbreviation for any or all of the following models:

- AX6700S series switch
- AX6600S series switch
- AX3630S series switch
- AX3640S series switch

The term switch (lower-case "s") might refer to a Switch, another type of switch from the current vendor, or a switch from another vendor. The context decides the meaning.

Trademarks

- Ethernet is a trade name of Xerox Corporation in the United States.
- Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other countries.
- Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
- FreeBSD is a registered trademark of The FreeBSD Project.
- BIND is a registered trademark of Internet Systems Consortium, Inc.
- Apache is a registered trademark of The Apache Software Foundation.
- Qpopper is a registered trademark of QUALCOMM Incorporated.
- Other company and product names in this manual are trademarks or registered trademarks of their respective owners.

Software versions used in this manual

- AX6700S ver.11.3.A
- AX6600S ver.11.3.A
- AX3630S ver.11.2.B
- AX3640S ver.11.2.B
- Windows Vista
- FreeBSD 6.3
- BIND ver.9.4.2
- Apache ver.2.2.8
- Postfix ver.2.4.6
- Qpopper ver.4.0.9

Revision history

Edition	Rev.	Date	Description	Applicable sections
Edition 1	1	April 14, 2008	First edition	--
Edition 2	0	May 19, 2010	A description for AX6600S series switches has been added.	2.1, 2.3
			Errors regarding supported functionality for AX3600S series switches have been corrected. (The policy routing functionality has been removed.)	2.3 2.2
			A description for AX3640S series switches has been added.	3.2.1

Contents

1. IPv6 Features

1.1. Features

This chapter describes the features of IPv6.

(1) 128-bit address space

IPv6 has a huge address space.

The following is a comparison between the numbers of addresses in IPv6 and in IPv4:

$$2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$$

$$2^{32} = 4,294,967,296$$

This clearly shows how big the address space of IPv6 is.

(2) Automatic address generation

IPv6 terminals can automatically generate IPv6 addresses. The 64-bit prefix (subnet) part is advertised from the router, and the 64-bit host address part can be generated from the MAC address or randomly created.

(3) Use of NDP (Neighbor Discovery Protocol)

ICMP now uses NDP, instead of ARP, for its functionality. NDP is used when the MAC address is resolved from an IPv6 address or when a router or switch advertises the IPv6 prefix part.

(4) Address allocation that reduces the routing table size

The IPv4 routing table size has been growing on the Internet, and the number of routing tables is ever increasing. This causes consumption of a large amount of resources, including router memory.

Based on the lessons learned from IPv4 address allocation, IPv6 address blocks are allocated to each Regional Internet Registry (such as APNIC), and then the Regional Internet Registries redistribute those address blocks to National Internet Registries (such as JPNIC). National Internet Registries further redistribute addresses to individual ISPs, which then assign IPv6 addresses to their contracted end-users.

This allocation system can assign the same address blocks to the same regions and thus aggregate address routes, resulting in a reduction in the number of full routes.

1.2. IPv6 address types

IPv6 addresses are classified into three address types: unicast, anycast, and multicast addresses. Broadcast addresses are no longer used in IPv6.

This chapter explains the unicast and multicast addresses supported by AX series products.

1.2.1. Unicast address

Several types of addresses are defined as unicast addresses. This subsection explains commonly used global, link-local, and loopback addresses.

(1) Global address

An IPv6 global address is an address where the first three bits of the address prefix are `001`. IPv6 global addresses are globally unique and used for communication over the Internet. A packet originating from an IPv6 global address is transferred according to the routing information. The following figure shows the structure of an IPv6 global address.

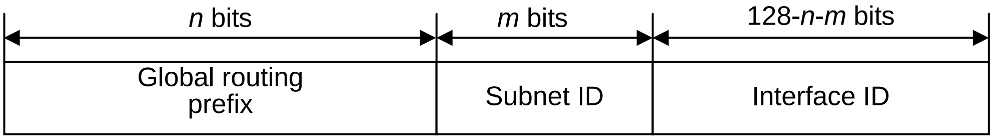


Figure 1.2-1 IPv6 global address

(2) Link-local address

An IPv6 link-local address is an address composed of the first 64 bits of the address prefix `fe80::` and the 64-bit interface ID part. An IPv6 link-local address is only valid within a single link (subnet) and used when no automatic addressing settings, NDP, or routers exist. The following figure shows the structure of an IPv6 link-local address.

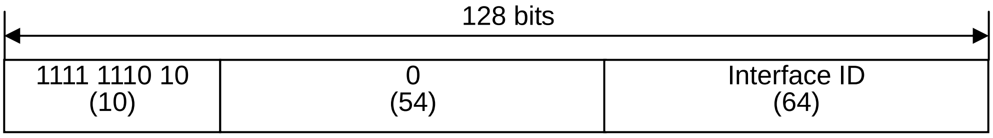


Figure 1.2-2 Link-local address

(3) Loopback address

The address 0:0:0:0:0:0:0:1 (::1) is defined as the loopback address. The loopback address is used as the destination address when packets are sent to their originating node. You cannot assign the loopback address to an interface. Also, an IPv6 packet with its destination address set to the loopback address is not allowed to be sent to any device other than the originating node or to be routed by routers. The following figure shows the loopback address.

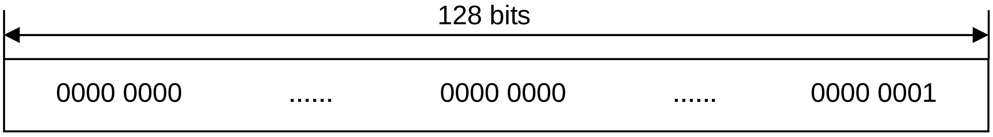


Figure 1.2-3 Loopback address

1.2.2. Multicast address

A multicast address is an identifier for a group of nodes. The first eight bits of the multicast address format prefix are `ff`. A node can belong to multiple multicast groups. You cannot use a multicast address as the source address of a packet. A multicast address has the address format prefix followed by the flags field (4 bits), the scope field (4 bits), and the group ID field (112 bits). The following figure shows the structure of an IPv6 multicast address.

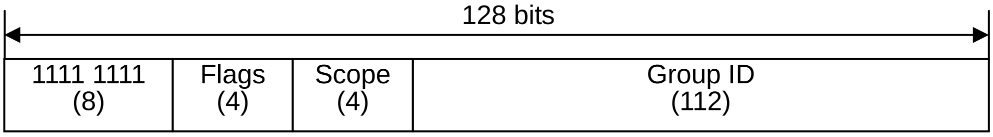


Figure 1.2-4 Multicast address

When multicast packets are sent, the first 16 bits of the destination MAC address are set to `33:33` and the remaining 32 bits are set to the last 32 bits of the multicast address.

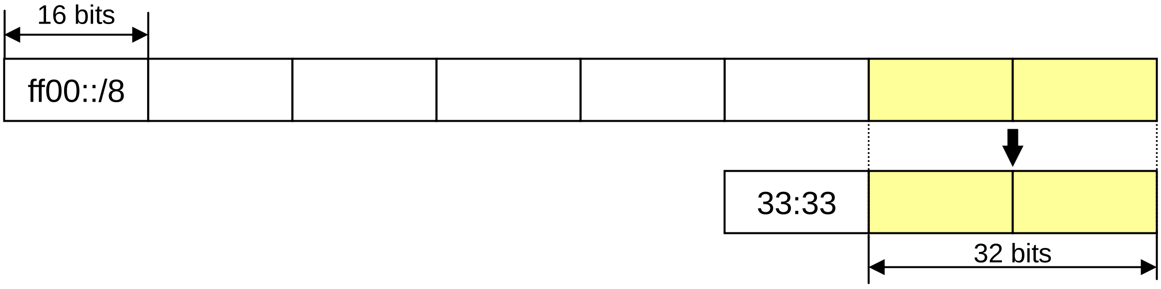


Figure 1.2-5 Multicast destination MAC address

1.3. Address format

An IPv6 address has a 128-bit space. The IPv6 address format is described below.

(1) An IPv6 address is represented by 16-bit hexadecimal values separated by colons (:).

Example: 2001:0db8:0811:ff02:0000:08ff:fe8b:3090

(2) Leading zeroes within a 16-bit segment separated by a colon can be omitted.

Example: 2001:db8:811:ff02:0:8ff:fe8b:3090
↑ ↑ ↑ ↑ These arrows indicate omitted zeroes.

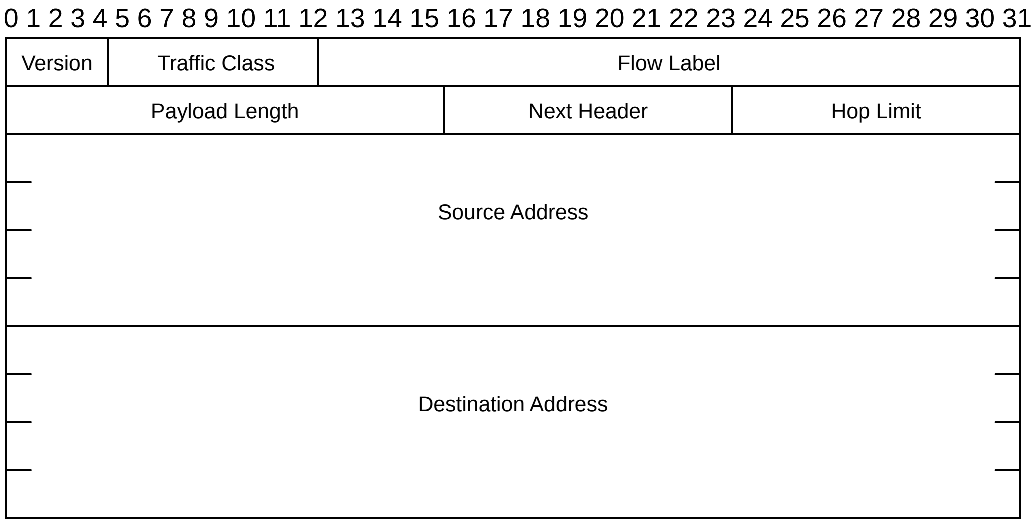
(3) Consecutive zeroes can be replaced by a double colon (::). Note, however, that :: can only appear once in an address.

Example: Replacing zeroes within an IPv6 address:
2001:0000:0000:1234:0000:0000:0000:3090 → 2001:0:0:1234::3090
2001::1234:0:0:0:3090

The following conversion is invalid because multiple double colons are used:
2001:0000:0000:1234:0000:0000:0000:3090 → Invalid = 2001::1234::3090 (This is not allowed.)

1.4. IPv6 header format

The IPv6 header format is shown below.



- Version (4 bits) IP version (always set to 6)
- Traffic Class (8 bits) Used for specifying and identifying the class and priority
- Flow Label (20 bits) Flow number to which the packet belongs
- Payload Length (16 bits) Payload length in octets
- Next Header (8 bits) Type of the header immediately following the IPv6 header
- Hop Limit (8 bits) Hop limit
- Source Address (128 bits) Source address of the packet
- Destination Address (128 bits) Destination address of the packet

Figure 1.4-1 IPv6 header format

1.5. NDP

NDP uses four ICMP packet types to, for example, distribute the prefix part for automatic address generation and to resolve the MAC address from an IPv6 address.

1.5.1. Router solicitation (RS)

IPv6 terminals send RS messages to routers to ask for router advertisements (RA). For example, when a terminal starts up, it can send an RS and receive an RA for automatic address generation and for IPv6 address assignment. Also, the sender of the received RA can be registered as the default gateway.

1.5.2. Router advertisement (RA)

RA messages are periodically sent by each router. When a terminal that has not automatically generated addresses receives an RA, the terminal uses the RA prefix to automatically generate IPv6 addresses.

1.5.3. Neighbor solicitation (NS)

IPv6 devices send NS messages when resolving MAC addresses from IPv6 addresses. (This functionality is the successor of ARP for IPv4.) The target IPv6 device sends back an NA response, which enables resolution of the MAC address from the IPv6 address.

Also, a functionality called Neighbor Unreachability Detection (NUD) has been added in IPv6. This functionality confirms that an IPv6 device is reachable. If the device is not reachable, its NDP entry is deleted.

1.5.4. Neighbor advertisement (NA)

IPv6 terminals send NA messages to respond to NS messages.

1.6. Automatic address generation

IPv6 terminals automatically generate IPv6 addresses.
The 64-bit prefix (subnet) part is advertised by RA.
The 64-bit host address part is automatically generated from the MAC address. When the host address part is generated from a MAC address, a numbering system called EUI-64 (Extended Unique Identifier-64) is used to generate a unique 64-bit value.

The following example shows the logic used by an IPv6 terminal to automatically generate IPv6 addresses.

(1) Prefix value of RA sent from the router: 2001:db8:2:3::/64

(2) Terminal MAC address: 00:12:e2:08:64:01
The host address part 0212:e2ff:fe08:6401 is generated according to EUI-64. (See ***Figure 1.6-1***.)

(3) Based on the above, the IPv6 addresses of this terminal will be:

Global address:	2001:db8:2:3:212:e2ff:fe08:6401
Link-local address:	fe80::212:e2ff:fe08:6401

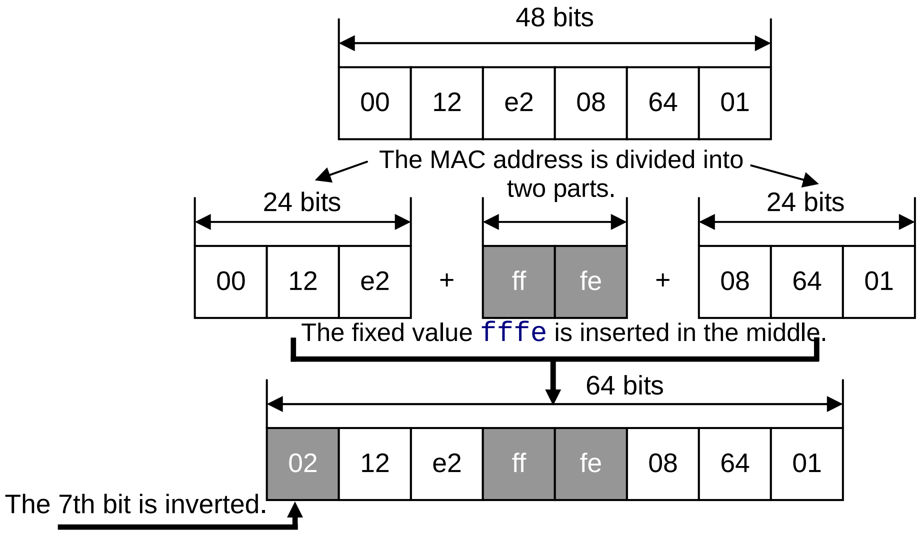


Figure 1.6-1 EUI-64

2. IPv6 Support in AX Series Switches

This chapter explains supported IPv6 functionality and the capacity limit of each model in the AX series.

2.1. AX6700S, AX6600S, and AX6300S series

In the AX6700S, AX6600S, and AX6300S series, the capacity limit varies depending on the types of modules that contain the ASIC hardware where packets are processed. Two types of modules are available for each of the series:

AX6700S	**BSU-LA** and **BSU-LB**
AX6600S	**CSU-1A** and **CSU-1B**
AX6300S	**MSU-1A** and **MSU-1B**

You can change the capacity limit by changing the allocation pattern of each module.

To change the allocation pattern, use the `fwdm prefer` command. The change is applied when the BSU or MSU restarts.

Table 2.1-1 Capacity limits and allocation patterns for the BSU-LA (AX6700S), CSU-1A (AX6600S), and MSU-1A (AX6300S)

Allocation patterns	Number of configured entries						
	IPv4 unicast active path	IPv4 multicast path	IPv6 unicast active path	IPv6 multicast path	MAC address	ARP	NDP
`default`	32768	4000	16384	1000	24576	12288	12288
`ipv4-uni`	65536	0	0	0	24576	12288	0
`ipv4-ipv6-uni`	32768	0	32768	0	24576	12288	12288
`vlan`	8192	0	8192	0	49152	8192	8192

Table 2.1-2 Capacity limits and allocation patterns for the BSU-LB (AX6700S), CSU-1B (AX6600S), and MSU-1B (AX6300S)

Allocation patterns	Number of configured entries						
	IPv4 unicast active path	IPv4 multicast path	IPv6 unicast active path	IPv6 multicast path	MAC address	ARP	NDP
`default`	65536	8000	32768	8000	65536	24576	24576
`ipv4-uni`	212992	0	0	0	24576	24576	0
`ipv4-ipv6-uni`	106496	0	106496	0	24576	24576	24576
`vlan`	8192	0	8192	0	122880	8192	8192

2.2. AX3600S series

By default, AX3600S series switches do not reserve IPv6 resources in the hardware table. To reserve resources for IPv6 entries, you must use the swrt_table_resource configuration command to set the allocation pattern to l3switch-2 or l3switch-3.
l3switch-3 is the IPv6 unicast priority mode available only for AX3640S.

Table 2.2-1 Capacity limits and allocation patterns for AX3600S

Item		Pattern		
		l3switch-1	l3switch-2	l3switch-3[#1]
IPv4	Unicast path	12288	8192	1024
	Multicast path	1024	256	16
	ARP	3072 5120[#2]	1024	128
IPv6	Unicast path	0	2048	5632
	Multicast path	0	128	16
	NDP	0	1024	1024

#1: Only available for AX3640S.

#2: Capacity limit for AX3640S

2.3. Supported IPv6 functionality

The following table lists the IPv6 functionality supported by AX series switches.

Table 2.3-1 IPv6 functionality supported by AX6300S, AX6600S, and AX6700S series switches

Category	Functionality
Layer 2 functionality	MLDv1/v2 snooping
Layer 3 functionality	Static routing, RIPng, OSPFv3, BGP4+ (optional) VRRP PIM-SM, PIM-SSM, MLD ver1, MLD ver2
Additional functionality	Filtering, QoS, IPv6 DHCP server (Prefix Delegation), multipaths (load balancing), policy routing
Network management	SNMP, IPv6 MIBs, VRRP (IPv6 MIB), Syslog
Operation and maintenance	ICMPv6, telnet (server/client), SSH (ver.1/ver.2/server/client), ftp (server/client), tftp, uRPF

Table 2.3-2 IPv6 functionality supported by AX3600S series switches

Category	Functionality
Layer 2 functionality	MLDv1/v2 snooping
Layer 3 functionality	Static routing, RIPng, OSPFv3, BGP4+ (optional) VRRP PIM-SM, PIM-SSM, MLD ver1, MLD ver2
Additional functionality	Filtering, QoS, IPv6 DHCP server (Prefix Delegation), multipaths (load balancing)
Network management	SNMP, IPv6 MIBs, VRRP (IPv6 MIB), Syslog
Operation and maintenance	ICMPv6, telnet (server/client), SSH (ver.1/ver.2/server/client), ftp (server/client), tftp, uRPF

Table 2.3-3 IPv6 functionality supported by AX2400S series switches

Category	Functionality
Layer 2 functionality	MLDv1/v2 snooping
Additional functionality	Filtering, QoS
Network management	SNMP, IPv6 MIBs, Syslog
Operation and maintenance	ICMPv6, telnet (server/client), SSH (ver.1/ver.2/server/client), ftp (server/client), tftp

3. Network Configuration

3.1. Network diagram

The following figure shows an IPv6 network configuration example.

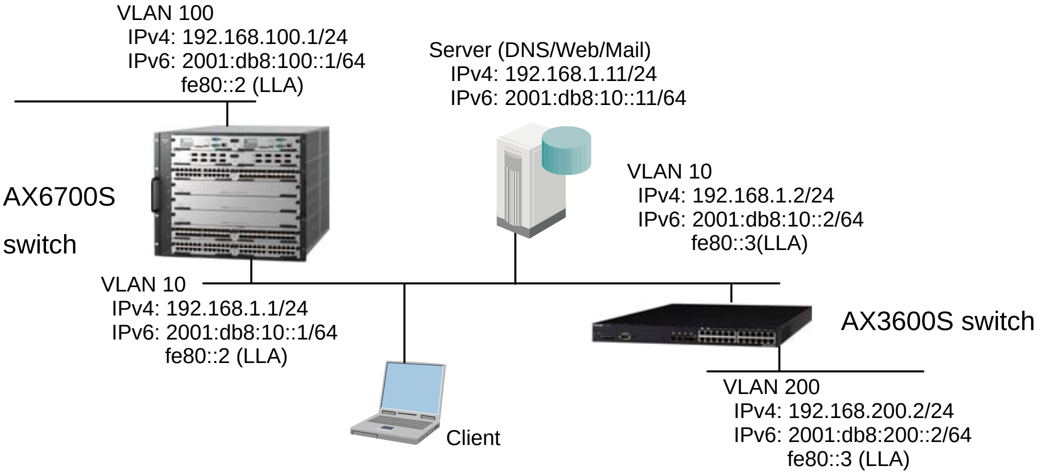


Figure 3.1-1 Network diagram

3.2. IPv6 settings

3.2.1. AX series switch settings

This subsection explains how to configure IPv6 for AX series switches.

(1) Reserving IPv6 table resources [only for AX3600S]

By default, AX3600S series switches do not reserve IPv6 table entries in the hardware table (default setting: `l3switch-1`). Therefore, you must specify to secure resources for IPv6 in the hardware table. AX6300S and AX6700S series switches have IPv6 resources reserved in the hardware table by default. Thus, you do not need to specify this setting. Simply select an allocation pattern suitable for your purpose.

Use the `swrt_table_resource` command to set `l3switch-2`. **You need to restart the device after executing this command.** Restart the device to apply the table pattern.

For AX3640S series switches, you can specify the *IPv6 unicast priority mode* by specifying `l3switch-3`.

Table 3.2-1 Reserving IPv6 table resources

<table>
<tr><th colspan="5">Reserving IPv6 table resources [only for AX3600S]</th></tr>
<tr><td rowspan="7">IPv4 or IPv6 mode
(config)# swrt_table_resource l3switch-2
IPv6 unicast priority mode (available only for AX3640S)
(config)# swrt_table_resource l3switch-3</td><td colspan="2">Allocation patterns</td><td>l3switch-2</td><td>l3switch-3</td></tr>
<tr><td rowspan="3">IPv4</td><td>Unicast</td><td>8192</td><td>1024</td></tr>
<tr><td>Multicast</td><td>256</td><td>16</td></tr>
<tr><td>ARP</td><td>1024</td><td>128</td></tr>
<tr><td rowspan="3">IPv6</td><td>Unicast</td><td>2048</td><td>5632</td></tr>
<tr><td>Multicast</td><td>128</td><td>16</td></tr>
<tr><td>NDP</td><td>1024</td><td>1024</td></tr>
</table>

(2) Specifying an IPv6 address

Set the IPv6 address of an interface (VLAN).

Configuring and enabling both IPv4 and IPv6 at the same time is called **dual stacking**. With dual stacking, the different IPv4 and IPv6 protocols work independently and concurrently.

You can choose whether to specify link-local addresses (LLAs). When LLAs are not specified, EUI-64 is used to automatically generate them.

See ***Table 3.2-2*** for a setup example, in which the same LLA (fe80::2) is specified for VLAN 10 and VLAN 100.

This is not allowed in IPv4 networks. Even in IPv6, a single global address is not assigned to multiple interfaces. This is only possible for LLAs.

An LLA is an address only valid within a single subnet. (See *1.2*.) This means that you can assign the same address to multiple subnets as long as the address is unique within each subnet. (For details, see Chapter *5*.)

Specifying `ipv6 enable` is required. Without this setting, IPv6 does not work.

Table 3.2−2 IPv6 address settings

IPv6 address settings for AX6700S	

`(config)# interface vlan 10`
`(config-if)# ip address 192.168.1.1 255.255.255.0`
`(config-if)# ipv6 address 2001:db8:10::1 /64`
`(config-if)# ipv6 address fe80::2 link-local`
`(config-if)# ipv6 enable`
	Specify addresses to the VLAN 10 interface.
	Specify an IPv4 address.
	Specify an IPv6 address.
	Specify an IPv6 LLA. (Optional. When this setting is omitted, EUI-64 is used.)
	Enable IPv6.

`(config)# interface vlan 100`
`(config-if)# ip address 192.168.100.1 255.255.255.0`
`(config-if)# ipv6 address 2001:db8:100::1/64`
`(config-if)# ipv6 address fe80::2 link-local`
`(config-if)# ipv6 enable`
	Specify addresses for the VLAN 100 interface.
	Specify the same address as for vlan 10.
	Make sure that the same LLA is used as for vlan 10.
	Specify an IPv6 LLA. (Optional. When this setting is omitted, EUI-64 is used.)
	Enable IPv6.

IPv6 address settings for AX3600S	

`(config)# interface vlan 10`
`(config-if)# ip address 192.168.1.2 255.255.255.0`
`(config-if)# ipv6 address 2001:db8:10::2/64`
`(config-if)# ipv6 address fe80::3 link-local`
`(config-if)# ipv6 enable`
	Specify addresses for the VLAN 10 interface.
	Specify an IPv4 address.
	Specify an IPv6 address.
	Specify an IPv6 LLA. (Optional. When this setting is omitted, EUI-64 is used.)
	Enable IPv6.

`(config)# interface vlan 200`
`(config-if)# ip address 192.168.200.1 255.255.255.0`
`(config-if)# ipv6 address 2001:db8:200::2/64`
`(config-if)# ipv6 address fe80::3 link-local`
`(config-if)# ipv6 enable`
	Specify addresses for the VLAN 100 interface.
	Specify the same address as for vlan 10.
	Make sure that the same LLA is used as for vlan 10.
	Specify an IPv6 LLA. (Optional. When this setting is omitted, EUI-64 is used.)
	Enable IPv6.

(3) OSPFv3 settings

Configure OSPFv3. IPv6 uses OSPF version 3 (denoted as OSPFv3).

Table 3.2–3 OSPFv3 settings

OSPFv3 settings for AX6700S	

`(config)# ipv6 router ospf 1`
`(config-rtr)# router-id 2.2.2.2`
`(config-rtr)# passive-interface vlan 100`
	Configure OSPFv3 (where 1 is the domain number).
	Specify a router ID (required).
	Use this command to configure the interface as a passive interface (optional).

`(config)# interface vlan 10`
`(config-if)# ipv6 ospf 1 area 0`
`(config-if)# ipv6 ospf cost 120`
	Configure OSPFv3 for the interface vlan 10.
	Enable OSPFv3. Specify domain 1 and area 0.
	Specify the cost setting for the interface.

`(config)# interface vlan 100`
`(config-if)# ipv6 ospf 1 area 0`
`(config-if)# ipv6 ospf cost 10`
	Configure OSPFv3 for the interface vlan 100.
	Enable OSPFv3. Specify domain 1 and area 0.
	Specify the cost setting for the interface.

OSPFv3 settings for AX3600S	

`(config)# ipv6 router ospf 1`
`(config-rtr)# router-id 3.3.3.3`
`(config-rtr)# passive-interface vlan 200`
	Configure OSPFv3 (where 1 is the domain number).
	Specify a router ID (required).
	Use this command to configure the interface as a passive interface (optional).

`(config)# interface vlan 10`
`(config-if)# ipv6 ospf 1 area 0`
`(config-if)# ipv6 ospf cost 120`
	Configure OSPFv3 for the interface vlan 10.
	Enable OSPFv3. Specify domain 1 and area 0.
	Specify the cost setting for the interface.

`(config)# interface vlan 200`
`(config-if)# ipv6 ospf 1 area 0`
`(config-if)# ipv6 ospf cost 10`
	Configure OSPFv3 for the interface vlan 200.
	Enable OSPFv3. Specify domain 1 and area 0.
	Specify the cost setting for the interface.

(4) RIPng settings

Table 3.2–4 RIPng settings

RIPng settings for AX6700S	

`(config)# ipv6 router rip`
`(config-rtr-rip)#`
	Activate RIPng.

`(config)# interface vlan 10`
`(config-if)# ipv6 rip enable`
	Configure RIPng for the interface vlan 10.
	Enable RIPng.

`(config)# interface vlan 100`
`(config-if)# ipv6 rip enable`
	Configure RIPng for the interface vlan 100.
	Enable RIPng.

RIPng settings for AX3600S	

`(config)# ipv6 router rip`
`(config-rtr-rip)#`
	Activate RIPng.

`(config)# interface vlan 10`
`(config-if)# ipv6 rip enable`
	Configure RIPng for the interface vlan 10.
	Enable RIPng.

`(config)# interface vlan 200`
`(config-if)# ipv6 rip enable`
	Configure RIPng for the interface vlan 200.
	Enable RIPng.

(5) Static route settings

Table 3.2–5 Static route settings

IPv6 static route settings	
`(config)# ipv6 route 2001:db8:4::/64 fe80::100 vlan 10`	Specify static route settings. Specify fe80::10 as the next hop to 2001:db8:4::/64. Specify `vlan 10`. This is required because you must indicate the interface when using an LLA to specify the next hop.

(6) DHCPv6 settings

IPv6 terminals can automatically generate IPv6 addresses, during which the network part is obtained from the received RA and the host address part is generated by using EUI-64 or other methods. However, DNS addresses cannot be mapped.

You can use a protocol called DHCPv6 to allocate DNS addresses to IPv6 terminals by configuring DHCPv6 on AX series switches.

With DHCPv6 configured, ALAXALA Networks Corporation has confirmed that in Windows Vista, IPv6 addresses are automatically generated and DHCPv6 servers (AX series switches) can obtain DNS server addresses.

Table 3.2–6 DHCPv6 settings for DNS server address allocation

DHCPv6 settings for DNS server address allocation	

`(config)# ipv6 dhcp pool POOL1`
`(config-dhcp)# dns-server 2001:db8:10::11`
`(config-if)# exit`
`(config)# interface vlan 10`
`(config-if)# ipv6 address 2001:db8:10::1 /64`
`(config-if)# ipv6 enable`
`(config-if)# ipv6 nd other-config-flag`
`(config-if)# ipv6 dhcp server POOL1`
	Configure IPv6 DHCP pool information (where the pool name is `POOL1`).
	Specify a DNS server address to be allocated.
	Switch to the interface where DHCPv6 is running.
	Specify `other configuration` for RA#.
	Specify a pool name.

#: The `other configuration` flag is now set for RA packets. If the flag is set, the terminal automatically obtains information for *non-IPv6 addresses* by using methods other than RA. In this example, this setting is specified so that the DNS server address can be automatically obtained via non-RA methods. More specifically, the DHCPv6 protocol is used to allocate DNS server addresses. For details on the `other configuration` flag, see RFC 4861 *Neighbor Discovery for IP version 6 (IPv6)*.

3.2.2. Terminal settings

(1) Windows Vista

In Windows Vista, the IPv6 protocol is installed by default. When the operating system receives an RA message, it uses the prefix in the message to automatically generate IPv6 addresses. For the host address part, random addresses are generated.

(2) FreeBSD

In FreeBSD, you are asked whether to enable IPv6 during installation. Selecting `enable` specifies the settings shown in ***Table 3.2-7***. When the operating system receives an RA message from a router or a switch, it uses the prefix in the message to automatically generate IPv6 addresses, based on EUI-64. The sender of the received RA is set as the default route. If the operating system receives RA messages from multiple routers or switches, the sender of the first RA is set as the default route.

Table 3.2-7 IPv6 settings for FreeBSD

File name: `/etc/rc.conf`
`ipv6_enable="YES"`

To configure static IPv6 addresses, instead of automatically generated addresses, use the commands shown below.

These settings are applied when the operating system restarts.

Table 3.2-8 Static IPv6 address settings for FreeBSD

File name: `/etc/rc.conf`	
`ipv6_enable="YES"`	Enable IPv6.
`ipv6_ifconfig_rl0="2001:db8:10::11 prefixlen 64"`	Specify a static IPv6 address for `rl0` (NIC).
`ipv6_defaultrouter="2001:db8:10::1"`	Specify a static IPv6 default route.

Specify DNS server settings in `/etc/resolv.conf`.

Table 3.2-9 DNS server settings for FreeBSD

File name: `/etc/resolv.conf`	
`domain      example.co.jp`	Specify the name of the domain to which the device belongs.
`nameserver  2001:db8:10::1`	Specify the IPv6 address of the DNS server.
`nameserver  192.168.1.11`	Specify the IPv4 address of the DNS server.

The following table shows commands that are helpful when FreeBSD uses IPv6.

Table 3.2–10 FreeBSD tips

Command	Description
`# ifconfig`	Checks the interface (NIC) address.
`# ifconfig rl0 inet6 2001:db8:10::8/64`	Specifies an IPv6 address for `rl0` (NIC).

[File name]
`/etc/start_if.rl0`
[Entry]
`ifconfig rl0 inet6 fe80::8 prefixlen 64 alias`
	If you want to explicitly specify an LLA, instead of using EUI-64 for automatic generation of an LLA, specify an entry in the file, as shown on the left column. This setting is applied when the operating system restarts.

`# netstat –rn`	Looks up the routing table. (Both IPv4 and IPv6 are displayed.)
`# route add –inet6 default 2001:db8:10::1`	Specifies an IPv6 default route.
`# route delete –inet6 default`	Deletes the IPv6 default route.
`# ndp –P`	Deletes IPv6 addresses. (This is available when addresses are set to be automatically generated.)
`# ndp –R`	Deletes the IPv6 default route. (This is available when addresses are set to be automatically generated.)
`# rtsol rl0`	Sends an RS message. A router or switch that receives an RS message sends back an RA. When FreeBSD receives the RA, it automatically generates addresses.
`# dhcp rl0`	Asks the DHCP server for addresses in IPv4. (For reference purposes)

4. Server Configuration

4.1. DNS server configuration

4.1.1. BIND - FreeBSD

This subsection explains how to configure one of the most popular DNS server applications, BIND, in FreeBSD.

(1) Installing BIND

BIND is installed on FreeBSD by default. Specify and restart the operating system, as shown below. The setting is applied when the operating system restarts.

Table 4.1-1 BIND setting

File name: `/etc/rc.conf`

```
named_enable="YES"
```

(2) Configuring the files

You must configure the seven types of files shown below.

As a file naming policy, this guide uses the file name extension `.zone` for forward lookup files[#1] and `.rev` for reverse lookup files[#2].

(2.1) Control file (named.conf)

This is a file referred to by the BIND program `named` during startup. Specify forward and reverse lookup files in this file. Use the file name `named.conf`. Do not use other file names.

Place `named.conf` in the directory `/etc/namedb` unless you need to do otherwise.

(2.2) Forward lookup file for IPv4 and IPv6 addresses

This is a file for mapping host names to IPv4 and IPv6 addresses.

(2.3) Reverse lookup file for IPv4 addresses

This is a file for mapping IPv4 address to host names.

(2.4) Reverse lookup file for IPv6 addresses

This is a file for mapping IPv6 addresses to host names.

(2.5) Forward lookup file for the IPv4 and IPv6 local host

This is a file for forward lookup of the IPv4 local host address 127.0.0.1 and the IPv6 local host address `::1`.

(2.6) Reverse lookup file for the IPv4 local host

This is a file for reverse lookup of the IPv4 local host address (1.0.0.127.in-arpa.).

(2.7) Reverse lookup file for the IPv6 local host

This is a file for reverse lookup of the IPv6 local host address.

#1: *Forward lookup* uses a domain name to find an IP address.

#2: *Reverse lookup* uses an IP address to find a domain name.

You must configure the files shown below.

Table 4.1−2 Control file

File name: `/etc/namedb/named.conf`			
1	`options {`	1	
2	`       directory "/etc/namedb";`	2	Specify a directory where the other setting files are placed.
3	`       pid-file "/var/run/named/pid";`	3	Specify a file where PID is stored.
4	`       listen-on-v6{`	4	
5	`              any;`	5	
6	`       };`	6	
7	`};`	7	
8		8	
9	`zone "example.co.jp" {`	9	Configure forward lookup for the domain `example.co.jp`.
10	`       type master;`	10	
11	`       file "example.co.jp.zone";`	11	Specify a file name.
12	`};`	12	
13		13	
14	`zone "1.168.192.in-addr.arpa" {`	14	Configure reverse lookup for IPv4 192.168.1.
15	`         type master;`	15	
16	`         file "example.co.jp.rev";`	16	Specify a file name.
17	`};`	17	
18		18	
19	`zone "0.0.0.0.0.1.0.0.8.b.d.0.1.0.0.2.ip6.arpa" {`	19	Configure reverse lookup for IPv6 2001:db8:10:0:0.
20	`    type master;`	20	
21	`    file "example.co.jp.ipv6.rev";`	21	Specify a file name.
22	`};`	22	
23		23	
24	`zone "localhost" {`	24	Configure forward lookup for the IPv4 and IPv6 local host.
25	`        type master;`	25	
26	`        file "localhost.zone";`	26	Specify a file name.
27	`};`	27	
28		28	
29	`zone "0.0.127.in-addr.arpa" {`	29	Configure reverse lookup for the IPv4 local host.
30	`         type master;`	30	
31	`         file "localhost.rev";`	31	Specify a file name.
32	`};`	32	
33		33	
34	`zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa"`	34	Configure reverse lookup for the IPv6 local host.
35	`{`	35	
36	`        type master;`	36	
37	`        file "localhost.ipv6.rev";`	37	Specify a file name.
	`};`		

Table 4.1−**3 Forward lookup file for IPv4 and IPv6 addresses**

File name: /etc/namedb/example.co.jp.zone			
1	`$TTL   86400`	1	
2	`@     IN   SOA   ns.example.co.jp. root.example.co.jp. (`	2	
3	`                     2007102601    ; Serial`	3	
4	`                     3600         ; Refresh 1hr`	4	
5	`                     900          ; Retry 15min`	5	
6	`                     604800        ; Expire 1w`	6	
7	`                     86400  )      ; Minimum 24hr`	7	
8		8	
9	`$ORIGIN example.co.jp.`	9	
10	`       IN     NS     ns.`	10	Specify the NS record setting.
11	`       IN     MX 10   ns.`	11	Specify the MX record setting.
12	`mono   IN     A      192.168.1.1`	12	From this line downward, configure the forward lookup database for mapping host names to IPv4 and IPv6 addresses.
13	`       IN     AAAA    2001:db8:10::1`	13	
14	`di      IN     A      192.168.1.2`	14	
15	`       IN     AAAA    2001:db8:10::2`	15	
16	`tri      IN     A      192.168.1.3`	16	Specify the A record setting.
17	`       IN     AAAA    2001:db8:10::3`	17	Specify the AAAA record setting.
18	`tetra    IN     A      192.168.1.4`	18	
19	`       IN     AAAA    2001:db8:10::4`	19	
20	`penta   IN     A      192.168.1.5`	20	
21	`       IN     AAAA    2001:db8:10::5`	21	
22	`hexa    IN     A      192.168.1.6`	22	
23	`       IN     AAAA    2001:db8:10::6`	23	
24	`hepta   IN     A      192.168.1.7`	24	
25	`       IN     AAAA    2001:db8:10::7`	25	
26	`octa    IN     A      192.168.1.8`	26	
27	`       IN     AAAA    2001:db8:10::8`	27	
28	`nona    IN     A      192.168.1.9`	28	
29	`       IN     AAAA    2001:db8:10::9`	29	
30	`deca    IN     A      192.168.1.10`	30	
31	`       IN     AAAA    2001:db8:10::10`	31	
32	`ns      IN     A      192.168.1.11`	32	
33	`       IN     AAAA    2001:db8:10::11`	33	
34	`www     IN     CNAME   ns`	34	Specify the CNAME record setting.

Table 4.1−**4 Reverse lookup file for IPv4 addresses**

File name: /etc/namedb/example.co.jp.rev			
1	`$TTL   86400`	1	
2	`@     IN   SOA   ns.example.co.jp. root.example.co.jp. (`	2	
3	`                     2007102601    ; Serial`	3	
4	`                     3600         ; Refresh 1hr`	4	
5	`                     900          ; Retry 15min`	5	
6	`                     604800        ; Expire 1w`	6	
7	`                     86400  )      ; Minimum 24hr`	7	
8		8	
9	`         IN NS    ns.`	9	
10	`1        IN PTR   mono.`	10	From this line downward, configure the reverse lookup database for the IPv4 subnet 192.168.1.. The name of the host 192.168.1.3 is tri.example.co.jp.
11	`2        IN PTR   di.`	11	
12	`3        IN PTR   tri.`	12	
13	`4        IN PTR   tetra.`	13	
14	`5        IN PTR   penta.`	14	
15	`6        IN PTR   hexa.`	15	
16	`7        IN PTR   hepta.`	16	
17	`8        IN PTR   octa.`	17	
18	`9        IN PTR   nona.`	18	
19	`10       IN PTR   deca.`	19	
20	`11       IN PTR   ns.`	20	

Table 4.1−5 Reverse lookup file for IPv6 addresses

File name:	`/etc/namedb/example.co.jp.ipv6.rev`		
1	`$TTL   86400`	1	
2	`@      IN   SOA   ns.example.co.jp. root.example.co.jp. (`	2	
3	`                    2007102601    ; Serial`	3	
4	`                    3600         ; Refresh 1hr`	4	
5	`                    900          ; Retry 15min`	5	
6	`                    604800        ; Expire 1w`	6	
7	`                    86400  )      ; Minimum 24hr`	7	
8		8	
9	`                                IN NS  ns.example.co.jp.`	9	
10	`1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR mono.example.co.jp.`	10	From this line downward, configure the reverse lookup file database for the IPv6 subnet 2001:db8:10::. The name of the host 2001:db8:10::3 is tri.example.co.jp.
11	`2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR di.example.co.jp.`	11	
12	`3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR tri.example.co.jp.`	12	
13	`4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR tetra.example.co.jp.`	13	
14	`5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR penta.example.co.jp.`	14	
15	`6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR hexa.example.co.jp.`	15	
16	`7.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR hepta.example.co.jp.`	16	
17	`8.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR octa.example.co.jp.`	17	
18	`9.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR nona.example.co.jp.`	18	
19	`0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR deca.example.co.jp.`	19	
20	`1.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0  IN PTR ns.example.co.jp.`	20	

Table 4.1−6 Forward lookup file for the IPv4 and IPv6 local host

File name:	`/etc/namedb/localhost.zone`		
1	`$TTL   86400`	1	
2	`@     IN   SOA   ns.example.co.jp. root.example.co.jp. (`	2	
3	`                     2007102601   ; Serial`	3	
4	`                     3600         ; Refresh 1hr`	4	
5	`                     900          ; Retry 15min`	5	
6	`                     604800       ; Expire 1w`	6	
7	`                     86400  )     ; Minimum 24hr`	7	
8		8	
9	`      IN     NS     ns.example.co.jp.`	9	
10	`      IN     A      127.0.0.1`	10	Specify the IPv4 local host.
11	`      IN     AAAA    ::1`	11	Specify the IPv6 local host.

Table 4.1−7 Reverse lookup file for the IPv4 local host

File name:	`/etc/namedb/localhost.rev`		
1	`$TTL   86400`	1	
2	`@     IN   SOA   ns.example.co.jp. root.example.co.jp. (`	2	
3	`                     2007102601   ; Serial`	3	
4	`                     3600         ; Refresh 1hr`	4	
5	`                     900          ; Retry 15min`	5	
6	`                     604800       ; Expire 1w`	6	
7	`                     86400  )     ; Minimum 24hr`	7	
8		8	
9	`      IN     NS     localhost.`	9	
10	`1     IN     PTR    localhost.`	10	Configure reverse lookup for 127.0.0.1.

Table 4.1-8 Reverse lookup file for the IPv6 local host

File name: /etc/namedb/localhost.ipv6.rev

```
$TTL   86400
@      IN   SOA   ns.example.co.jp. root.example.co.jp. (
                       2007102601   ; Serial
                       3600         ; Refresh 1hr
                       900          ; Retry 15min
                       604800       ; Expire 1w
                       86400  )     ; Minimum 24hr

        IN      NS      localhost.
1       IN      PTR     localhost.
```

Configure reverse lookup for ::1.

(3) Starting the BIND program (named)

Execute the following command as a user with root permissions:

```
    # sh /etc/rc.d/named start
```

This starts the BIND program, and the settings files are automatically loaded.

To confirm that the program has started, execute the following command:

```
    # ps –ax | grep named
```

The following is an example display:

```
80526  ??  Ss     0:00.30 /etc/sbin/named
80528  p0  RL+    0:00.01 grep named
```

When named is displayed as shown in the first line (80526), the program is running normally. Note that the number (80526 in this example) varies every time the program starts.

(4) Testing forward and reverse lookup

To check that DNS settings work properly, use the dig command of FreeBSD to test forward and reverse lookup.

(4.1) Specifying a DNS server address

Specify a DNS server address in the file /etc/resolv.conf so that FreeBSD runs as a DNS client. The example below specifies the local host address because the Switch itself runs as a DNS server. The address specified first within the file is set to the primary DNS server. Therefore, specify an IPv6 address first so that the IPv6 protocol is used to ask the DNS server for addresses.

Table 4.1-9 DNS server settings for FreeBSD

File name: /etc/resolv.conf	
`domain        example.co.jp`	Specify a domain name.
`nameserver     ::1`	Specify an IPv6 DNS server. (IPv6 local host)
`nameserver     127.0.0.1`	Specify an IPv4 DNS server. (IPv4 local host)

(4.1) Forward lookup of IPv4 addresses

The following table shows how to find an IPv4 address from a host name.

Table 4.1-10 Forward lookup of an IPv4 address

Using the dig command for forward lookup of an IPv4 address	
# dig –t A octa.example.co.jp ; <<>> DiG 9.3.4-P1 <<>> -t A octa.example.co.jp ;; global options: printcmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38809 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0 ;; QUESTION SECTION: ;octa.example.co.jp. IN A ;; ANSWER SECTION:	Execute the dig command. Use the A record for IPv4 forward lookup. Ask for the IPv4 address of octa.example.co.jp.
octa.example.co.jp. 86400 IN A 192.168.1.8	192.168.1.8 is returned.
;; AUTHORITY SECTION: example.co.jp. 86400 IN NS ns. ;; Query time: 0 msec ;; SERVER: ::1#53(::1) ;; WHEN: Mon Mar 31 20:07:20 2008 ;; MSG SIZE rcvd: 68	

(4.2) Forward lookup of IPv6 addresses

The following table shows how to find an IPv6 address from a host name.

Table 4.1-11 Forward lookup of an IPv6 address

Using the dig command for forward lookup of an IPv6 address	
# dig –t AAAA octa.example.co.jp ; <<>> DiG 9.3.4-P1 <<>> -t AAAA octa.example.co.jp ;; global options: printcmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18675 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0 ;; QUESTION SECTION: ;octa.example.co.jp. IN AAAA ;; ANSWER SECTION:	Execute the dig command. Use the AAAA record for IPv6 forward lookup. Ask for the IPv6 address of octa.example.co.jp.
octa.example.co.jp. 86400 IN AAAA 2001:db8:10::8	2001:db8:10::8 is returned.
;; AUTHORITY SECTION: example.co.jp. 86400 IN NS ns. ;; Query time: 0 msec ;; SERVER: ::1#53(::1) ;; WHEN: Mon Mar 31 20:07:33 2008 ;; MSG SIZE rcvd: 80	

(4.3) Reverse lookup of IPv4 addresses

The following table shows how to find a host name from an IPv4 address.

Table 4.1‒12 Reverse lookup of an IPv4 address

Using the dig command for reverse lookup of an IPv4 address	

`# dig –x 192.168.1.8`
 	
`; <<>> DiG 9.3.4-P1 <<>> -x 192.168.1.8`
`;; global options:  printcmd`
`;; Got answer:`
`;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12910`
`;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0`
 	
`;; QUESTION SECTION:`
`;8.1.168.192.in-addr.arpa.        IN  PTR`
	Execute the `dig` command.
	Ask for the host name of the IP address 192.168.1.8.

`;; ANSWER SECTION:`
`8.1.168.192.in-addr.arpa.  86400  IN  PTR  octa.`
	`octa` is returned.

`;; AUTHORITY SECTION:`
`1.168.192.in-addr.arpa.    86400  IN  NS   ns.`
 	
`;; Query time: 0 msec`
`;; SERVER: ::1#53(::1)`
`;; WHEN: Mon Mar 31 20:08:02 2008`
`;; MSG SIZE  rcvd: 76`	

(4.4) Reverse lookup of IPv6 addresses

The following table shows how to find a host name from an IPv6 address.

Table 4.1–13 Reverse lookup of an IPv6 address

Using the dig command for reverse lookup of an IPv6 address

```
# dig –x 2001:db8:10::8

; <<>> DiG 9.3.4-P1 <<>> -x 2001:db8:10::8
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60061
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; QUESTION SECTION:
;8.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.8.b.d.0.1.0.0.2.ip6.arpa. IN PTR

;; ANSWER SECTION:
8.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
  86400 IN  PTR  octa.example.co.jp.

;; AUTHORITY SECTION:
0.0.0.0.0.1.0.0.8.b.d.0.1.0.0.2.ip6.arpa. 86400     IN NS ns.example.co.jp.

;; ADDITIONAL SECTION:
ns.example.co.jp. 86400    IN      A       192.168.1.11
ns.example.co.jp. 86400    IN      AAAA    2001:db8:10::11

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Mon Mar 31 20:08:18 2008
;; MSG SIZE  rcvd: 183
```

Execute the `dig` command.

`octa.example.co.jp` is returned.

(5) Troubleshooting

There are cases when the program `named` cannot start due to various reasons. Also, a warning might be displayed even when the program has successfully started.

In such cases, check the following for log entries to be used for debugging.

Log file: `/var/log/messages`

Log entries are added at the end of the file. You can see the most recent log entries by using the `tail` command, which displays the end part of the file.

```
    # tail /var/log/messages
```

(6) Updating the database

After you add, delete, or modify hosts or addresses in a file, increase the `Serial` value at the top of the file. If a secondary DNS server exists, it compares the Serial number of its own file and that of the corresponding file of the primary DNS server. If the Serial number of the primary DNS server file is larger, the secondary DNS server obtains the primary DNS server file.

After you modify any setting, execute the following command to make sure that the relevant file is reloaded:

```
# kill -HUP `cat /var/run/named/pid`
```

4.2. Web server configuration

4.2.1. Apache - FreeBSD

This subsection explains how to configure one of the most popular HTTP server applications, Apache, in FreeBSD.

The description in this subsection is based on Apache 2.2.8 (the most recent version as of February 1, 2008)#.

#: See the Apache website: http://www.apache.org/

(1) Installing Apache

In this subsection, `ports` is used to install the program. Obtain the latest `ports.tar.gz` on the FreeBSD website#, and extract and then install the file.

#: ports: ftp://ftp.freebsd.org/pub/FreeBSD/ports/ports/ports.tar.gz

Table 4.2–1 Installing Apache

Installing Apache	

As a root user, execute the following commands:
 	
`# cd /usr/ports`
`# gzip –d ports.tar.gz`
`# tar xvf ports.tar`
`# cd /usr/ports/www/apache22`
`# make install`
	(Obtain `ports.tar.gz` on the website beforehand.)
	Change the directory.
	Decompress the file.
	Extract the file.
	Change the directory.
	Install Apache.
	This compiles the program, which takes some time.

The following is the installed program:

```
/usr/local/sbin/httpd
```

(2) Setting up the configuration file (httpd.conf)

During installation of the program, the configuration file is saved. Edit the file as required.

Configuration file: `/usr/local/etc/apache22/httpd.conf`

Table 4.2-2 Control file http.conf

File name: /usr/local/etc/apache22/httpd.conf	
# ServerRoot: The top of the directory tree under which the server's # configuration, error, and log files are kept. # # Do not add a slash at the end of the directory path. If you point # ServerRoot at a non-local disk, be sure to point the LockFile directive # at a local disk. If you wish to share the same ServerRoot for multiple # httpd daemons, you will need to change at least LockFile and PidFile. # **ServerRoot "/usr/local"**	
# Listen: Allows you to bind Apache to specific IP addresses and/or # ports, instead of the default. See also the <VirtualHost> # directive. # # Change this to Listen on specific IP addresses as shown below to # prevent Apache from glomming onto all bound IP addresses. # #Listen 12.34.56.78:80 **Listen 80**	httpd runs via TCP port 80.
<IfModule !mpm_netware_module> # # If you wish httpd to run as a different user or group, you must run # httpd as root initially and it will switch. # # User/Group: The name (or #number) of the user/group to run httpd as. # It is usually good practice to create a dedicated user and group for # running httpd, as with most system services. # **User www** **Group www** </IfModule>	
# ServerAdmin: Your address, where problems with the server should be # e-mailed. This address appears on some server-generated pages, such # as error documents. e.g. admin@your-domain.com # **ServerAdmin robbie.robertson@example.co.jp**	Administrator email address
# DocumentRoot: The directory out of which you will serve your # documents. By default, all requests are taken from this directory, but # symbolic links and aliases may be used to point to other locations. # **DocumentRoot "/usr/local/www/apache22/data"**	Specify the document root. Actual content is placed under this directory.

File name: /usr/local/etc/apache22/httpd.conf

```
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
   DirectoryIndex index.html
</IfModule>

# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog /var/log/httpd-error.log
```

Specify the name of the file in which error Log entries are recorded.

(3) Starting the Apache program (httpd)

Execute the following command as a user with root permissions:

```
     # /usr/local/sbin/apachectl start
```

The Apache program starts and runs as an IPv4 and IPv6 HTTP server.

4.3. Configuring the mail server

This section explains how to configure a mail server that uses Postfix for the SMTP server and Qpopper for the POP3 daemon.

4.3.1. Postfix - FreeBSD

This subsection explains how to configure the SMTP server software Postfix in FreeBSD.
The description in this subsection is based on Postfix 2.4.6 (the most recent version as of February 1, 2008)[#].

[#]: See the Postfix website: http://www.postfix.org/

(1) Installing Postfix

In this subsection, `ports` is used to install the program.

Table 4.3-1 Installing Postfix

Installing Postfix	

As a root user, execute the following commands:
`# cd /usr/ports/mail/postfix`
`# make install`
	(Obtain `ports.tar.gz` on the website beforehand.)
	Change the directory.
	Install Postfix.
	This compiles the program, which takes some time.

The following is the installed program:

```
/usr/local/sbin/postfix
```

(2) Setting up the configuration file (main.cf)

Edit the configuration file of Postfix.

Configuration file: /usr/local/etc/postfix/main.cf

The red letters in the following table indicate modifications of and additions to the default settings in main.cf.

Table 4.3‒2 Control file main.cf

File name: /usr/local/etc/postfix/main.cf

```
# Global Postfix configuration file. This file lists only a subset
# of all parameters. For the syntax, and for a complete parameter
# list, see the postconf(5) manual page (command: "man 5 postconf").
#
# For common configuration examples, see BASIC_CONFIGURATION_README
# and STANDARD_CONFIGURATION_README. To find these documents, use
# the command "postconf html_directory readme_directory", or go to
# http://www.postfix.org/.
#
# For best results, change no more than 2-3 parameters at a time,
# and test if Postfix still works after every change.
```

(Omitted)

```
# INTERNET HOST AND DOMAIN NAMES
#
# The myhostname parameter specifies the internet hostname of this
# mail system. The default is to use the fully-qualified domain name
# from gethostname(). $myhostname is used as a default value for many
# other configuration parameters.
#
#myhostname = host.domain.tld
#myhostname = virtual.domain.tld
myhostname = ns.example.co.jp
```

Your host (mail server) name

```
# The mydomain parameter specifies the local internet domain name.
# The default is to use $myhostname minus the first component.
# $mydomain is used as a default value for many other configuration
# parameters.
#
#mydomain = domain.tld
mydomain = example.co.jp
```

Your domain name

```
# SENDING MAIL
#
# The myorigin parameter specifies the domain that locally-posted
# mail appears to come from. The default is to append $myhostname,
# which is fine for small sites.  If you run a domain with multiple
# machines, you should (1) change this to $mydomain and (2) set up
# a domain-wide alias database that aliases each user to
# user@that.users.mailhost.
#
# For the sake of consistency between sender and recipient addresses,
# myorigin also specifies the default domain name that is appended
# to recipient addresses that have no @domain part.
#
```

File name: /usr/local/etc/postfix/main.cf

```
#myorigin = $myhostname
myorigin = $mydomain
```

When email is sent from the local host, the domain name is appended after @ in the source email address.

```
# RECEIVING MAIL

# The inet_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on.  By default,
# the software claims all active interfaces on the machine. The
# parameter also controls delivery of mail to user@[ip.address].
#
# See also the proxy_interfaces parameter, for network addresses that
# are forwarded to us via a proxy or network address translator.
#
# Note: you need to stop/start Postfix when this parameter changes.
#
inet_interfaces = all
#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost
```

Allow reception of email coming from an external network.

(Omitted)

```
# The mydestination parameter specifies the list of domains that this
# machine considers itself the final destination for.
#
# These domains are routed to the delivery agent specified with the
# local_transport parameter setting. By default, that is the UNIX
# compatible delivery agent that lookups all recipients in /etc/passwd
# and /etc/aliases or their equivalent.
#
# The default is $myhostname + localhost.$mydomain.  On a mail domain
# gateway, you should also include $mydomain.
#
# Do not specify the names of virtual domains - those domains are
# specified elsewhere (see VIRTUAL_README).
#
# Do not specify the names of domains that this machine is backup MX
# host for. Specify those names via the relay_domains settings for
# the SMTP server, or use permit_mx_backup if you are lazy (see
# STANDARD_CONFIGURATION_README).
#
# The local machine is always the final destination for mail addressed
# to user@[the.net.work.address] of an interface that the mail system
# receives mail on (see the inet_interfaces parameter).
#
# Specify a list of host or domain names, /file/name or type:table
# patterns, separated by commas and/or whitespace. A /file/name
# pattern is replaced by its contents; a type:table is matched when
# a name matches a lookup key (the right-hand side is ignored).
# Continue long lines by starting the next line with whitespace.
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
#mydestination = $myhostname, localhost.$mydomain, localhost
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#        mail.$mydomain, www.$mydomain, ftp.$mydomain
```

Allow reception of email destined for your domain.

(Omitted)

```
# Alternatively, you can specify the mynetworks list by hand, in
# which case Postfix ignores the mynetworks_style setting.
#
```

File name: `/usr/local/etc/postfix/main.cf`

```
# Specify an explicit list of network/netmask patterns, where the
# mask specifies the number of bits in the network part of a host
# address.
#
# You can also specify the absolute pathname of a pattern file instead
# of listing the patterns here. Specify type:table for table-based lookups
# (the value on the table right-hand side is not used).
#
#mynetworks = 168.100.189.0/28, 127.0.0.0/8
mynetworks = 192.168.0.0/16, 127.0.0.0/8, [2001:db8::]/32, [::1]/128
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/usr/local/etc/postfix/network_table
(Omitted)
# ALIAS DATABASE
#
# The alias_maps parameter specifies the list of alias databases used
# by the local delivery agent. The default list is system dependent.
#
# On systems with NIS, the default is to search the local alias
# database, then the NIS alias database. See aliases(5) for syntax
# details.
#
# If you change the alias database, run "postalias /etc/aliases" (or
# wherever your system stores the mail alias file), or simply run
# "newaliases" to build the necessary DBM or DB file.
#
# It will take a minute or so before changes become visible.  Use
# "postfix reload" to eliminate the delay.
#
#alias_maps = dbm:/etc/aliases
alias_maps = hash:/etc/aliases
#alias_maps = hash:/etc/aliases, nis:mail.aliases
#alias_maps = netinfo:/aliases

# The alias_database parameter specifies the alias database(s) that
# are built with "newaliases" or "sendmail -bi".  This is a separate
# configuration parameter, because alias_maps (see above) may specify
# tables that are not necessarily all under control by Postfix.
#
#alias_database = dbm:/etc/aliases
#alias_database = dbm:/etc/mail/aliases
alias_database = hash:/etc/aliases
#alias_database = hash:/etc/aliases, hash:/opt/majordomo/aliases
(Omitted)
# DELIVERY TO MAILBOX
#
# The home_mailbox parameter specifies the optional pathname of a
# mailbox file relative to a user's home directory. The default
# mailbox file is /var/spool/mail/user or /var/mail/user.  Specify
# "Maildir/" for qmail-style delivery (the / is required).
#
#home_mailbox = Mailbox
#home_mailbox = Maildir/

# The mail_spool_directory parameter specifies the directory where
# UNIX-style mailboxes are kept. The default setting depends on the
# system type.
#
```

Only allow processing of emails coming from specified addresses. Enclose IPv6 addresses in square brackets ([]).

File name: `/usr/local/etc/postfix/main.cf`	
`mail_spool_directory = /var/mail`	Directory where email is stored
`#mail_spool_directory = /var/spool/mail`	
(Omitted)	
`# SHOW SOFTWARE VERSION OR NOT`	
`#`	
`# The smtpd_banner parameter specifies the text that follows the 220`	
`# code in the SMTP server's greeting banner. Some people like to see`	
`# the mail version advertised. By default, Postfix shows no version.`	
`#`	
`# You MUST specify $myhostname at the start of the text. That is an`	
`# RFC requirement. Postfix itself does not care.`	
`#`	
`#smtpd_banner = $myhostname ESMTP $mail_name`	
`#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)`	
`smtpd_banner = $myhostname ESMTP unknown`	Hide the mail server name.
(Omitted)	
`inet_protocols = ipv4 , ipv6`	Specify that the program supports both IPv4 and IPv6. You must add this line because it does not exist in the default file.

(3) Configuration

(3.1) Stopping sendmail

In FreeBSD, sendmail starts by default.

The following is a result of the ps command that checks that sendmail is running.

```
# ps –ax | grep sendmail
1026 ??  Ss  0:00.34 sendmail: accepting connections (sendmail)
1030 ??  Is   0:00.01 sendmail: Queue runner@00:30:00 for /var/spool/clientmqueue
(sendmail)
```

This default setting causes multiple SMTP servers to start. To prevent this, stop sendmail.

In this example, the sendmail startup script is deleted.

```
# cd /etc/rc.d
# rm –rf sendmail
```

(3.2) Creating aliases.db

You need aliases.db.

aliases.db for sendmail is in the directory /etc/mail. Copy the file to /etc.

```
# cp /etc/mail/aliases.db /etc
```

If the file does not exist, execute the following commands to create the file, and then copy the file to the directory /etc:

```
# newaliases
# cp /etc/mail/aliases.db /etc
```

(4) Starting Postfix

Execute the following command as a user with root permissions:

```
    # /usr/local/sbin/postfix start
```

The Postfix program starts and runs as an SMTP server.

You can check log entries in the log file as follows:

```
    # tail /var/log/maillog
```

(5) Email storage location

The configuration in this subsection specifies the following as the email storage location:

```
    /var/mail
```

A file for storing mail messages is created for each user.

4.3.2. Qpopper - FreeBSD

This subsection explains how to configure the POP3 daemon Qpopper in FreeBSD. Qpopper does not support IPv6, but using ports for installation of the daemon automatically applies IPv6 patches. This makes it possible to obtain email in IPv6 by POP3.

The description in this subsection is based on Qpopper 4.0.9 (the most recent version as of February 1, 2008)#.

#: See the Qpopper website: http://www.eudora.com/products/unsupported/qpopper/

(1) Installing Qpopper

In this subsection, `ports` is used to install the program.

Table 4.3-3 Installing Qpopper 4.0.9

Installing Qpopper	

As a root user, execute the following commands:
`# cd /usr/ports/mail/qpopper`
`# make install`
	(Obtain `ports.tar.gz` on the website beforehand.)
	Change the directory.
	Install Qpopper.
	This compiles the program, which takes some time.

The following is the installed program:

```
    /usr/local/libexec/qpopper
```

(2) Starting Qpopper

Qpopper is generally started via inetd or xnetd.

Add the two lines shown in ***Table 4.3-4*** to `/etc/inetd.conf`.

Table 4.3−4 Qpopper settings

File name: `/etc/inetd.conf`

```
pop3 stream tcp nowait root /usr/local/libexec/qpopper qpopper –s
pop3 stream tcp6 nowait root /usr/local/libexec/qpopper qpopper -s
```

(3) Reloading inetd.conf

After editing `inetd.conf`, execute the following command to reload `inetd.conf` and apply the changes:

```
     # kill -HUP `cat /var/run/inetd.pid`
```

5. IPv6 Communication

This chapter explains how IPv6 communication takes place.

(1) Automatically generating IPv6 addresses based on router advertisements (RA)

When IPv6 terminals start, they receive RA messages from IPv6 routers and switches, and IPv6 addresses are automatically generated. The following table describes how IPv6 addresses are obtained.

Table 5-1 Automatic generation of IPv6 addresses

No.	Action	Address	
1	A terminal starts.		
2	The terminal sends a router solicitation. ICMPv6 ICMP Type: 133	Dst.MAC	33:33:*xx*:*xx*:*xx*:*xx* (where *xx*:*xx*:*xx*:*xx* is the lower 32 bits of the destination IPv6 address.)
		Src.MAC	Sender MAC address
		Dst.IPv6	ff02::2 Link-Local Scope: All Routers Address[#]
		Src.IPv6	Sender LLA
3	An IPv6 router sends back a router advertisement. The prefix is reported. ICMPv6 ICMP Type: 134	Dst.MAC	33:33:*xx*:*xx*:*xx*:*xx* (where *xx*:*xx*:*xx*:*xx* is the lower 32 bits of Dst.IPv6)
		Src.MAC	Sender MAC address
		Dst.IPv6	ff02::1 Link-Local Scope: All Nodes Address[#]
		Src.IPv6	LLA of the router that sent the message
4	The terminal uses the following to automatically generate IPv6 address: Upper 64 bits: Received prefix Lower 64 bits: EUI-64		

#: See ftp://ftp.rfc-editor.org/in-notes/rfc2375.txt for RFC 2375.

(2) Starting IPv6 communication

When a terminal starts to communicate with another terminal for the first time, the source terminal does not know the mapping between the IPv6 and MAC addresses of the target terminal. In IPv4, ARP is used to resolve addresses. In IPv6, NDP is used to resolve addresses. The flow from address resolution via NDP through the establishment of communication is shown below.

Table 5-2 Starting IPv6 communication

No.	Action	Address	
1	A source terminal starts communication with a target terminal. Two types of communication: - Within the same subnet - Between global addresses are possible.		
2	The source terminal sends a neighbor solicitation. ICMPv6 ICMP Type: 135	Dst.MAC	33:33:*xx*:*xx*:*xx*:*xx* (where *xx*:*xx*:*xx*:*xx* is the lower 32 bits of Dst.IPv6)
		Src.MAC	Sender MAC address
		Dst.IPv6	ff02::1:ff*xx*:*xxxx* *xx*:*xxxx* - Lower 24 bits of the source MAC address Link-Local Scope: Solicited-Node Address#
		Src.IPv6	Global IPv6 address of the source terminal
3	The target terminal sends back a Neighbor advertisement. ICMPv6 ICMP Type: 136	Dst.MAC	Source Sender MAC address
		Src.MAC	Sender MAC address
		Dst.IPv6	Global IPv6 address of the source sender
		Src.IPv6	Global IPv6 address of the sender
4	Communication starts.		

#: See ftp://ftp.rfc-editor.org/in-notes/rfc2375.txt for RFC 2375.

(3) Communication using a link-local address (LLA)

A link-local address must be unique only within a single subnet. You can use the same LLA across different subnets.

In the figure below, the Switch is connected to three different subnets, each of which contains an IPv6 terminal. The LLAs of the terminals are all fe80::10. For the AX6708S to communicate with the LLA of terminal A, the operator must explicitly specify the interface.

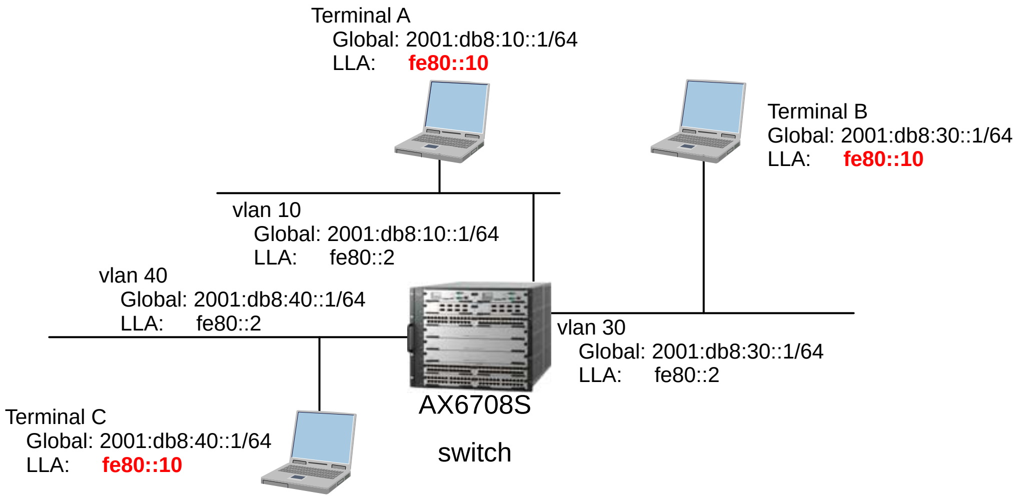

Terminal A
Global: 2001:db8:10::1/64
LLA: fe80::10
Terminal B
Global: 2001:db8:30::1/64
LLA: fe80::10
vlan 10
Global: 2001:db8:10::1/64
LLA: fe80::2
vlan 40
Global: 2001:db8:40::1/64
LLA: fe80::2
vlan 30
Global: 2001:db8:30::1/64
LLA: fe80::2
AX6708S
switch
Terminal C
Global: 2001:db8:40::1/64
LLA: fe80::10

Figure 5-1 LLA diagram

Table 5-3 Communication with LLA

Command	Description
AX series switch	

`# ping ipv6 fe80::8%VLAN0010`
`# telnet fe80::8%VLAN0010`
	After specifying the LLA, specify the following (where <*interface-name*> is the VLAN ID):
	%<*interface-name*>
	Use upper-case letters to specify the VLAN. The number must be a four-digit value.

FeeBSD	

`# ping6 fe80::8%rl0`
`# telnet fe80::8%rl0`
	After specifying the LLA, specify the following (where <*interface-name*> is the NIC driver name):
	%<*interface-name*>

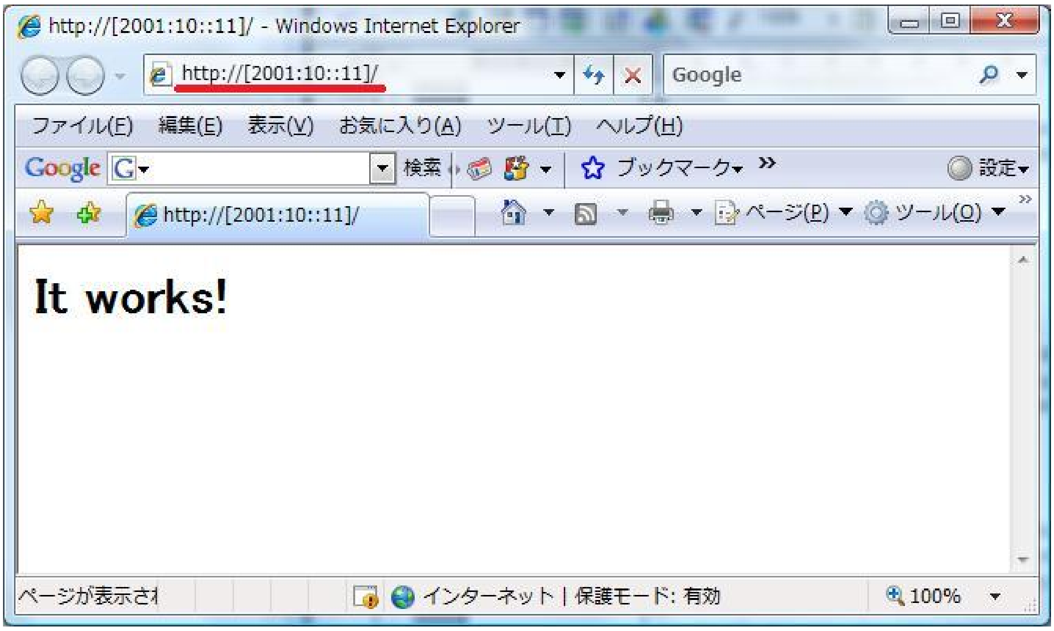
Windows Vista	

`C:¥> ping fe80::9%8`
`C:¥> telnet fe80::9%8`
	After specifying the LLA, specify the following:
	%V<*interface-name*>
	Use the `ipconfig` command to check the interface name.

(4) Specifying addresses in the browser

To directly enter an IPv6 address in a browser, enclose the address in square brackets (`[]`).

Figure 5-2 Directly entering an IPv6 address in a browser (IE)


http://[2001:10::11]/ - Windows Internet Explorer
http://[2001:10::11]/
Google
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
Google G 検索 ブックマーク 設定
http://[2001:10::11]/
ページ(P) ツール(O)
ページが表示され
インターネット | 保護モード: 有効
100%

It works!

Edition 2 – May 19, 2010

Network Technical Support
ALAXALA Networks Corporation

Shin-Kawasaki Mitsui Bldg West Tower, 890
Kashimada, Saiwai-ku, Kawasaki-shi,
Kanagawa 212-0058, JAPAN
http://www.alaxala.com/en/index.html