

PureFlow WSX

Unified Network Controller NF7600 series Configuration Guide TCP Acceleration Edition

Fifth Edition

- For safety and warning information, please read this manual before attempting to use the equipment.
- Additional safety and warning information is provided within the Unified Network Controller Operation Manual (NF7600-W011E). Please also refer to this document before using the equipment.
- Keep this manual with the equipment.

ANRITSU NETWORKS CO., LTD.

Safety Symbols

To prevent the risk of personal injury or loss related to equipment malfunction, Anritsu Networks Co., Ltd. uses the following safety symbols to indicate safety-related information. Ensure that you clearly understand the meanings of the symbols BEFORE using the equipment. Some or all of the following symbols may be used on all Anritsu Networks Co., Ltd. equipment. In addition, there may be other labels attached to products that are not shown in the diagrams in this manual.

Symbols used in manual



DANGER

This indicates a very dangerous procedure that could result in serious injury or death if not performed properly.



WARNING

This indicates a hazardous procedure that could result in serious injury or death if not performed properly.



CAUTION

This indicates a hazardous procedure or danger that could result in light-to-severe injury, or loss related to equipment malfunction, if proper precautions are not taken.

Safety Symbols Used on Equipment and in Manual

The following safety symbols are used inside or on the equipment near operation locations to provide information about safety items and operation precautions. Ensure that you clearly understand the meanings of the symbols and take the necessary precautions BEFORE using the equipment.



This indicates a prohibited operation. The prohibited operation is indicated symbolically in or near the barred circle.



This indicates an obligatory safety precaution. The obligatory operation is indicated symbolically in or near the circle.



This indicates a warning or caution. The contents are indicated symbolically in or near the triangle.



This indicates a note. The contents are described in the box.

PureFlow WSX
Unified Network Controller NF7600 series
Configuration Guide

9 March 2016 (First Edition)
30 October 2017 (Fifth Edition)

Copyright © 2016-2017, ANRITSU NETWORKS CO., LTD.

All rights reserved. No part of this manual may be reproduced without the prior written permission of the publisher.

The contents of this manual may be changed without prior notice.

Printed in Japan

Anritsu Networks Co., Ltd. Contact

In the event that this equipment malfunctions, contact an Anritsu Networks Co., Ltd. Service and Sales office. Contact information can be found on the last page of the printed version of this manual, and is available in a separate file on the CD-ROM.

Maintenance Contract

Anritsu Networks Co., Ltd. can provide a range of optional services under a maintenance contract. For details, contact an Anritsu Networks Co., Ltd. Service and Sales office.

Notes On Export Management

This product and its manuals may require an Export License/Approval by the Government of the product's country of origin for re-export from your country.

Before re-exporting the product or manuals, please contact us to confirm whether they are export-controlled items or not.

When you dispose of export-controlled items, the products/manuals need to be broken/shredded so as not to be unlawfully used for military purpose.

Trademark or Registered Trademark

Windows, Windows Server, and Active Directory are trademarks or registered trademarks of Microsoft Corporation in the United States of America and other countries.

About This Manual

This operation manual describes how to configure and use the software running on the PureFlow WSX Unified Network Controller (hereinafter "this device"). This manual is intended for network administrators who install, implement, and administer this device. This manual is aimed at readers who have basic knowledge about the following aspects of internetworking:

- Local area networks (LAN)
- Ethernet
- Internet protocol (IP)

This manual is applicable to the following models of this equipment:

- NF7601A
- NF7602A
- NF7603A
- NF7604A
- NF7605A

The manual of this device consists of the following four manuals. This document is <3>.

<1> Operation Manual TCP Acceleration Edition (NF7600-W011E)

Describes in detail the installation and handling of this device.

<2> Command Reference TCP Acceleration Edition (NF7600-W012E)

Describes in detail the commands used in this device.

<3> Configuration Guide TCP Acceleration Edition (NF7600-W013E)

Describes the basic features of this device and provides specific examples of the settings required to build a network using these features.

<4> Web GUI Operation Manual TCP Acceleration Edition (NF7600-W014E)

Describes the operation for setting and display of this device using a Web browser.

If the following documents related to this device or other documents related to the features of this device are issued, be sure to read them:

Release notes

(For details of the issuance of release notes, contact your dealer or an Anritsu Networks Co., Ltd. customer support center.)

Table of Contents

About This Manual	I
Chapter 1 Overview of the Software.....	1-1
Chapter 2 Basic Features	2-1
2.1 Traffic Control	2-2
2.2 Link-down Transfer	2-2
2.3 SSH.....	2-2
2.4 Simple Network Management Protocol (SNMP)	2-2
2.5 Statistics.....	2-2
2.6 RADIUS.....	2-3
2.7 WebAPI	2-3
2.8 WebGUI.....	2-3
2.9 OpenFlow Function.....	2-3
2.10 Network Bypass Function.....	2-3
2.11 Top Counter Function	2-3
Chapter 3 Configuring Settings	3-1
3.1 Command Line Interface (CLI)	3-2
3.2 Command Structure.....	3-3
3.3 Command Syntax	3-4
3.4 Help Feature.....	3-5
3.5 Command Omission and Fill In	3-5
3.6 History Feature	3-6
3.7 Command Edit Feature	3-7
3.8 Pager Feature.....	3-8
3.9 Launch and Login	3-9
3.10 How to Save the Settings.....	3-11
3.11 How to Restore the Settings.....	3-11
3.12 Startup Time	3-12
Chapter 4 Displaying and Setting Information	4-1
4.1 Date/Time	4-2
4.2 Simple Network Time Protocol (SNTP).....	4-4
4.3 User Name and Password	4-5
4.4 syslog	4-6
4.5 Module Information	4-9
4.6 License Key	4-12

Chapter 5 Ethernet Port Settings	5-1
Chapter 6 Network Port Settings	6-1
6.1 Overview	6-2
6.2 Setting Network Port Attributes	6-4
6.3 Setting the Maximum Frame Length.....	6-6
6.4 Checking Settings and States	6-8
Chapter 7 System Interface Settings	7-1
7.1 Overview	7-2
7.2 System Interface Communication.....	7-3
7.3 System Interface Filter	7-8
7.4 Configuration Examples	7-9
7.5 Checking Settings and States	7-11
Chapter 8 Traffic Control	8-1
8.1 Overview	8-2
8.2 Traffic Acceleration	8-3
8.3 Traffic Shaping.....	8-3
8.4 Application to Large-scale Network	8-4
8.5 Channel	8-5
8.6 Scenario	8-6
8.7 Hierarchical Scenario.....	8-10
8.8 Acceleration Tunnel	8-15
8.9 Setting Procedure	8-17
8.10 How to Set a Rule List.....	8-37
8.11 Channel interface communication	8-39
8.12 Application Acceleration Function.....	8-41
8.13 Configuration Example.....	8-46
8.14 Advanced Settings.....	8-54
8.15 Address during the traffic acceleration.....	8-86
Chapter 9 Link-down Transfer.....	9-1
9.1 Link-down Transfer	9-2
Chapter 10 SSH	10-1
10.1 Overview	10-2
10.2 Specifications	10-3
10.3 Using SSH.....	10-4

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Appendix

Chapter 11	SNMP Setting.....	11-1
11.1	Overview of SNMP.....	11-2
11.2	SNMPv1/SNMPv2c Setting.....	11-3
11.3	SNMPv3 Setting.....	11-5
11.4	TRAP Setting.....	11-7
Chapter 12	Statistics.....	12-1
12.1	Port Statistics.....	12-2
12.2	Scenario Statistics.....	12-3
Chapter 13	RADIUS.....	13-1
13.1	Overview.....	13-2
13.2	Controlling Login Authentication.....	13-3
13.3	Controlling Login Mode.....	13-3
13.4	Setting Up the RADIUS Feature.....	13-4
13.5	RADIUS Server Settings.....	13-6
Chapter 14	Downloading and Uploading Data.....	14-1
14.1	Downloading/Uploading Software.....	14-2
14.2	Downloading the Software Update Patch.....	14-6
14.3	Downloading/Uploading Configuration Data.....	14-7
14.4	Restarting the Software.....	14-11
Chapter 15	WebAPI.....	15-1
15.1	Overview.....	15-2
15.2	Communication Protocol.....	15-3
15.3	HTTP Methods.....	15-3
15.4	JSON Format.....	15-4
15.5	API List.....	15-5
15.6	Common Error Messages.....	15-6
15.7	List of Error Messages.....	15-7

Chapter 16 OpenFlow Function.....	16-1
16.1 Overview	16-2
16.2 OpenFlow Version	16-3
16.3 Supported OpenFlow Messages	16-4
16.4 OpenFlow Messages Supported for CLI Commands ...	16-6
16.5 JSON Format.....	16-7
16.6 Supported Command List.....	16-8
16.7 Common Error Messages	16-9
16.8 Error Message List.....	16-10
Chapter 17 Network Bypass Function	17-1
17.1 Overview	17-2
17.2 Setting and Checking the Function.....	17-3
17.3 Precautions.....	17-5
Chapter 18 Top Counter	18-1
18.1 Overview	18-2
18.2 Display Unit of the Top Counter	18-2
18.3 Measurement Range of the Top Counter.....	18-3
18.4 Traffic Counter	18-4
18.5 Measuring Traffic at Specific Application Ports	18-5
18.6 Operation Command List	18-5
18.7 Operation Procedure.....	18-6
18.8 Operation Example	18-7
18.9 Cautions	18-9
Appendix A Default Values.....	A-1
Appendix B syslog Messages.....	B-1
Appendix C List of SNMP Traps	C-1
Appendix D Enterprise MIB List.....	D-1
Appendix E JSON Format	E-1

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Appendix

Appendix F	Details of WebAPI.....	F-1
Appendix G	WebAPI Sample Programs	G-1
Appendix H	Details of OpenFlow Message Supported for CLI Command.	H-1
Appendix I	Details of OpenFlow Messages	I-1

Chapter 1 Overview of the Software

This chapter provides an overview of the software of this device.

The basic features are as follows:

- Traffic control
- Link-down transfer
- SSH
- Simple Network Management Protocol (SNMP)
- Statistics
- RADIUS
- WebAPI
- WebGUI
- OpenFlow Function
- Network Bypass Function
- Top Counter

(Blank page)

Chapter 2 Basic Features

This chapter describes the basic features of the software of this device.

2.1	Traffic Control	2-2
2.2	Link-down Transfer	2-2
2.3	SSH.....	2-2
2.4	Simple Network Management Protocol (SNMP)	2-2
2.5	Statistics	2-2
2.6	RADIUS.....	2-3
2.7	WebAPI	2-3
2.8	WebGUI.....	2-3
2.9	OpenFlow Function.....	2-3
2.10	Network Bypass Function.....	2-3
2.11	Top Counter Function	2-3

2.1 Traffic Control

This device has a traffic acceleration function and traffic shaping function.

Currently, the demand for the centralized server/storage allocation model is increasing at the data center to reduce operational costs and enhance security. To securely recover server data in case of a disaster, demand for remote transfer of backup data is increasing. However, as transfer of TCP/IP uses a lot of data, communication decelerates due to line delay. The traffic acceleration function of this device enhances the TCP/IP transfer performance affected by line delay, and provides high-speed data communication. The traffic shaping function prevents the packet from being discarded in the network, and accelerates the TCP/IP transfer rate.

For details about traffic acceleration and traffic shaping, see Chapter 8 “Traffic Control”.

2.2 Link-down Transfer

When a link-down is detected on one side of the link, this feature brings down the other side of the link and reports a link error.

For details about link down transfer, see Chapter 9 “Link-down Transfer”.

2.3 SSH

The SSH server feature encrypts communication between this device and SSH clients, enabling secure remote operation even via a network where safety is not guaranteed.

For details about SSH, see Chapter 10 “SSH”.

2.4 Simple Network Management Protocol (SNMP)

SNMP is a protocol to remotely manage network devices such as routers and servers over the network.

For details about SNMP, see Chapter 11 “SNMP Setting”.

2.5 Statistics

Statistics information includes information on counters and queue buffers.

For details about statistics information, see Chapter 12 “Statistics”.

2.6 RADIUS

The RADIUS feature performs user authentication by using RADIUS (RFC2865) upon a log in to Telnet, SSH, or a serial console.

For details about RADIUS, see Chapter 13 “RADIUS”.

2.7 WebAPI

The WebAPI feature performs the settings by using HTTP (Hypertext Transfer Protocol: RFC2616) upon setting of the traffic control feature of this device.

For details about WebAPI, see Chapter 15 “WebAPI”.

2.8 WebGUI

The WebGUI feature performs the settings and displays of this device by using a Web browser of the terminal connected to the network.

For details about WebGUI, see “WebGUI Operation Manual (NF7600-W014E)”.

2.9 OpenFlow Function

The OpenFlow function uses the OpenFlow protocol for setting of the traffic control function of this equipment.

For detailed description of the OpenFlow function, refer to "Chapter 16 OpenFlow Function".

2.10 Network Bypass Function

NF7603A and NF7604A and NF7605A have the Network port bypass function. This function can secure a communication path by bypassing the Network port when an equipment error occurs.

For a detailed description of the network bypass function, refer to "Chapter 17 Network Bypass Function".

2.11 Top Counter Function

The top counter feature helps you to understand the usage status of traffic.

For details about the top counter, see Chapter 18 “Top Counter”.

(Blank page)

Chapter 3 *Configuring Settings*

This chapter describes how to configure settings.

3.1	Command Line Interface (CLI)	3-2
3.2	Command Structure.....	3-3
3.3	Command Syntax	3-4
3.4	Help Feature.....	3-5
3.5	Command Omission and Fill In	3-5
3.6	History Feature	3-6
3.7	Command Edit Feature	3-7
3.8	Pager Feature.....	3-8
3.9	Launch and Login	3-9
3.10	How to Save the Settings.....	3-11
3.11	How to Restore the Settings.....	3-11
3.12	Startup Time	3-12

Settings for this device are configured by using the Command Line Interface (hereafter referred to as “CLI”). CLI enables remote access to the terminal connected to the console port via a console cable, and remote access to the system's IP network interface (system interface) via Telnet and SSH on the network. Communication to the system interface can be performed via the Ethernet port.

3.1 Command Line Interface (CLI)

CLI is used to configure and display the operating parameters of the system. For details about the commands, see “PureFlow WSX Unified Network Controller NF7600 series Command Reference”.

(1) Console port

Connection conditions of the console port are as follows:

Communication speed:	9600 bits/s
Character length:	8 bits
Parity:	None
Stop bit length:	1 bit
Flow control:	None

The serial interface for connecting the console is located on the front of this device. Use the supplied console cable to connect the console.

Note:

When the communication speed is set to 115200 bits/s, the text may be corrupted or omitted depending on the environment used (device hardware, software). If this happens, lower the communication speed.

This device can change the communication speed to any of 9600 bits/s, 19200 bits/s, 38400 bits/s, or 115200 bits/s by using the "set console baudrate" command.

(2) Telnet

To use Telnet, the system interface of this device must be set up. Up to 4 sessions can be used simultaneously for SSH and Telnet sessions.

Use Telnet on a device connected to the network via the Ethernet port.

For more information on system interface settings, see Chapter 7 “System Interface Settings”.

If you do not use Telnet, run the “**set telnet**” command to disable Telnet.

(3) SSH

SSH (Secure Shell) for this device supports SSH Version 2. Up to 4 sessions can be used simultaneously for SSH and Telnet sessions.

If you do not use SSH, run the “**set ssh**” command to disable SSH.

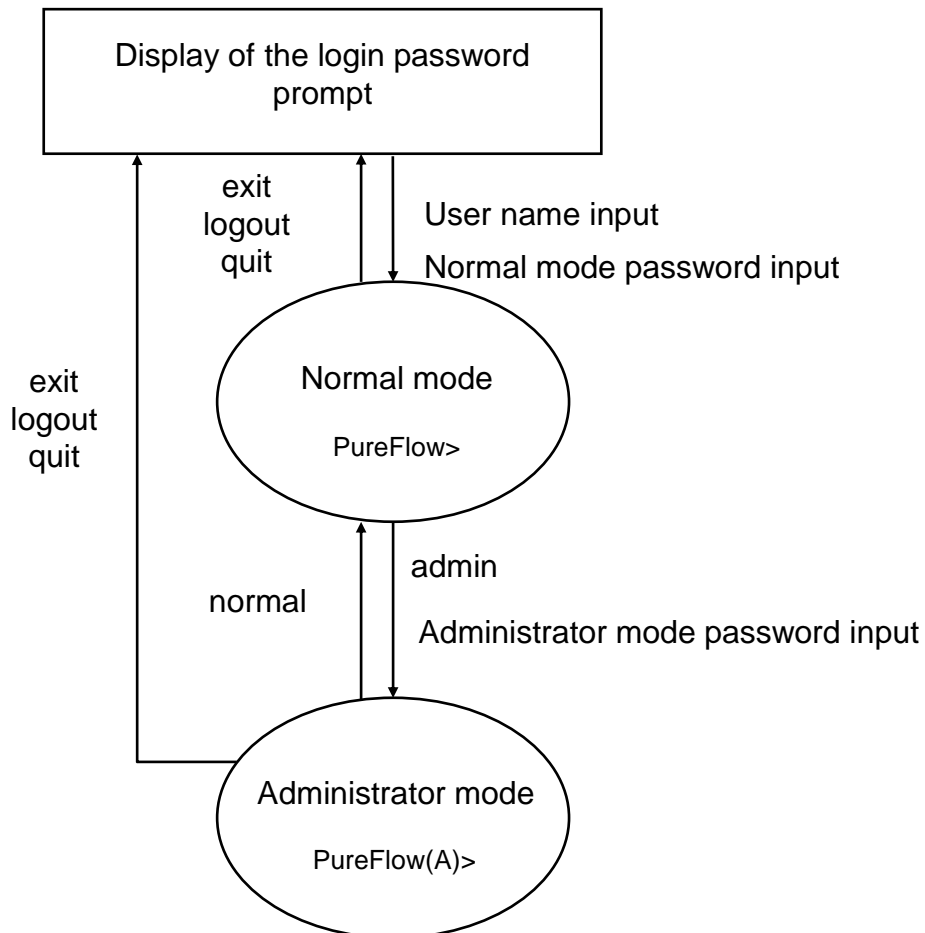
3.2 Command Structure

This device supports two types of CLI: normal mode and administrator mode. In normal mode, you can only display the status, counter, and setting values. In administrator mode, you can set, modify, and display all settings.

To maintain device security, you can set passwords to enter the normal mode and administrator mode separately. If passwords are set, users have to provide the correct password to enter these modes.

When the RADIUS feature is used for login authentication, you can enter the normal mode or administrator mode according to the service type specified per user on the RADIUS server. For details, see the “RADIUS feature section”.

CLI prompt	CLI mode
PureFlow>	Normal mode
PureFlow(A)>	Administrator mode



3.3 Command Syntax

The CLI command syntax for this device is as follows:

Action Item Value

E.g.

Action	Item	Value
↓	↓	↓
Set	Time	Value
↓	↓	↓
set	date	20150430101010

Since there various setting items for each feature, some setting items are grouped in layers, for example “Item” is “Group + Item”.

Example of a setting group

ip
scenario
port

Following is an example of the command syntax for a setting group:

Action	Group	Item	Value
↓	↓	↓	↓
Set	PORT group	SPPED 1/2	Fixed to 100 M
↓	↓	↓	↓
set	port	speed 1/2	100M

3.4 Help Feature

Input a question mark (?) on the system prompt or middle of a command to show a list of commands available for each command input mode.

```
PureFlow(A)> ?
```

Command	Description
?	Lists the top-level commands available
add	Adds some parameters, use 'add ?' for more information
arp	Shows address resolution table and control
clear	Clears system statistics, use 'clear ?' for more information
delete	Deletes some parameters, use 'delete ?' for more information
.	.
.	.

```
PureFlow(A)> set port ?
```

flow_control	Sets the flow control parameters
speed	Sets the port speed

Note:

The question mark (?) should be input at the end of the command line to use the help feature.

3.5 Command Omission and Fill In

A command can partially be omitted if it is identifiable. For example, the save, set, and show commands, which start with the letter “s”, have different second letters, so when “se” is input, the “set” command can be determined. The following two commands have the same meaning:

```
set port autonegotiation 1/2 disable = se po au 1/2 d
```

Input the minimum letters required to distinguish a command, and then press the **TAB** key to display the rest of the command.

```
PureFlow(A)> set po<TAB>
      ↓
PureFlow(A)> set port
```

Note:

The fill-in feature using the **TAB** key only works at the tail end of the command line. The omission and **TAB** key fill-in feature may not work depending on the command keywords. In this case, use the help feature to confirm the keyword, and enter the entire keyword.

3.6 History Feature

How to use the command history feature.

CLI has a history (log) feature for input commands.

You can call a command similar to the command you are inputting from the list of recorded commands, and then use the edit feature (described later) to edit and run the command.

Use the following keys to call the command history:

Ctrl-P key or Up arrow key

Calls the latest command in the history buffer. Repeat this key operation to call older commands consecutively.

Ctrl-N key or Down arrow key

Returns to the latest command in the history buffer after a command is called by the Ctrl-P or Up arrow key. Repeat this key operation to call later commands consecutively.

You can also use the **show history** command to show the history of commands.

3.7 Command Edit Feature

The command edit feature provides the following key strokes:

Ctrl-B key or Left arrow key

Moves the cursor one letter back.

Ctrl-F key or Right arrow key

Moves the cursor one letter forward.

Ctrl-A key

Moves the cursor to the start of the line.

Ctrl-E key

Moves the cursor to the end of the line.

Ctrl-D or Delete key

Deletes the letter in front of the cursor.

Ctrl-H or BS key

Deletes the letter behind the cursor.

Ctrl-K key

Deletes all the letters in front of the cursor and copies them to the buffer.

Ctrl-W key

Deletes the letters selected by the cursor and copies them to the buffer.

Ctrl-Y key

Pastes the content of the buffer to the cursor position.

Ctrl-U key

Deletes the line before the cursor and copies it to the buffer.

Note:

The command line edit feature only works for a single line command.

3.8 Pager Feature

When running a command that shows more than 24 lines of data on the terminal, the pager feature pages data in screen or line units. In this case, the message “– More –” is displayed on the last line to indicate there is more data than the data displayed.

When “– More –” is displayed, the following keys can be used:

To disable the pager feature, set by using the command below from CLI:

```
PureFlow(A)> set pager disable
```

On the other hand, to enable the pager feature, set by using the command below from CLI:

```
PureFlow(A)>set pager enable
```

When “– More –” is displayed, the following keys can be input.

Space or F key

Shows the next screen.

Enter key

Shows the next line.

Q key

Exits the screen.

3.9 Launch and Login

When the power supply is turned on, this device starts up and automatically reads the software object in the internal flash memory. When this device starts up with a CF card or USB flash drive (“external media” hereafter) containing the software object (nf7600.bin) connected, it reads the software object in the external media on a priority basis. For the priority of the external media, USB flash drives take precedence over the CF card and then other external media.

This device also reads the configuration file (extcnf.txt) in the external media on a priority basis if an external media is connected.

Disconnecting the external media or turning off the power supply while this device is accessing the external media to read data may damage the media.

If the terminal is connected to this device's console port, the following launch message is displayed (items in the launch message may differ depending on the software version).

```
Anritsu PureFlow NF7600-S001A Software Version 2.1.1
Copyright 2016-2017 ANRITSU NETWORKS CO., LTD. All rights reserved.

Power Supply 0          ... [OK]
Power Supply 1          ... [NONE]
Fan 0                   ... [OK]
Fan 1                   ... [OK]
Serial Port             ... [OK]
Backup Memory Checking ... [OK]
Real Time Clock Checking ... [OK]
File System Checking   ... [OK]
EEPROM Checking        ... [OK]
Ethernet Controller Checking
    Management Port     ... [OK]
    Internal Port       ... [OK]

Software License : NF7600-L201A (TCP Acceleration Software)

.
Loading Forwarding Processor module software
..... completed

Slot 1 boot up complete
    Medium type 10GBase-R 4 ports

System booting up
.....
Loading Configuration from Master.

Restoration in Progress
```

```
100 % done

Restoration completed

PureFlow login:
```

When configuring settings, connect this device as the system console to the console port via the console cable. With the console connected, press the **Enter** key to show the following message for login:

```
PureFlow login:
```

The user name of this device is “root”. By factory default, no login password is set. When the login is authenticated, the prompt is displayed to accept commands.

```
PureFlow login:root
Password: (Press the Enter key)
PureFlow>
```

In the normal mode, you can view the settings but cannot modify them. You need to activate the administrator mode to configure the settings. To do so, run the “admin” command.

```
PureFlow>admin
Enter the Admin Password: (Press the Enter key)
PureFlow(A)>
```

In the administrator mode, you can not only view various parameters but also edit operating parameters and set passwords. Multiple users can enter the administrator mode and modify the settings simultaneously. In administrator mode, be sure to specify a password and configure other settings to manage users with administrator privileges.

3.10 How to Save the Settings

Changes to settings are enabled by running respective commands but are lost at shutdown, and not recovered at reboot. This device can save the settings as a configuration file in the internal flash memory. To enable the settings even after a reboot, run the “save” command to save the settings in the internal flash memory.

The saving procedure is as follows:

```
PureFlow(A)> save config
Do you wish to save the system configuration into the flash memory (y/n)? y
.....
Done
PureFlow(A)>
```



CAUTION

The setting value may not be stored properly when the power of this device is turned off before "Done" is displayed on the console screen. In addition, it may possibly cause a failure of the internal flash memory.

3.11 How to Restore the Settings

When the power supply is turned on, this device automatically reads the configuration file saved in the internal flash memory. When this device starts up with the CF card or USB flash drive (hereafter, referred to as external media) containing the configuration file (extcnf.txt) connected, it reads the configuration file in the external media on a priority basis. For the priority of the external media, USB flash drives take precedence over other external media.

Disconnecting the external media or turning off the power supply while this device is accessing the external media to read data may damage the media.

3.12 Startup Time

The execution time of the “save” command and the startup time of this device differ depending on the amount of information in the configuration file. Reference values are shown in the table below.

	save command execution time	Startup time
Default	-	3 minutes 00 seconds
100 scenarios 100 filters	5 seconds	3 minutes 10 seconds

- * For a description of filter and scenario, see Chapter 8 “Traffic Control”.
- * The “save” command execution time and this device startup time may change depending on the number of lines and parameters.

Chapter 4 *Displaying and Setting Information*

This chapter provides how to display the device information and settings.

4.1	Date/Time	4-2
4.2	Simple Network Time Protocol (SNTP).....	4-4
4.3	User Name and Password	4-5
4.4	syslog	4-6
4.5	Module Information	4-9
4.6	License Key	4-12

This device has settings related to the entire device such as time and CLI password, as well as information related to the entire device such as hardware/software versions. This chapter describes how to display such information and specify settings.

The table below lists the device information and setting items of this device.

Date/Time	This is the calendar clock built in the device. It is used for recording syslog events.
SNTP	Simple Network Time Protocol (SNTP) client
User name and password	User name and password for controlling access to the device via CLI
syslog setting	Saves state change events and error events of the device to the internal memory or battery backup memory, or sends them to the remote host.
Module information	Information of each module in the device (such as version)

4.1 Date/Time

This device supports a calendar feature. The date and time are used to record events in syslog. The date and time can be set manually by using CLI commands, and can be adjusted automatically in synchronization with the time of the NTP server by using the SNTP client feature.

Setting the date and time by using CLI commands

Use the following CLI commands to set the date and time:

set date <yyyymmddhhmmss>	Sets the date and time.
set timezone <hours-offset> [<minutes-offset>]	Sets the time zone offset from the UTC (Coordinated Universal Time). The default value is +9 [hours] 0 [minutes].
set summertime from <week> <day> <month> <hh> to <week> <day> <month> <hh> [<offset>]	Sets the application period in summer time (daylight saving time). The default value is that summer time is not set.
unset summertime	Cancels the summer time setting.
show date	Displays the date and time.

The following is a command execution example.

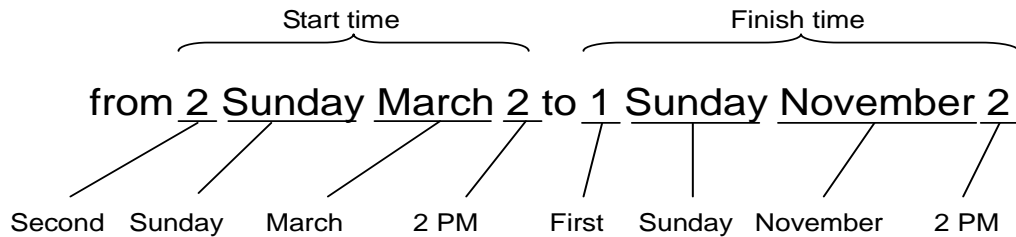
```
PureFlow(A)> set timezone +9
PureFlow(A)> set summertime from 2 Sunday March 2 to 1 Sunday November 2
PureFlow(A)> set date 20120630124530
PureFlow(A)> show date
May 18 2005(Mon) 12:45:32
UTC Offset    : +09:00
Summer Time   : From Second Sunday March 02:00
                To   First Sunday November 02:00
                Offset 60 minutes

PureFlow(A)>
```

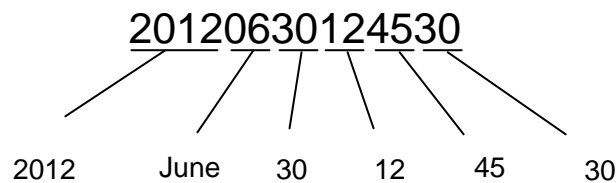
For the time zone setting, enter a signed value indicating the number of hours the time is offset from the UTC (Coordinated Universal Time). Enter a minutes offset value as required.

For the summer time setting, specify the start and end date and time of summer time. Enter the summer time value with a minutes offset value as required. If the minutes offset value is omitted, an offset value of 60 [minutes] is applied.

Specify the start and end date and time of summer time in the format shown below.



To set the date and time, enter the year, month, day, hour, minute, and second using 14 digits in a row.



The time set in the calendar clock is driven by the internal battery and continues even when this device is turned off.

4.2 Simple Network Time Protocol (SNTP)

This device has a SNTP client feature. The SNTP client communicates with the NTP server via the system interface to synchronize the date and time of this device with that of the NTP server. To use the SNTP client, the system interface of this device must be set up. For more information on system interface settings, see Chapter 7 “System Interface Settings”.

To set up the SNTP client, use the following commands.

set sntp {enable disable}	Enables and disables the SNTP client feature. Time synchronization starts when the time set in interval elapses after it is enabled.
set sntp server <IP_address>	This command sets the IP address of the NTP server. Only one NTP server can be specified.
set sntp interval <interval>	Specifies the interval for making regular time inquiries to the NTP server in seconds. The setting range is 60 to 86400 [seconds]. The default value is 3600 [seconds]. Although the values that can be set are as described above, the values for the actual operation are rounded up in 60-second units. Time synchronization starts when the time set in interval elapses after it is changed.
sync sntp	Makes an inquiry to the NTP server about time. This command can be executed only when the SNTP client feature is enabled.
show sntp	Displays the state and settings of the SNTP client function.

To set an NTP server of 192.168.10.10 and an inquiry interval of 86400 seconds, execute the following commands:

```
PureFlow(A)> set sntp server 192.168.10.10
PureFlow(A)> set sntp interval 86400
PureFlow(A)> set sntp enable
PureFlow(A)> sync sntp
Transmitted to the server.
PureFlow(A)> show sntp
Status      : enable
Server      : 192.168.10.10
Interval    : 86400
Sync        : kept
PureFlow(A)>
```

If Sync of the “show sntp” command is “kept”, the device is in synchronization with the NTP server.

4.3 User Name and Password

To ensure the security of the device, authentication with a user name and password is required before device settings are performed on the serial console or via Telnet. The user can change the password.

set password	Sets the login password. The login password can be up to 16 characters.
set adminpassword	Sets the login password to switch to Administrator mode. The login password can be up to 16 characters.

The following is a command execution example:

```
PureFlow(A)> set password
New Password: ← Enter the password to be set.
Retype the new Password: ← Enter the new password again.
```

The following ASCII characters can be used for login passwords:

```
1234567890
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
!#$%&'()*=~-^_|¥@`[]{}:*;+_/.,<>
```

To cancel the login password setting, enter no password for “New Password” and press the **Enter** key.

4.4 syslog

Events that occur in this device, such as error events, link-up, and link-down (hereafter referred to as “log data”), can be recorded in multiple ways. This device can save log data of up to 8000 events in the internal memory when the power is on. The log data saved in the internal memory is lost when the power is turned off. Log data can be recorded in the syslog host over the network as well as in the internal backup memory. The internal backup memory can save log data of up to 1,200 events each for the previous startup and the one before that. The log data saved in the internal backup memory is not lost when the power is turned off.

show syslog	Displays the log data recorded in the internal memory.
show backup syslog [last second_last]	Displays the log data recorded in the internal backup memory.
clear syslog	Clears the log data recorded in the internal memory.
set syslog severity <severity_level>	Specifies the level for recording log data.
show syslog host	Displays settings for system log output.
set syslog host {enable disable}	Enables and disables recording to the syslog host.
add syslog host <IP_address> [<udp_port>]	Adds the IPv4 address and UDP port of the syslog host.
delete syslog host <IP_address>	Deletes the IPv4 address and UDP port of the syslog host.
set syslog facility {ccpu fcpu} <facility_code>	Sets the facility of the system log. ccpu: Log message detected and recorded in the control system CPU fcpu: Log message detected and recorded in the forwarding system CPU

Log data is recorded in the device as text data in the following format:

- The log data saved in the internal memory displayed by using show syslog command.

Data/Time	Host	Ident	PID	Message
Jun 30 16:51:19	PureFlow	System	[10330]	Port 1/1 changed Up from Down.

- The log data saved in the internal backup memory displayed by using show backup syslog command.

Priority	Date/Time	Message
134	2012 Jun 30 16:51:19	Port 1/1 changed Up from Down.

Date/Time

This indicates the date and time when the event occurred.

Host

The name of the host that recorded the system log information. This value is fixed to "PureFlow".

Ident

The identifier of the program that recorded the system log information. This value is fixed to "System".

PID

The process ID of the process that recorded the system log information.

Message

This field contains messages indicating the details of events.

You can also display messages by using the show syslog command.

```
PureFlow> show syslog
-----
Date    Time    Host    Ident  [PID]  Message
-----
Jan 25  21:50:54  PureFlow System  [10330]: Port 1/1 changed Up from Down.
```

Data is saved in the memory when the power is on, but the operator can clear the message.

```
PureFlow(A)> clear syslog
PureFlow(A)> show syslog
-----
Date    Time    Host    Ident  [PID]  Message
-----
PureFlow(A)>
```

Priority

Priority is a code indicating the characteristics of the log message. The priority code is calculated and saved according to the method specified in RFC3164. A priority code is expressed as a combination of two values: Facility indicating the message category and Severity indicating the severity of the message.

$$\text{Priority} = \text{Facility} \times 8 + \text{Severity}$$

You can set the facility of a syslog message in this device. The setting range for facility is 0 to 23. The default value is as follows.

control CPU: 16
forwarding CPU: 17

The following shows a command execution example.

```
PureFlow(A)> set syslog facility ccpu 18 ← Sets facility of the control CPU to 18.  
PureFlow(A)> set syslog facility fcpu 19 ← Sets facility of the forwarding CPU to 19.
```

Severity stores a value from 0 to 6. Priority 0 is the highest severity; the higher the value, the lower the severity. The severity of each message is assigned based on the following standard as specified in RFC 3164:

Numerical Code	Severity	
0	Emergency:	system is unusable
1	Alert:	action must be taken immediately
2	Critical:	critical conditions
3	Error:	error conditions
4	Warning:	warning conditions
5	Notice:	normal but significant condition
6	Informational:	informational messages

For example, a message with priority of 129 ($16 \times 8 + 1$) has a facility of 16 and severity of 1. Therefore, it is an Alert level (emergency) message detected by the control CPU.

4.5 Module Information

This command displays information on each module in the system. The version, production number, and other information can be confirmed.

show module	Displays the module information.
-------------	----------------------------------

The module information includes the following:

Management MAC Address

Indicates the system interface MAC address.

Forwarding MAC Address

Indicates the channel interface MAC address.

Chassis Model Name

Shows the main model name. The model name is as follows:

NF7601A : PureFlow WSX
 NF7602A : PureFlow WSX Lite
 NF7603A : PureFlow WSX
 NF7604A : PureFlow WSX
 NF7605A : PureFlow WSX Lite

Chassis Serial Number

Shows the production No. of the main unit.

Control Module Version

Shows the hardware version of the control module.

Shaper Module Version

Shows the hardware version of the shaper module.

Bypass Module Version

Shows the hardware version of the Bypass module.

Displayed when the model of the device is NF7603A, NF7604A or NF7605A.

Software Version

Shows the version of the installed software.

Software License

Shows the currently-operated software license.

Management U-Boot Version, Forwarding U-Boot Version

Shows the U-Boot version.

MCU-C Version, MCU-S Version, MCU-B Version

Shows the MCU version.

The version of MCU-B is displayed when the model of the device is NF7603A, NF7604A or NF7605A.

Uptime

Shows the operation time starting from startup of this device.

Temperature

Shows the intake temperature.

Power Supply Unit N

Shows the power unit state.

FAN Unit N

Shows the fan unit state.

The following is a command execution example.

```
PureFlow(A)> show module
Anritsu PureFlow NF7600-S001A Software Version 2.1.1
Copyright 2016-2017 ANRITSU NETWORKS CO., LTD. All rights reserved.

Management MAC Address      : 00-00-91-12-34-56
Forwarding MAC Address      : 00-00-91-12-34-57

Chassis Model Name          : NF7605A
Chassis Serial Number       : 1234567890

Control Module Version      : 01A
Shaper Module Version       : 00A
Bypass Module Version       : 00A
Software Version            : 2.1.1
Software License            : NF7600-L201A (TCP Acceleration Software)
Management U-Boot Version   : 1.1.6
Forwarding U-Boot Version   : 1.1.6
MCU-C Version               : 109
MCU-S Version               : 109
MCU-B Version               : 109

Uptime                      : 0 days, 00:27:17
Temperature
  Intake Temperature        : 32°C
Power Supply Unit 0
  Operation Status          : operational
  Fan Speed                 : 6240[rpm]
Power Supply Unit 1
```

```
Operation Status      : not present
Fan Speed             : 0[rpm]
FAN Unit 0
  Operation Status    : operational
  Fan Speed           : 3840[rpm]
FAN Unit 1
  Operation Status    : operational
  Fan Speed           : 3960[rpm]
PureFlow(A)>
```

4.6 License Key

By purchasing a license key, you can extend the functionality and performance of this device.

A license key is provided in the license document. You will be asked the serial number of your device when you purchase a license key after purchasing the device.

To set the license key to the device, enter the “set option” command. When a message prompting you to enter the license key appears, enter the license key. When entering the license key, entry of the hyphens delimiting every 4 characters is optional. The license key you entered and the serial number of the device are compared, and the license becomes available if they match.

The commands related to license keys are as follows:

set option	Sets a license key to this device.
show option	Displays the valid licenses.

The following is a command execution example.

```
PureFlow(A)> set option  
Enter the option key : XFS8wbFEFBNkfqLJ
```

Authentication succeed.

Making be available : License Key NF7600-L214A (10G Bandwidth License)

Updation done.

Enter update scenario command to change port bandwidth.

```
PureFlow(A)>  
PureFlow(A)> show option  
License Key NF7600-L214A available (10G Bandwidth License)  
PureFlow(A)>
```


Chapter 5 Ethernet Port Settings

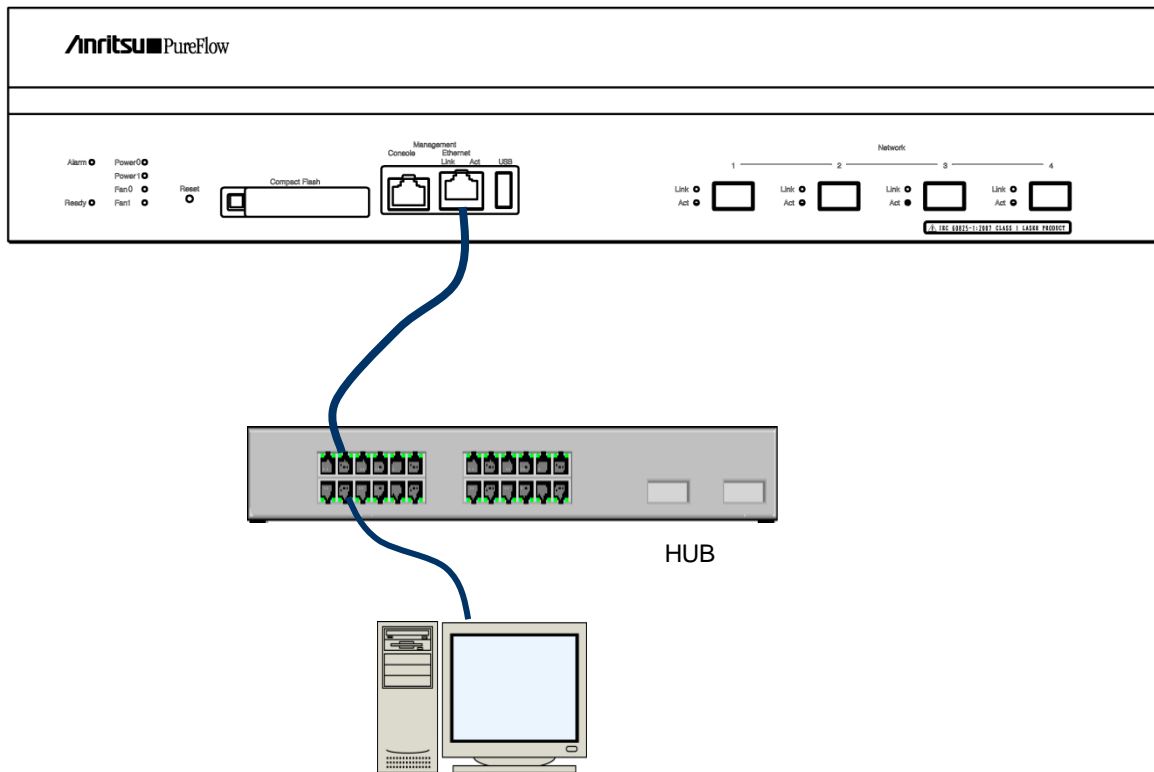
This device has an Ethernet port in the front for remote setting and control over the network. This port, which is a local port for management, is separate from the Network port. This port is a 10/100/1000 BASE-T port that supports Auto-MDIX.

The following settings are effective for the Ethernet port.

Enabling/Disabling AutoNegotiation (Refer to Note 1.)

Communication speed (10 Mbit/s, 100 Mbit/s, 1 Gbit/s) (Refer to Note 2.)

duplex mode (full, half) (Refer to Note 2.)



5

Ethernet Port Settings

For remote setting and control over the network connected to the Ethernet port, the system interface of this device must be set up. For more information on system interface settings, see Chapter 7 “System Interface Settings”.

Note 1:

For the communication at 1 Gbit/s of 10/100/1000BASE-T (RJ-45/SFP), enable AutoNegotiation.

Note 2:

The communication speed and duplex mode settings are effective only when AutoNegotiation is disabled. If AutoNegotiation is enabled, the result of AutoNegotiation is reflected and this setting is not applied, while if AutoNegotiation is disabled, this setting is applied. If the link status of the "show port" command is half duplex, check that AutoNegotiation, communication speed, and duplex mode setting are suitable for the connected device.

Note 3:

The maximum frame length of the Ethernet port is fixed to 1518 bytes.

(Blank page)

Chapter 6 Network Port Settings

This chapter describes the Network port settings of this device.

6.1	Overview	6-2
6.2	Setting Network Port Attributes	6-4
6.3	Setting the Maximum Frame Length.....	6-6
6.4	Checking Settings and States	6-8

6.1 Overview

The Network ports are used to control traffic on the network (traffic control).

The Network port of this device can be equipped with the following SFP/SFP+: (See Note 1.)
SFP+ 10GBASE-SR/10GBASE-LR is available only in NF7601A PureFlow WSX.

- SFP+ 10GBASE-SR / 10GBASE-LR (LC connector)
- SFP 1000BASE-SX / 1000BASE-LX (LC connector)
- SFP 1000BASE-T (RJ-45 / Auto-MDIX)

The following settings are available for the Network ports.

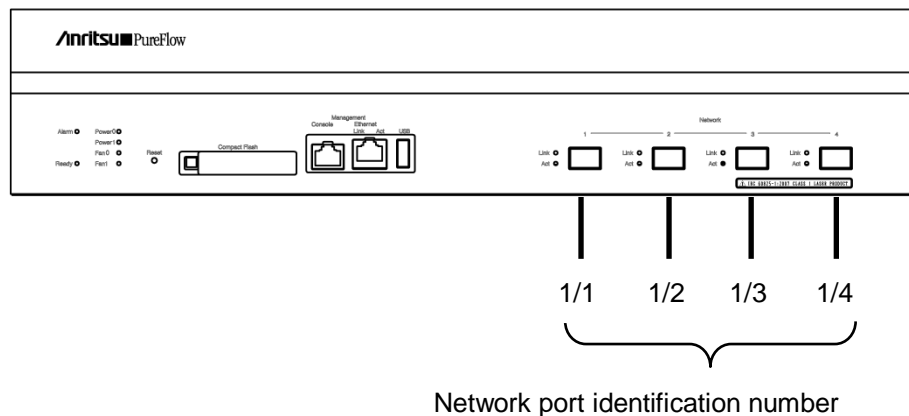
- Auto negotiation enable/disable (see Note 2,4)
- Flow control (auto, pause frame send/receive)
- Communication speed (10 Mbit/s, 100 Mbit/s, 1 Gbit/s) (see Note 3)
- Duplex mode (full, half) (see Note 3)
- Maximum frame length (2048 bytes, 10240 bytes) (see Note 5)

The application scope of the above settings differs depending on the SFP installed.

	10GBASE-SR/LR	1000BASE-SX/LX	1000BASE-T
AutoNegotiation	N/A	Enable/Disable	Enable/Disable
Communication speed:	10 G only	1 G only	10 M/100 M/1 G
Duplex mode	Full only	Full only	Full/Half
Flow control	Reception ON/OFF Transmission ON/OFF	Auto Reception ON/OFF Transmission ON/OFF	Auto Reception ON/OFF Transmission ON/OFF
Maximum frame length	2048/10240 [Byte]	2048/10240 [Byte]	2048/10240 [Byte]

To specify a Network port from CLI, specify it as the combination of a slot number and a port number. Specify “1” for the slot number of this device

The ports in the slot are numbered as 1/1, 1/2, 1/3, and 1/4 from the left. Therefore, the ID numbers of the Network ports are as shown below.



Note 1:

In this device, the device must be restarted when SFP+ and SFP are replaced after startup of this device in order to identify SFP+ or SFP mounted in the Network port during startup of this device. If SFT+ and SFP are replaced after the device starts, the Active/Link LED of the relevant Network port flashes to indicate the necessity of restarting.

Replacing between several SFP+ (10GBASE-SR, 10GBASE-LR) or between several SFP (1000BASE-SX/LX, 1000BASE-T) does not require restarting the device.

If neither SFP+ nor SFP is mounted in the Network port during startup of the device, operation of this Network Port is the same as that of the Network port including SFP+.

Note 2:

For 10GBASE-SR/LR, the AutoNegotiation configuration is not applied.

Note 3:

For 10GBASE-SR/LR, the communication speed is 10G, and duplex mode is Full only.

For 1000BASE-SX/LX, the communication speed is 1G, and duplex mode is Full regardless of the AutoNegotiation configuration.

The communication speed and duplex mode settings of 1000BASE-T SFP are effective only when auto negotiation is disabled. These settings are invalid when auto negotiation is enabled.

Note 4:

The maximum frame length setting is not applied to the system interface. The maximum frame length of the system interface is fixed to 1518 bytes.

Note 5:

If the link status of the "show port" command is half duplex, check that AutoNegotiation, communication speed, and duplex mode setting are suitable for the connected device.

6.2 Setting Network Port Attributes

When the 1000BASE-T SFP is used and auto negotiation is disabled, the operation attributes of the Network port such as communication speed or the duplex mode can be changed from CLI. Normally, these Network port attributes are automatically set to the appropriate operation mode by auto negotiation. If the destination switch or node does not support auto negotiation, you need to set the communication speed or the duplex mode of the Network port manually. If auto negotiation is enabled for the communicating device, enable auto negotiation for this device. If auto negotiation is disabled (manual setting) for one side and is enabled for the other side, normal connection cannot be established.

set port autonegotiation <slot/port> {enable disable}	Enables and disables auto negotiation of the Network port. The default value is enable.
set port speed <slot/port> {10M 100M 1G}	Specifies the communication speed of the Network port. This is the communication speed setting when auto negotiation is disabled. This setting is invalid when auto negotiation is enabled. The default is 1G. Note: Enable auto negotiation when using 1 Gbit/s communication on 1000BASE-T.
set port duplex <slot/port> {full half}	Specifies the duplex mode of the Network port. This is the duplex mode setting when auto negotiation is disabled. This setting is invalid when auto negotiation is enabled. The default value is full.

To disable auto negotiation, set the communication speed to 100 Mbit/s, set duplex mode to full for Network port 1/2, and execute the following commands:

```
PureFlow(A)> set port autonegotiation 1/2 disable
PureFlow(A)> set port speed 1/2 100M
PureFlow(A)> set port duplex 1/2 full
```

(Note)

If the port link status is the half duplex, the frame may be discarded due to frame collision. Check the link status by the “show port” command.

For either type of SFP+/SFP, full control of the Network port can be changed via CLI.

<pre>set port flow_control <slot/port> auto set port flow_control <slot/port> {recv send} {on off}</pre>	<p>Specifies the flow control of the Network port. The default setting is auto.</p> <p>If auto is specified, the flow control works as follows by the port type.</p> <p>If the port type is 1000BASE-T or 1000BASE-X: Pause frame reception and transmission is determined by AutoNegotiation.</p> <p>If AutoNegotiation is disabled, both reception and transmission are enabled.</p> <p>If the port type is 10GBASE-R: Both reception and transmission are enabled.</p>
--------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Also, to set flow control of Network port 1/2 so that no pause frame is sent or received, execute the following commands:

```
PureFlow(A)> set port flow_control 1/2 recv off
PureFlow(A)> set port flow_control 1/2 send off
PureFlow(A)>
```

6.3 Setting the Maximum Frame Length

The maximum frame length that can be transferred by the Network port can be changed by using the CLI command. Generally, MTU (Maximum Transmission Unit) means the payload length that excludes the header or FCS. In this command, the entire frame length that includes the Ethernet header and FCS are specified. The actual MTU is "this setting value + 4" bytes and "this setting value + 8" bytes in the cases of the frame with the VLAN Tag and the frame with duplex VLAN Tag, respectively. The maximum frame length is the setting value that is common to all of the Network ports.

set port mtu {2048 10240}	Set the maximum frame length of the Network port. The default value is 2048 Byte.
-----------------------------	--------------------------------------------------------------------------------------

It is necessary to restart the device in order to apply the setting change of the maximum frame length. To set the maximum frame length to 10240 bytes, execute the following commands:

```
PureFlow(A)> set port mtu 10240
Warning
This configuration change will be take effect on next boot.
Please save the system configuration and reboot the system.
If changed to 10240, some scenario parameters will be rounded as below.
bandwidth minimum      10k -> 50k
bandwidth resolution   1k -> 5k
buffer size minimum    2k -> 11k
If changed to 2048, channel mtu specified larger than 2048 will be rounded.

Do you wish to save the system configuration into the flash memory (y/n)? y

Done

Rebooting the system, ok (y/n)? y
```

Executing the command displays a prompt to check whether to save the configuration along with the message indicating the necessity of restart-up as well as a warning message relating to the scenario parameter setting range. Enter "y" to save the configuration. Next, the prompt for rebooting the device appears. Enter "y" and reboot the device. The setting change to 10240 bytes is applied after restarting the device.

Note 1:

This setting value changes the setting range and setting unit that are effective in the scenario parameters described below:

When the already-registered scenario parameter is out of range due to the change of the maximum frame length, parameter values are rounded to those within the range. Or in the case of additional registration, a warning message indicating that the rounding process is applied is displayed. In any case, the traffic control is executed using the rounded value.

Scenario parameter		Maximum frame length (Network port)	
		2048[Byte]	10240[Byte]
Minimum bandwidth	Setting range	10 k[bit/s] to 10G[bit/s] and 0	50k[bit/s] to 10G[bit/s] and 0
	Setting unit	1k[bit/s]	5k[bit/s]
Maximum bandwidth	Setting range	10k[bit/s] to 10G[bit/s]	50k[bit/s] to 10G[bit/s]
	Setting unit	1k[bit/s]	5k[bit/s]
Input burst length	Setting range	2 k[Byte] to 1G[Byte]	11k[Byte] to 1G[Byte]
	Setting unit	1k[Byte]	1k[Byte]

Note 2:

This setting value changes the setting range that is effective in the channel parameters described below:

When the already-registered channel parameter is out of range due to the change of the maximum frame length to 2048 bytes, parameter values are rounded to those within the range.

Channel parameter		Maximum frame length (Network port)	
		2048[Byte]	10240[Byte]
MTU	Setting range	300 to 10200[Byte] When the channel MTU is set the value greater than 2008, it is rounded to the default value of 1488 [Byte].	300 to 10200[Byte]

(Note 3)

This setting value changes the setting range that is effective in the peak burst size described below:

When the already-registered peak burst size is out of range due to the change of the maximum frame length to 2048 bytes, it is rounded to those within the range.

Peak burst size		Maximum frame length (Network port)	
		2048[Byte]	10240[Byte]
Peak burst size	Setting range	1536 to 9216[Byte] When the peak burst size is set the value greater than 2048, it is rounded to the default value of 1536 [Byte].	7680 to 46080[Byte]

6.4 Checking Settings and States

To check the settings specified by the setting commands and the current operation state of the Network port, use the “show port” command.

```
PureFlow(A)> show port
Port      Type           Status      Link      Autonego    Speed  Duplex
----      -
1/1       10GBASE-R     Enabled     Up        -----    10G    Full
1/2       10GBASE-R     Enabled     Up        -----    10G    Full
1/3       1000BASE-T    Enabled     Up        Enabled     100M   Full
1/4       1000BASE-T    Enabled     Up        Enabled     100M   Full
system    1000BASE-T    Enabled     Up        Enabled     100M   Full
PureFlow(A)>
```

The “show port” command allows you to check the state of all the Network ports mounted. To check more detailed information, specify the Network port ID in the command argument.

```
PureFlow> show port 1/1

Slot/Port      : 1/1
Port type      : 10GBASE-R
Admin status   : Enabled
Oper status    : Up
Auto negotiation : -----
Admin speed    : 1G
Oper speed     : 10G
Admin duplex   : Full
Oper duplex    : Full
Tx Flow control : Auto
Rx Flow control : Auto
Admin MTU      : 2048
Oper MTU       : 2048
PureFlow>
```

To check the statistics information of the Network port, use the “show counter” command. The counter length displayed in this command is 32 bits.

```
PureFlow(A)> show counter
Port          Rcv Octets    Rcv Packets   Trs Octets    Trs Packets
-----
1/1           57566366     14194297      0             0
1/2           0            0             59383412     14195494
1/3           57566366     14194297      0             0
1/4           0            0             59383412     14195494
system        58368        152           85424         152

Port          Rcv Broad    Rcv Multi     Trs Broad     Trs Multi
-----
1/1           10000        14208097      0             0
1/2           0            0             10000         14209615
1/3           10000        14208097      0             0
1/4           0            0             10000         14209615
system        5            0             10            0

Port          Err Packets   Collision     Discard
-----
1/1           0             0             0
1/2           0             0             0
1/3           0             0             0
1/4           0             0             0
system        N/A           N/A           N/A
```

You can also view detailed information by specifying the Network port ID in the command argument. The counter length that is displayed by this command is 64 bits. Be careful that the value different from that shown in the 64-bit counter of the "show counter <slot/port>" command appears if the 32-bit counter of the "show counter" command has wrapped around.

```
PureFlow(A)> show counter 1/1
Rcv Packets          14194297
Rcv Broad            10000
Rcv Multi            14208097
Rcv Octets           57566366
Rcv Rate              16 [kbps]
Trs Packets          0
Trs Broad            0
Trs Multi            0
Trs Octets           0
Trs Rate              0 [kbps]
Collision            0
Drop                 0
Discard              0
Error Packets        0
  CRC Align Error    0
  Undersize Packet   0
  Oversize Packet    0
  Fragments          0
  Jabbers            0
```

(Blank page)

Chapter 7 System Interface Settings

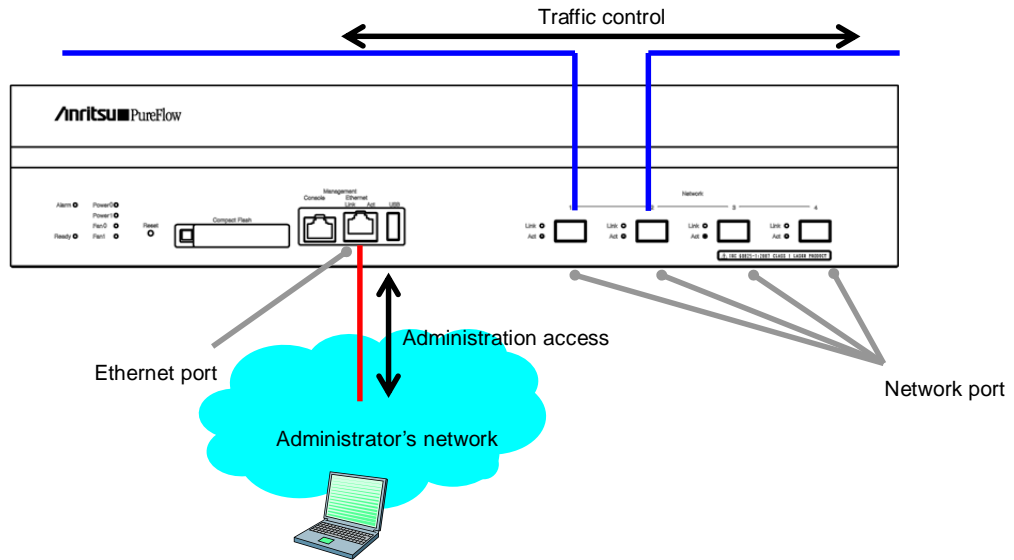
This chapter describes how to set up the system interface of this device.

7.1	Overview	7-2
7.2	System Interface Communication.....	7-3
7.3	System Interface Filter	7-8
7.4	Configuration Examples	7-9
7.5	Checking Settings and States	7-11

7.1 Overview

The system interface is an IP network interface for administrators to perform remote access to this device over the network. To control this device remotely, you can use methods such as Telnet and SNMP for setting and state monitoring of this device.

You can locate the administrator's terminal at the administrator's network other than the network that is used to control traffic (I/O from the Network port) and control the terminal via the Ethernet port.



7.2 System Interface Communication

Frame communication to the system interface without VLAN Tag can be performed. The filter feature can be used to restrict communication to the system interface from an unspecified number of terminals.

System interface communication supports simultaneous use of IPv4 and IPv6, but some features only support IPv4.

Feature	IPv4	IPv6
Telnet	✓	✓
SSH	✓	✓
RADIUS	✓	✓
TFTP	✓	✓
FTP	✓	✓
syslog	✓	✓
SNTP	✓	✓
SNMP	✓	–
PING	✓	✓
Telnet client	✓	✓
System interface filter	✓	✓
WebAPI	✓	✓
WebGUI	✓	✓
OpenFlow	✓	✓
NF7201A Monitoring Manager2	✓	–

If security settings such as a firewall are specified, change the settings to allow the following services to communicate.

Port number	TCP/UDP	Service name	Remarks
23	TCP	telnet	Telnet connection
22	TCP	ssh	SSH connection
1812	UDP	radius	RADIUS authentication
69	UDP	tftp	TFTP connection
21	TCP	ftp	FTP control
20	TCP	ftp	FTP data transfer
514	UDP	syslog	syslog transmission
123	UDP	ntp	SNTP client feature
161	UDP	snmp	SNMP monitoring
162	UDP	snmptrap	SNMP TRAP transmission
80	TCP	http	WebAPI, WebGUI
443	TCP	https	WebAPI, WebGUI
6653	TCP	openflow	OpenFlow connection (default value)
51967	TCP	–	Connection to the Monitoring Manager2

To set up the system interface, use the following commands:

set ip system <IP_address> netmask <netmask> [{up down}]	Sets the IP address of the system interface. The default IPv4 address is 192.168.1.1. The default subnet mask is 255.255.255.0. The default IPv6 address is ::192.168.1.1(::C0A8:101). The default prefix length is 64.
set ip system gateway <gateway>	Specifies the default gateway address of the system interface.
unset ip system gateway <gateway>	Clears the default gateway address of the system interface.
show ip system	Displays system interface information.

To set the IPv4 address (192.168.10.3), subnet mask (255.255.255.0), and default gateway (192.168.10.1) to the system interface, execute the following commands:

```
PureFlow(A)> set ip system 192.168.10.3 netmask 255.255.255.0 up
PureFlow(A)> set ip system gateway 192.168.10.1
```

Set the IPv6 address in the same manner as IPv4. Execute the following commands to set the IPv6 address (2001:DB8::1), the prefix length (32), and the default gateway (2001:DB8::FE) to the system interface.

For netmask of the set ip system command, specify the IPv6 prefix length:

```
PureFlow(A)> set ip system 2001:db8::1 netmask 32 up
PureFlow(A)> set ip system gateway 2001::db8:fe
```

The system interface also allows you to perform a communication check of the network by using the following commands.

ping <IP_address>	Sends a ICMP ECHO_REQUEST packet to the specified IP address. (IPv4 / IPv6)
arp -a arp -d <IP_address>	Displays (-a) or deletes (-d) the content of the ARP entry. (IPv4 only)
delete ndp neighbor <IP_address>	Deletes an NDP entry. (IPv6 only)
show ndp neighbor	Displays the content of the NDP entry. (IPv6 only)

To perform a communication check with the IPv4 address 192.168.10.100, execute the following commands:

```
PureFlow(A)> ping 192.168.10.100
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=0.372 ms

--- 192.168.10.100 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.372/0.372/0.372/0.000 ms
PureFlow(A)> arp -a
IP address      MAC address      type
-----
192.168.10.3    00-00-91-01-11-23    permanent publish
192.168.10.100  00-00-91-01-23-45
PureFlow(A)>
```

When the communication check fails, the following is displayed. Check the system interface settings and network connection.

```
PureFlow(A)> ping 192.168.10.101
PING 192.168.10.101 (192.168.10.101) 56(84) bytes of data.

--- 192.168.10.101 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 100ms
PureFlow(A)>
```

To delete the ARP entry of the IPv4 address 192.168.10.101, execute the following commands:

```
PureFlow(A)> arp -d 192.168.10.100
PureFlow(A)> arp -a
IP address      MAC address      type
-----
192.168.10.3    00-00-91-01-11-23    permanent publish
PureFlow(A)>
```

To perform a communication check with the IPv6 address 2001:DB8::1, execute the following commands:

```
PureFlow(A)> ping 2001:db8::1
PING 2001:db8::1 (2001:db8::1) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=0.372 ms
```

```
--- 2001:db8::1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.372/0.372/0.372/0.000 ms
PureFlow(A)> show ndp neighbor
IP address      MAC address      type
-----
2001:db8::1     00-00-91-01-23-45  reachable
PureFlow(A)>
```

When the communication check fails, the following is displayed. Check the system interface settings and network connection.

```
PureFlow(A)> ping 2001:db8::10
PING 2001:db8::10 (2001:db8::10) 56(84) bytes of data.

--- 2001:db8::10 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 100ms
PureFlow(A)>
```

To delete the NDP entry of the IPv4 address 2001:db8::10, execute the following commands:

```
PureFlow(A)> delete ndp neighbor 2001:db8::10
PureFlow(A)> show ndp neighbor
IP address      MAC address      type
-----
PureFlow(A)>
```

7.3 System Interface Filter

You can permit or deny communication to the system interface in units of hosts, etc.

You can define rules to identify communication to the system interface by using system filters. Define filters by using the following fields of the IP packet or a combination of them.

- Source IP address
- Destination IP address
- Protocol number
- Source port number (Sport)
- Destination port number (Dport)

Note:

A ToS value can be specified but filtering based on ToS values is not supported. Command including the tos specification can be accepted, however, the contents of the tos specification cannot be reflected in the filter operation.

To set a system interface filter, use the following commands:

add ip system filter	Sets a system interface filter.
delete ip system filter	Deletes a system interface filter.
show ip system	Displays system interface information.

To set the IPv4 address 192.168.10.3 and subnet mask 255.255.255.0 to the system interface to allow access to the device only from the PC with the IPv4 address 192.168.10.100, execute the following commands:

```
PureFlow(A)> set ip system 192.168.10.3 netmask 255.255.255.0 up
PureFlow(A)> set ip system gateway 192.168.10.1
PureFlow(A)> add ip system filter 20 sip 192.168.10.100 permit
PureFlow(A)> add ip system filter 30 deny
```

To cancel all the system interface filters, execute the following command:

```
PureFlow(A)> delete ip system filter all
```

To cancel system interface filter 30, execute the following command:

```
PureFlow(A)> delete ip system filter 30
```

Caution:

Be careful when setting a system interface filter.

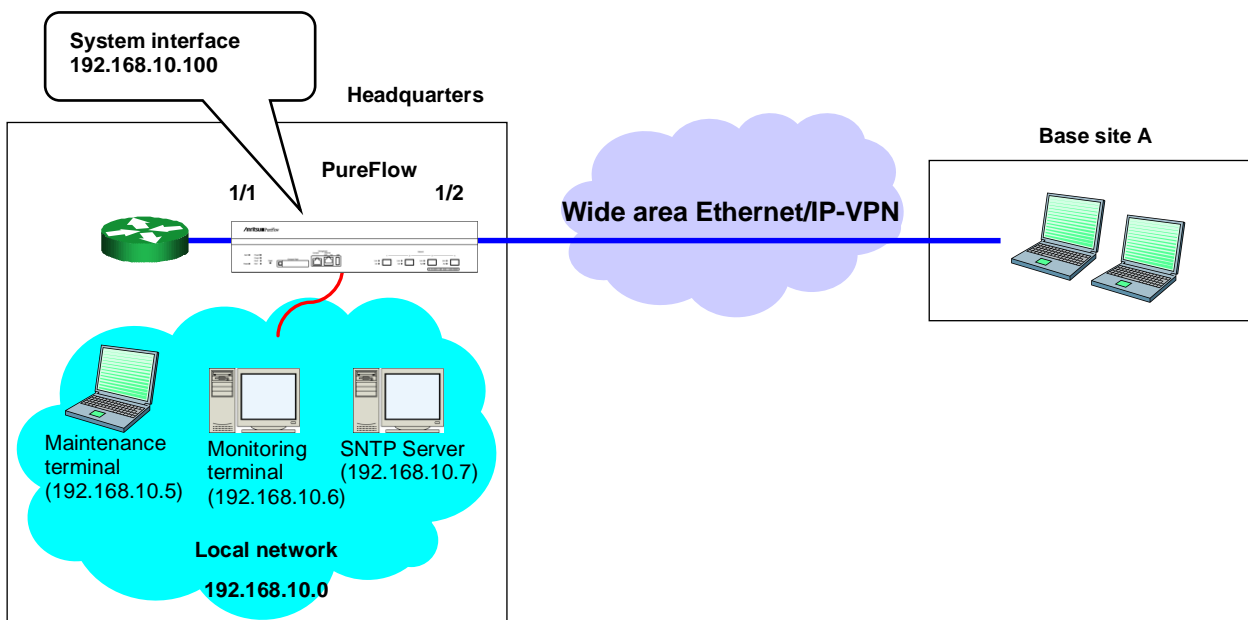
To enable the filter, set permit first, and set deny after that. To delete the filter, delete deny first, and then delete permit. Or delete all by using the “delete ip system filter all” command.

7.4 Configuration Examples

This section shows configuration examples of remote maintenance and monitoring in the following network environments.

Case 1 Performing maintenance and monitoring from the local network via the Ethernet port

- The local network within the headquarters is 192.168.10.0/255.255.255.0.
- The IPv4 address of the system interface is 192.168.10.100, and the subnet mask is 255.255.255.0.
- The default gateway address of the system interface is 192.168.10.1.
- The IPv4 address of the maintenance terminal (CLI, download/upload) is 192.168.10.5.
- The IPv4 address of the monitoring terminal (SNMP, Syslog) is 192.168.10.6.
- The IPv4 address of the Sntp server is 192.168.10.7.



Execute the following commands.

System interface setting:

```
PureFlow(A)> set ip system 192.168.10.100 netmask 255.255.255.0 up
```

```
PureFlow(A)> set ip system gateway 192.168.10.1
```

SNMP host setting:

```
PureFlow(A)> add snmp view All iso included
```

```
PureFlow(A)> add snmp community honsya_system_management view All
```

```
PureFlow(A)> add snmp host 192.168.10.6 version v2c
community honsya_system_management trap
```

Syslog host setting:

```
PureFlow(A)> add syslog host 192.168.10.6
```

```
PureFlow(A)> set syslog host enable
```

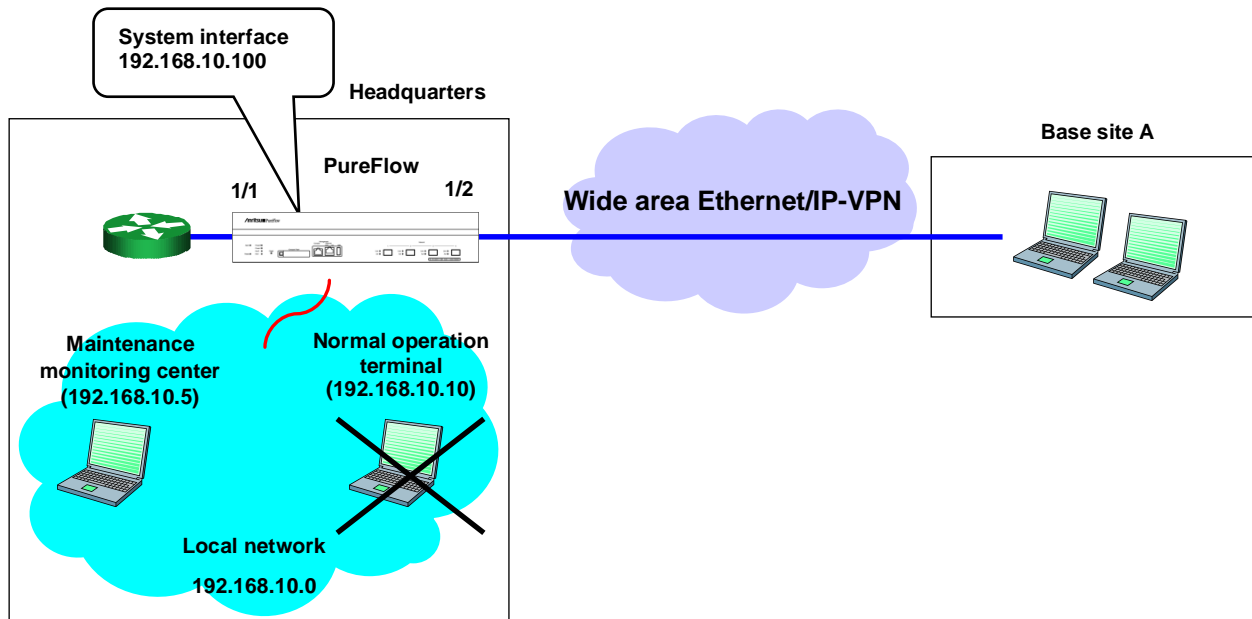
SNTP server setting:

```
PureFlow(A)> set sntp server 192.168.10.7
```

```
PureFlow(A)> set sntp enable
```

Case 2 Performing maintenance and monitoring from the specified terminal via the Ethernet port. No monitoring from unidentified terminals.

- The local network within the headquarters is 192.168.10.0/255.255.255.0.
- The IPv4 address of the system interface is 192.168.10.100, and the subnet mask is 255.255.255.0.
- The default gateway address of the system interface is 192.168.10.1.
- The IPv4 address of the maintenance terminal (CLI, download/upload) is 192.168.10.5.
- The IPv4 address of the normal operation terminal is 192.168.10.10.



Execute the following commands.

System interface setting:

```
PureFlow(A)> set ip system 192.168.10.100 netmask 255.255.255.0 up
```

```
PureFlow(A)> set ip system gateway 192.168.10.1
```

System interface filter setting:

```
PureFlow(A)> add ip system filter 10 sip 192.168.10.5 permit
```

```
PureFlow(A)> add ip system filter 20 deny
```

7.5 Checking Settings and States

To check the settings configured by the setting commands of the system interface, use the “show ip system” command.

```
PureFlow(A)> show ip system
Status                : Up
IP Address             : 192.168.10.3
Netmask               : 255.255.255.0
Broadcast             : 192.168.10.255
Default Gateway       : 192.168.10.1
IPv6 Address          : 2001:DB8::1
Prefix                : 32
Default Gateway       : 2001:DB8::FE
Port                  : Network (1/2)
VID                   : 20
TPID                  : 0x8100
Inner-VID             : none
Inner-TPID           : ----
```

Number of system filter entries: 0

```
PureFlow(A)>
```

To check the statistics information of the system interface, use the “show counter” command. The counter length displayed in this command is 32 bits.

```
PureFlow(A)> show counter
```

Port	Rcv Octets	Rcv Packets	Trs Octets	Trs Packets
1/1	57566366	14194297	0	0
1/2	0	0	59383412	14195494
1/3	57566366	14194297	0	0
1/4	0	0	59383412	14195494
system	58368	152	85424	152

Port	Rcv Broad	Rcv Multi	Trs Broad	Trs Multi
1/1	10000	14208097	0	0
1/2	0	0	10000	14209615
1/3	10000	14208097	0	0
1/4	0	0	10000	14209615
system	N/A	N/A	N/A	N/A

Port	Err Packets	Collision	Discard
1/1	0	0	0
1/2	0	0	0
1/3	0	0	0
1/4	0	0	0
system	N/A	N/A	N/A

You can also display detailed information by specifying the system interface in the command argument. The counter length of Rcv Packets, Rcv Octets, Trs Packets, and Trs Octets that are displayed by this command is 64 bits. Be careful that the value different from that shown in the 64-bit counter of the "show counter system" command appears if the 32-bit counter of the "show counter" command has wrapped around.

```
PureFlow(A)> show counter system
Rcv Packets                152
Rcv Broad                  N/A
Rcv Multi                  N/A
Rcv Octets                 58368
Rcv Rate                   N/A
Trs Packets                152
Trs Broad                  N/A
Trs Multi                  N/A
Trs Octets                 85424
Trs Rate                   N/A
Collision                  N/A
Drop                       N/A
Discard                    N/A
Error Packets              N/A
    CRC Align Error        N/A
    Undersize Packet        N/A
    Oversize Packet        N/A
    Fragments               N/A
    Jabbers                  N/A
```


Chapter 8 Traffic Control

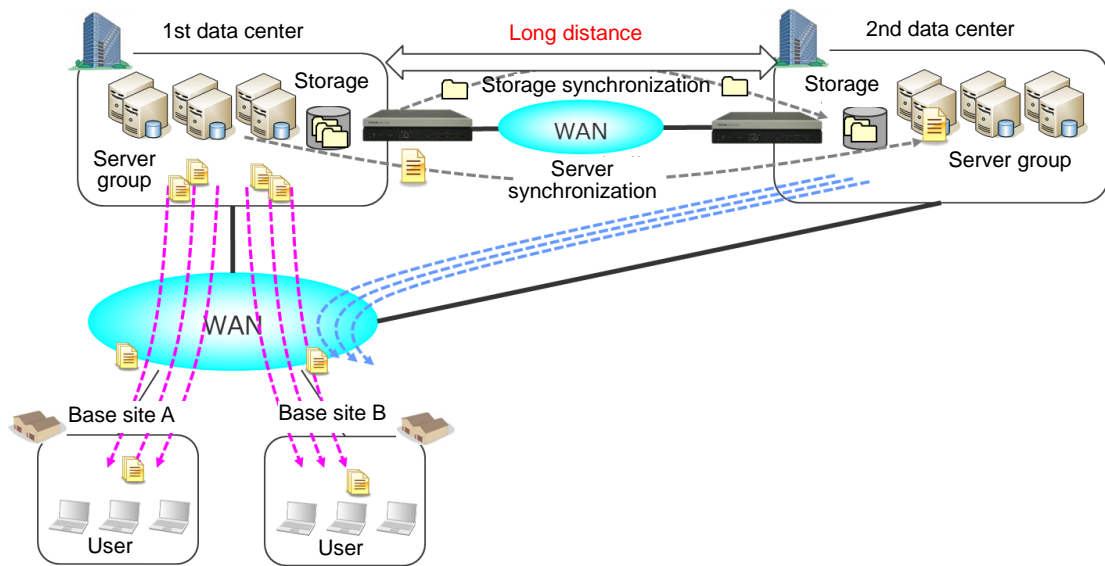
This chapter describes the traffic control feature and settings.

8.1	Overview	8-2
8.2	Traffic Acceleration	8-3
8.3	Traffic Shaping.....	8-3
8.4	Application to Large-scale Network	8-4
8.5	Channel	8-5
8.6	Scenario	8-6
	8.6.1 Traffic Attribute.....	8-7
	8.6.2 Filter	8-8
8.7	Hierarchical Scenario.....	8-10
	8.7.1 Hierarchical relation of filters	8-11
	8.7.2 Relationship between filters and scenarios ...	8-12
	8.7.3 Rule list.....	8-14
8.8	Acceleration Tunnel	8-15
8.9	Setting Procedure	8-17
8.10	How to Set a Rule List.....	8-37
8.11	Channel interface communication	8-39
8.12	Application Acceleration Function.....	8-41
	8.12.1 SMB protocol acceleration function	8-42
	8.12.2 Precautions for SMB protocol acceleration function.....	8-45
8.13	Configuration Example.....	8-46
8.14	Advanced Settings	8-54
	8.14.1 Flow.....	8-55
	8.14.2 Queues	8-56
	8.14.3 Communication gap mode.....	8-64
	8.14.4 Traffic acceleration bypass.....	8-66
	8.14.5 Traffic acceleration redundancy.....	8-70
	8.14.6 TCP-FEC function.....	8-77
	8.14.7 TCP congestion control function.....	8-80
	8.14.8 Remarking function	8-82
8.15	Address during the traffic acceleration.....	8-86

8.1 Overview

This device has a traffic acceleration function that enables high-speed data communication that is not affected by the long distance line delay, such as WAN. Currently, data centers are increasing demand for the centralized server/storage allocation model to reduce operational costs and enhance security. To securely recover server data in the event of a disaster, demand for remote transfer of backup data is increasing. However, as the transfer of TCP/IP uses a lot of data, communication decelerates due to line delay. The traffic acceleration function of this device prevents the TCP/IP data transfer deceleration affected by the line delay, and provides high-speed data communication.

Additionally, this device also has a traffic shaping function that smoothes the burst traffic simultaneously transmitted from multiple servers or clients, and prevents packets from being discarded by using the router or switches allocated in the network.



8.2 Traffic Acceleration

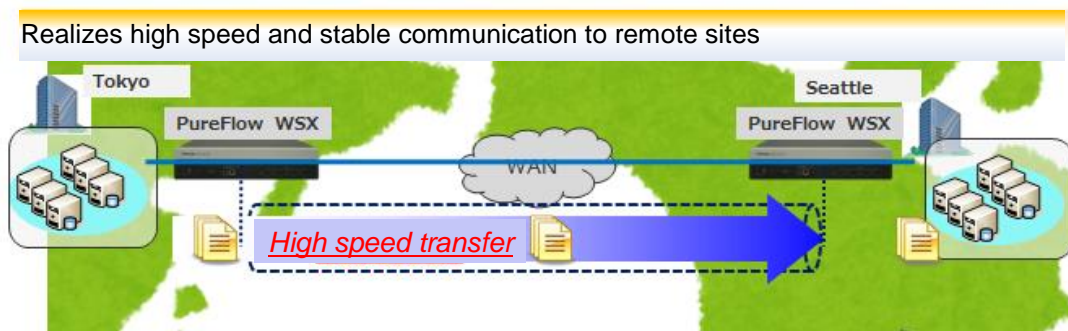
This function accelerates the TCP/IP data communication over a long distance. For the data transfer over a long distance, the transmission path is physically long, and many relay devices are allocated such as a router or switch. Therefore, the delay time increases between transferring the packet from the server on the transmission side and receiving packets by the clients on the receiving side. Data transfer of the TCP/IP protocol used by many servers and clients decelerates if the delay is increased. This device reduces the data transfer deceleration and enables high-speed data transfer even via the communication line that delays by several dozens of milliseconds.

In addition, this device enables high-speed data transfer by restraining data transfer speed reduction when using a line with a high packet-discarding rate in the WAN line and others due to the TCP-FEC function and TCP congestion control function.

For details about the TCP-FEC function, see "8.14.7 TCP-FEC function".

For details about the TCP congestion control function, see "8.14.8 TCP congestion control function".

The traffic acceleration of IP version supports IPv4 and IPv6.



8.3 Traffic Shaping

Additionally, this function smoothes the burst traffic simultaneously transmitted from multiple servers or clients, and prevents packets from being discarded by using the router or switches allocated in the network. This enables high-speed and stable TCP/IP communication.

The traffic shaping of IP version supports both IPv4 and IPv6.

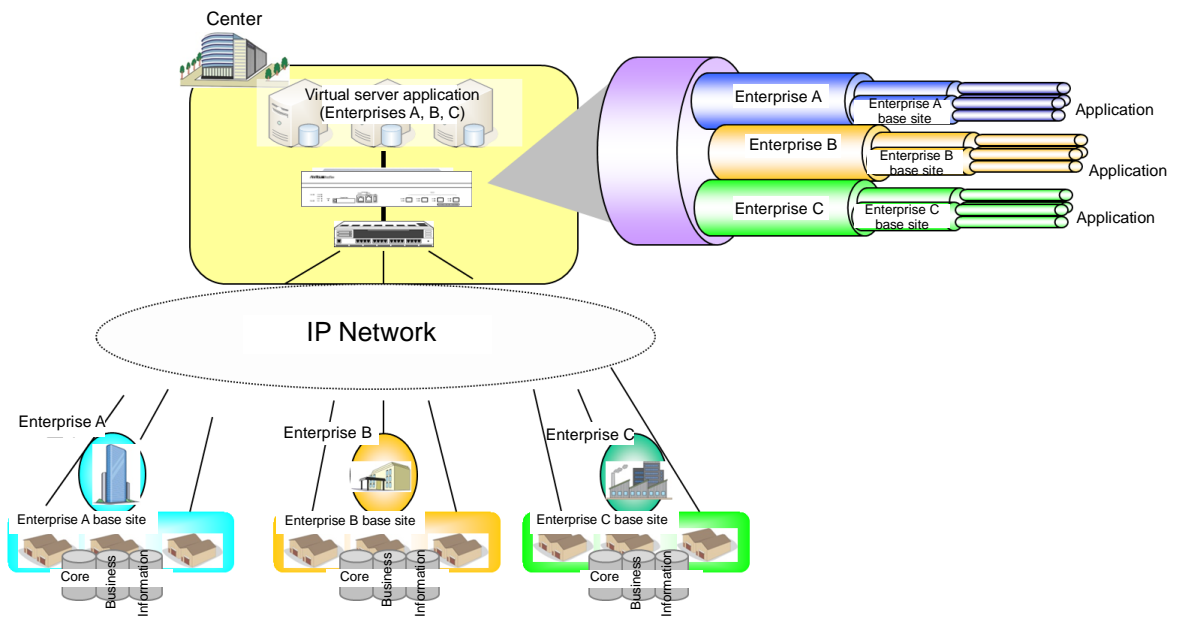
Smoothing burst traffic



8.4 Application to Large-scale Network

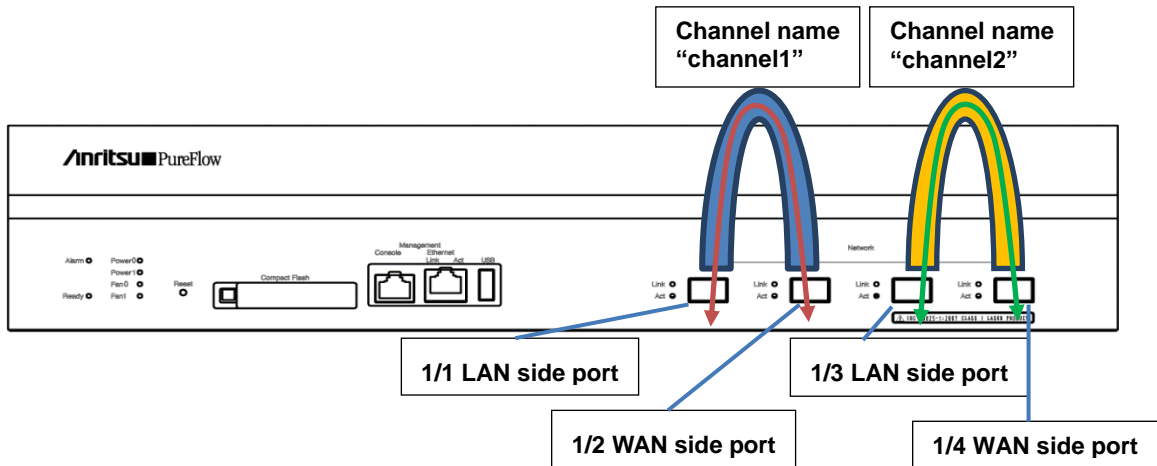
This device can be applied to a large-scale corporate core network that has a large number of centers and a large-scale network that provides cloud service to multiple business bases. The traffic can be grouped hierarchically and managed in a group unit for easy operation.

For example, group the traffic in a business base unit, break it up into a center unit, and then break it up in application (service) or user units. Traffic shaping and traffic acceleration can be performed for each group. The traffic can be hierarchically classified into the desired group such as a business base, application, or user to enable the traffic shaping or traffic acceleration.



8.5 Channel

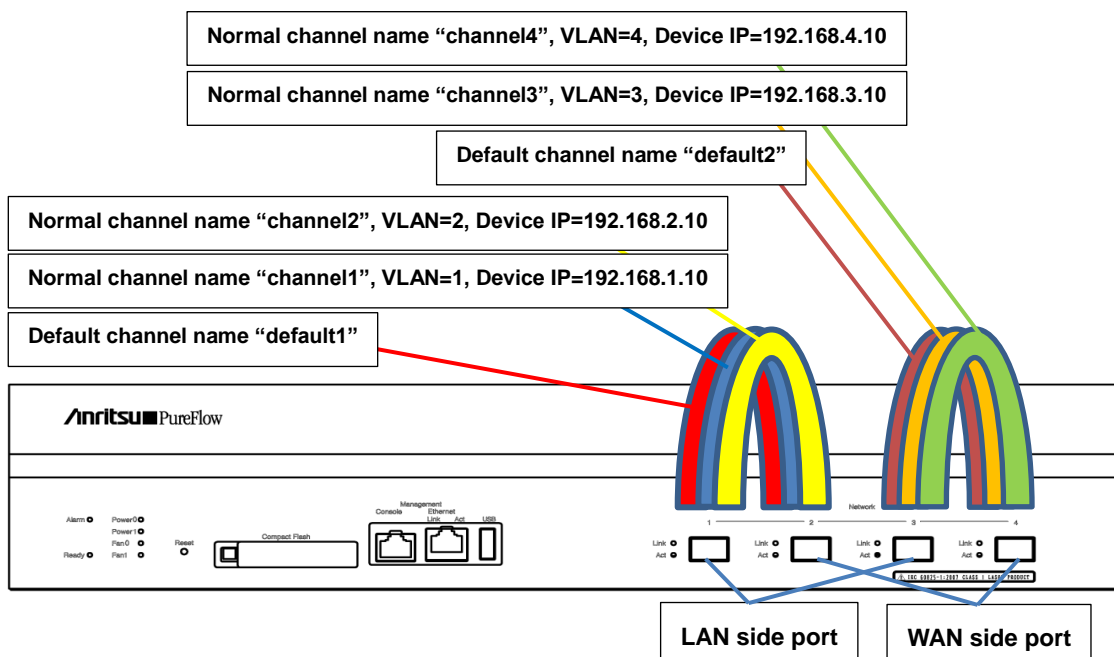
This device has 4 network ports and performs bridge operation between any 2 ports. A combination of 2 channels between which bridge operation is performed is called a “channel”. This device connects the network of the LAN side and WAN side by using channels. Therefore, it is required to specify the LAN side port and the WAN side port for channel.



Any 2 ports out of the 4 network ports can be combined as a channel. Unless there is any special reason, use 1/1 and 1/2 or 1/3 and 1/4 as the channel combination. Using other combinations (e.g. 1/1 and 1/3) may result in lower device performance.

For channels, set the device IP address of the IP network interface (channel interface) that is used for traffic acceleration. This device configures the acceleration tunnel to implement the TCP/IP data communication at high speed via this channel interface. For details about the acceleration tunnel, see "8.9 Acceleration Tunnel".

For the wide area Ethernet line in which the WAN line uses VLANs, each VLAN needs a separate channel and device IP address. There are two types of channels: a normal channel that transfers applicable flow to registered VLAN, and a default channel that transfers non-applicable flow to the normal channel.

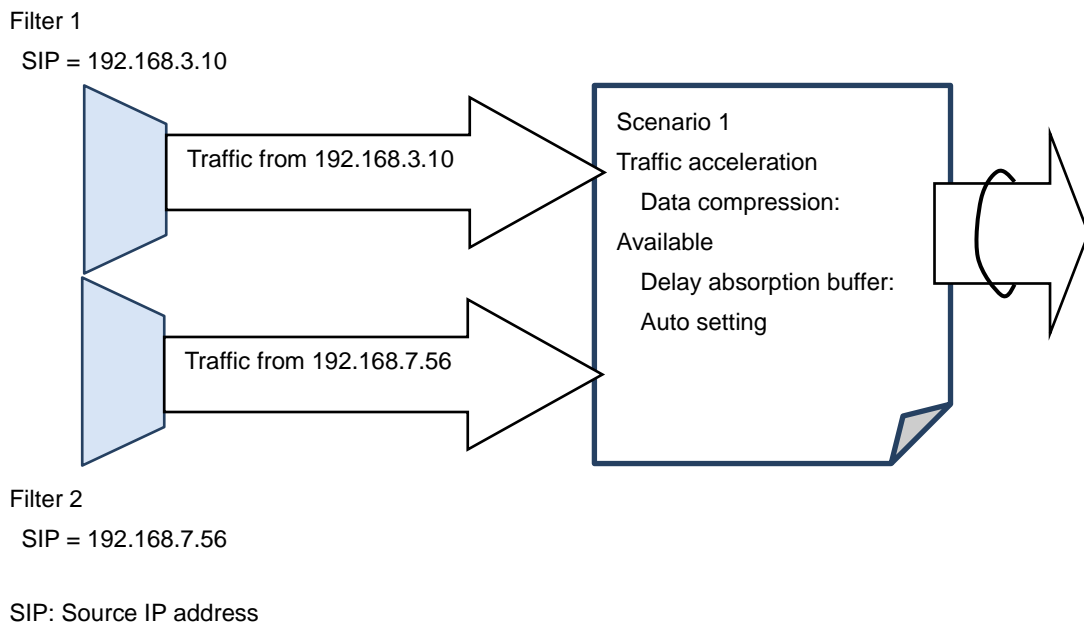


8.6 Scenario

This device specifies the control of the traffic that flows in the network in the unit that is referred to as a scenario. For the scenario, the filter that includes the descriptions on the traffic classification criteria and traffic attribute that specifies the control of the classified traffic are specified.

This device classifies the passing packets according to the filter rule, and groups the traffic. The grouped traffic is controlled due to the traffic attribute that is referred to as a scenario.

Multiple filters can be set for one scenario.



8.6.1 Traffic Attribute

The traffic attribute specifies the types (acceleration or shaping) of the traffic control for the traffic that flows in the network and parameters that control traffic.

The traffic attribute action has the following modes:

- (1) Acceleration mode (Wan-accel mode)
- (2) Aggregate queue mode (Aggregate mode)
- (3) Individual queue mode (Individual mode)
- (4) Discard mode for discarding packets (Discard mode)

The acceleration mode (Wan-accel mode) performs the traffic acceleration for the TCP traffic that matches the filter to accelerate the TCP communication. For traffic other than the TCP traffic, the received packet is directly transferred.

The Aggregate queue mode (Aggregate mode) controls the communication bandwidth as a group of the traffic that matches the filter.

The Individual queue mode (Individual mode) additionally classifies the traffic that matches the filter into each flow (the minimum unit of the traffic that can be identified in the device), and controls the communication bandwidth of each flow.

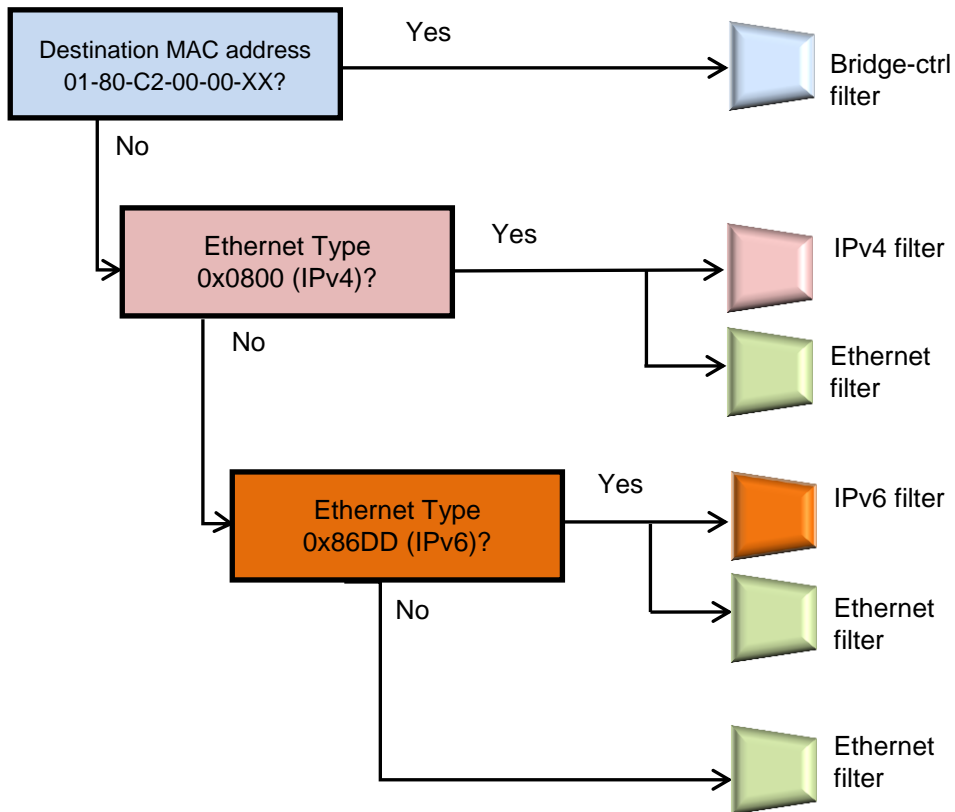
The Discard mode discards the traffic that matches the filter.

8.6.2 Filter

The criteria to classify the packet for each scenario are set in the filter. There are 3 types of filters: the Bridge-ctrl filter that classifies the Bridge-ctrl frames only; the Ethernet filter that classifies the length/type fields and VLAN Tag fields of the Ethernet header; and the IP filter that classifies the VLAN Tag fields, IP headers, and Protocol headers.

1. The Bridge-ctrl filter targets MAC addresses reserved for switch control such as the spanning tree protocol BPDU or link aggregation LACP.
For example, use this to prioritize BPDU or secure the bandwidth under the spanning tree environment.
Targeted MAC addresses are as follows:
 - Destination MAC addresses: 01-80-C2-00-00-00 to 01-80-C2-00-00-FF
2. The Ethernet filter targets the entire Ethernet frame.
Use this to classify packets by VLAN or packet type.
For example, specifying VLAN alone enables bandwidth control per VLAN.
Specifying Ethernet Type “0806” enables prioritization of ARP packets or securing of bandwidth.
3. The IP filter targets IP packets.
Use this to classify IP packets by IP packet field.
When classifying IP packets by IP filter, further classification is available using the following IP packet fields:
 - VLAN ID
 - Source IP address (SIP)
 - Destination IP address (DIP)
 - Protocol number
 - Source port number (Sport)
 - Destination port number (Dport)

For traffic classification by filter, the applicable filter type is fixed according to the packet. For frames with the MAC address 01-80-C2-00-00-XX, only the Bridge-ctrl filter is applied regardless of other fields' contents. Provided the MAC address is not 01-80-C2-00-00-XX, only the IPv4 filter and Ethernet filter are applied to packets with Ethernet Type 0x0800, and only the IPv6 filter and Ethernet filter are applied to packets with Ethernet Type 0x86DD. For packets other than the above, only the Ethernet filter is applied.

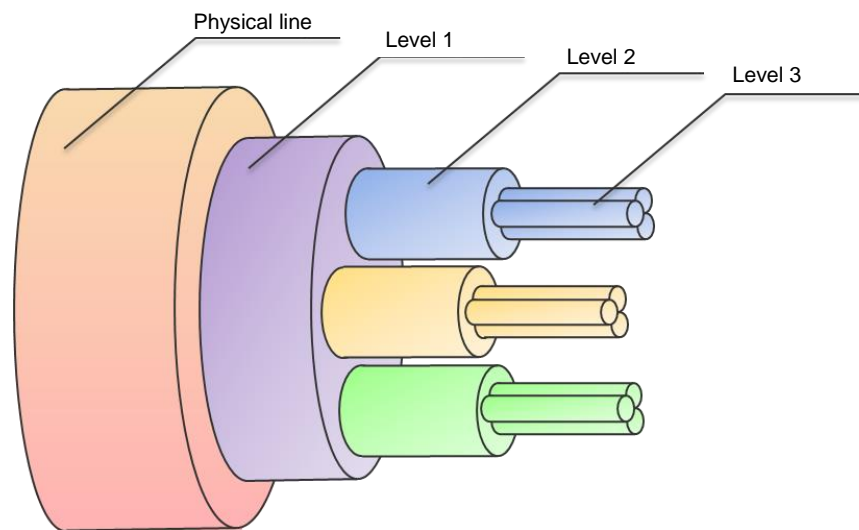


8.7 Hierarchical Scenario

This device can specify the scenarios hierarchically.

On the first level (Level 1), the physical line bandwidth is controlled (traffic shaping) at any bandwidth. On the second level (Level 2), the traffic of business bases and users is classified to enable the traffic acceleration or traffic shaping. By flowing the traffic into the virtual circuits of Level 2, the line bandwidth is divided into virtual circuits and separated bandwidths are allocated. On the third level (Level 3) and later, like the above, the bandwidth allocated to the upper level can be divided and controlled.

The following is a conceptual diagram of hierarchical scenario:



Level 1 (the first hierarchy):

The entire bandwidth of Level 1 can be shaped (traffic shaping).

Level 1 can comprise one or more Level 2 bandwidths.

Level 2 (the second hierarchy)

The bandwidth of Level 1 is classified and controlled.

The traffic acceleration and traffic shaping for the traffic can be performed.

Level 2 can comprise one or more Level 3 bandwidths.

Level 3 (the third hierarchy)

The bandwidth of Level 2 is divided and controlled.

The traffic acceleration and traffic shaping for the traffic can be performed.

Level 3 can comprise one or more Level 4 bandwidths.

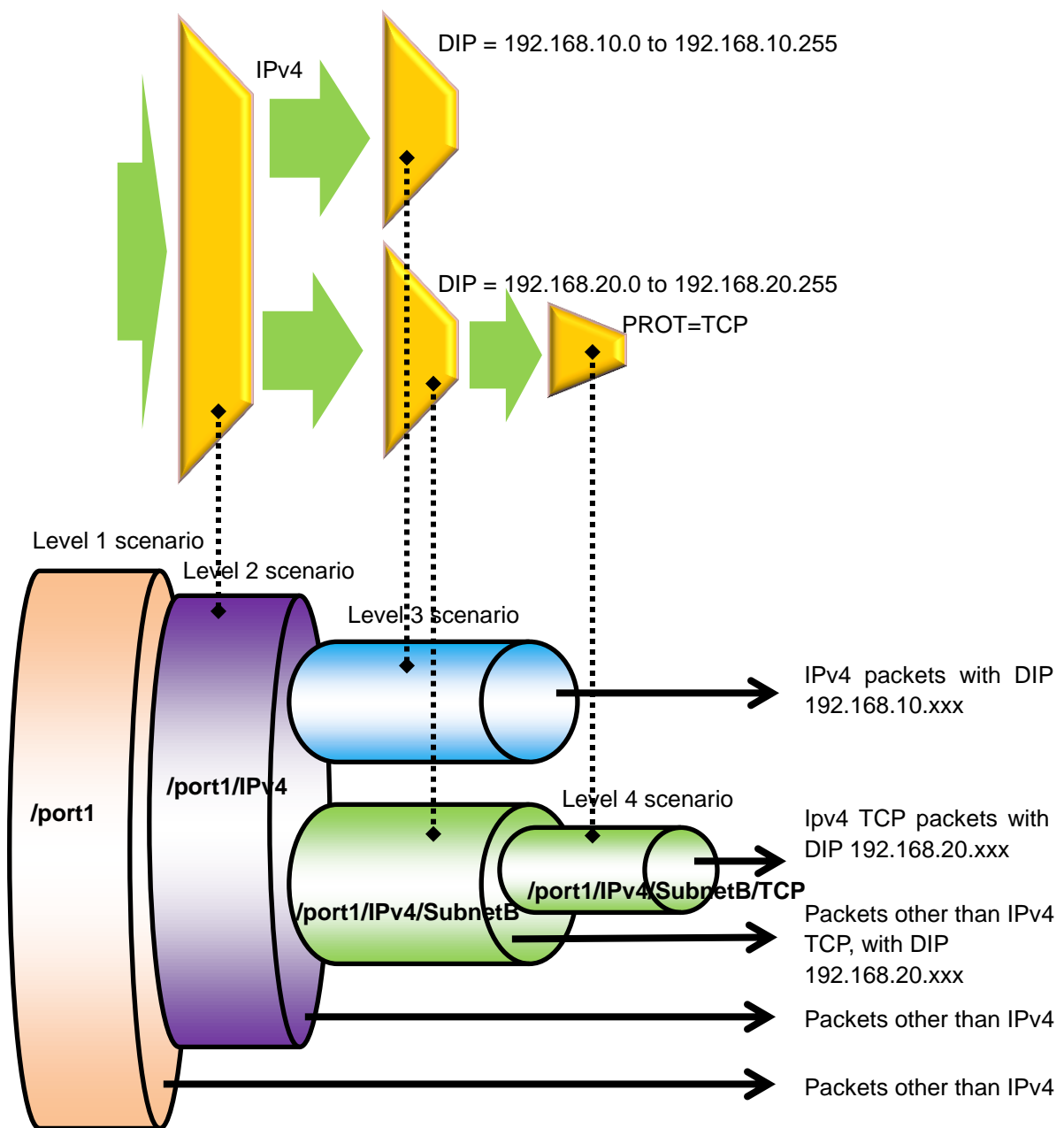
In the same way, the bandwidth can be divided and controlled up to Level 8.

8.7.1 Hierarchical relation of filters

Filters of each scenario inherit the filter criteria of the upper level scenario and classify packets hierarchically.

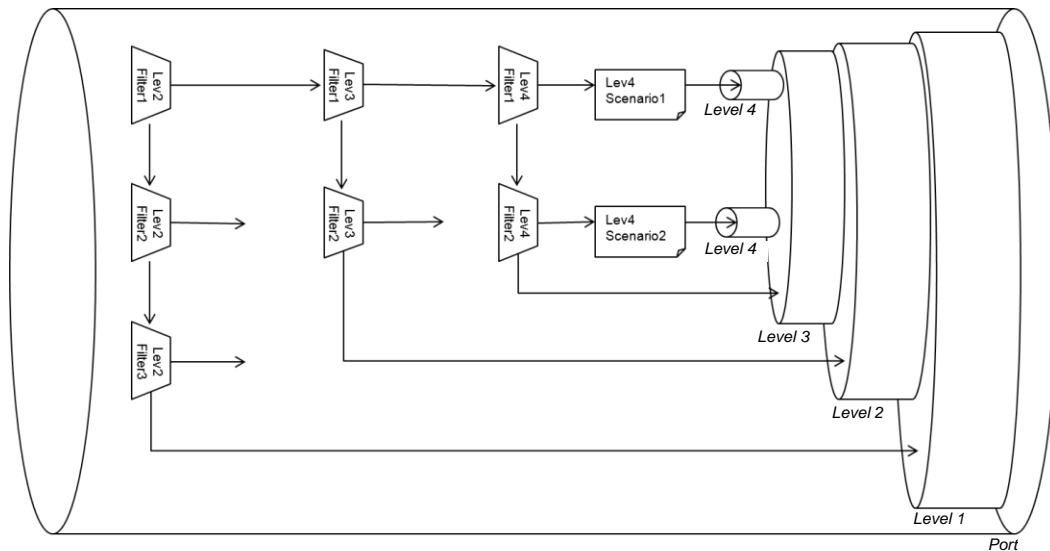
Traffic that matches both the upper level scenario filter criteria and the lower level scenario filter criteria is classified as lower level traffic. Traffic that matches the upper level scenario filter criteria but not the lower level scenario filter criteria is classified as upper level traffic, and is transmitted in an available bandwidth of the upper level scenario.

The following diagram is an example of classifying packets in a hierarchy. For the filter of the Level 2 scenario, specify IPv4 to classify packets into IPv4 packets and other packets. For the filter of the Level 3 scenario, specify the Subnet address to classify packets into Subnet A packets, Subnet B packets, and other Subnet IPv4 packets. For the filter of the Level 4 scenario, specify Protocol TCP to classify packets into Subnet B TCP packets and other packets.



8.7.2 Relationship between filters and scenarios

This device classifies packets flowing through the physical line by using filters to extract traffic. It performs traffic control transfer of extracted traffic according to traffic attributes such as bandwidth or buffer size.



The figure above is a conceptual diagram illustrating the relationship between the filter and scenario settings and the actual traffic control operation.

Bandwidth control from Level 1 to Level 8, and discard and transfer control by the filter setting are available.

Additionally, the traffic acceleration from Level 2 to Level 8 is available.

For the filter action, “aggregate”, “individual”, “discard” and “wan-accel” can be specified. Packets that match the filter rules follow the configured operation.

This device uses Level 2 filters in a priority order to verify whether received packets match the filter rules.

If the packet matches a Level 2 filter and operation of the Level 2 scenario associated with the filter is “aggregate”, the device transfers the packet according to the traffic attribute specified for the scenario. The device then uses Level 3 filters of the Level 3 scenario associated with that scenario in priority order to verify whether the packet matches the filter rules.

In the case of “individual”, the device transfers the packet according to the traffic attribute specified for the scenario. A scenario and a filter can be registered under the “individual” scenario, but filters at lower levels than the “individual” scenario do not work. Packets are not transferred for scenarios at lower levels than the “individual” scenario, and filters are disabled. In the case of “discard”, packets are discarded. A scenario and a filter can be registered under the “discard” scenario, but filters at lower levels than the “discard” scenario do not work. Packets are not transferred for scenarios at lower levels than the “discard” scenario, and filters are disabled.

In the case of “wan-accel”, the traffic acceleration is performed. A scenario and a filter can be registered under the “wan-accel” scenario, but filters at lower levels than the “wan-accel” scenario do not work. Packets are not transferred for scenarios at lower levels than the “wan-accel” scenario, and filters are disabled.

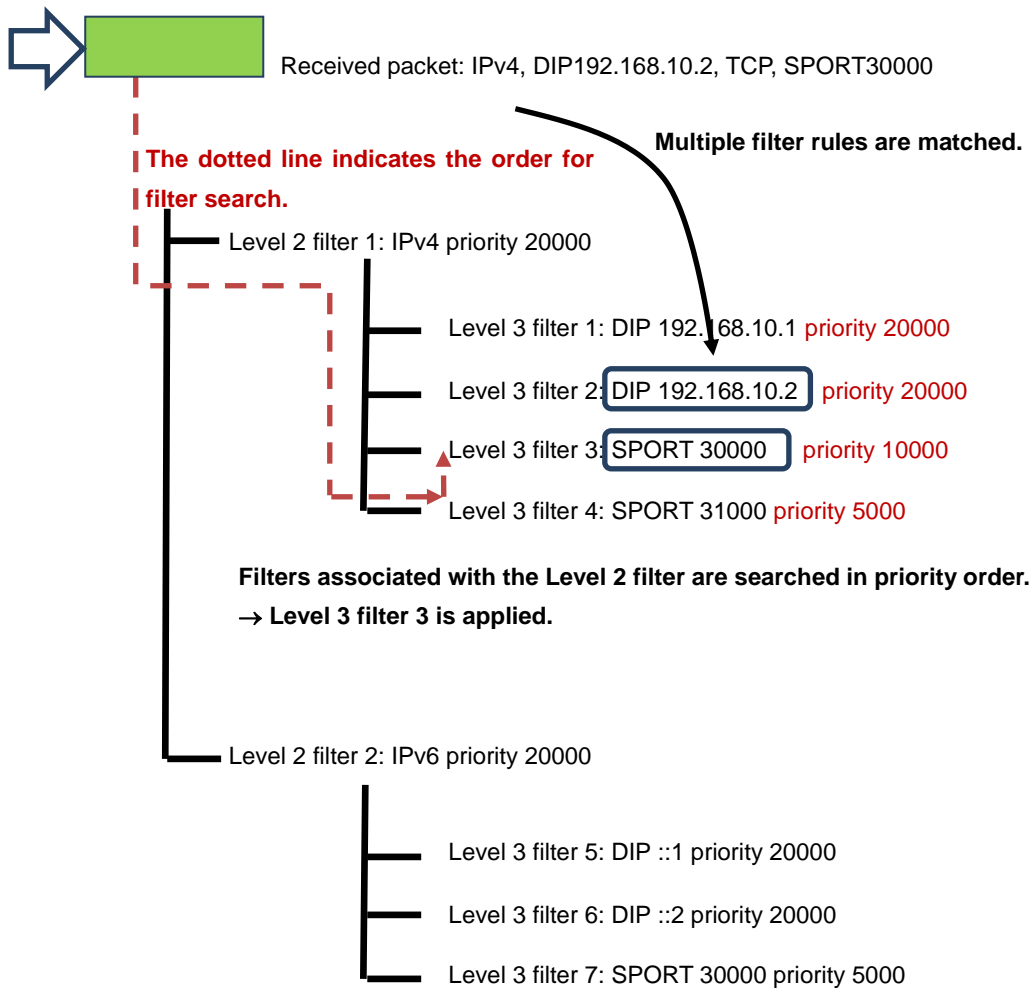
The registered scenario and filter at lower levels than the “wan-accel” scenario are disabled.

Filters at Level 3 to Level 8 work in the same way.

The Bridge-ctrl filter, Ethernet filter and IP filter can be specified. Use any character string to specify a filter name. For all filters, a total of 40000 filter rules can be created.

You can also assign priority to each filter rule.

If multiple filter rules among the same level filters associated with the scenario are matched, the filter to be applied is determined according to the filter priority. A smaller value means a higher filter priority.



If matched filter rules have the same priority, the filter to be applied is determined in any order. For the filter configuration where multiple filter rules are matched, it is recommended to adjust the priority to distinctively specify the filter to be applied. If the filter priority is omitted, 20000 is automatically applied.

8.7.3 Rule list

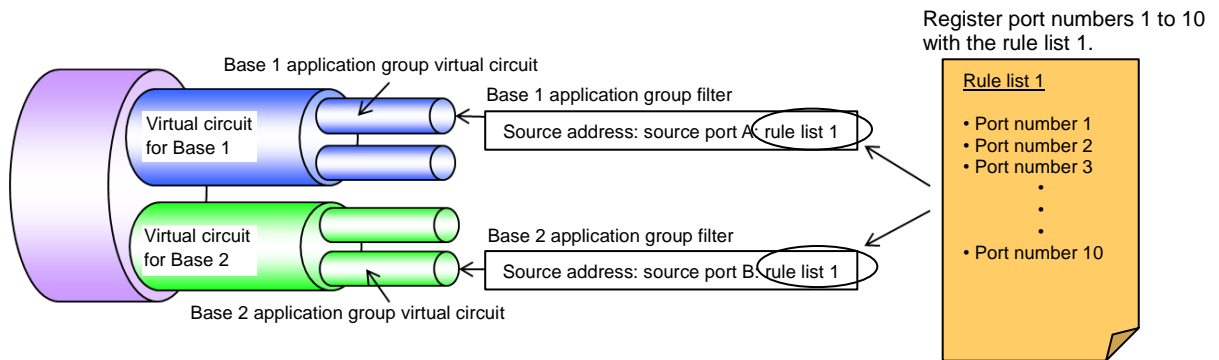
The rule list is a feature to group multiple classification conditions for traffic (IP address, port number, etc.). By using this feature, multiple classification conditions for traffic can be specified by a single rule list name.

To set a rule list as traffic classification conditions, specify the rule list name as an argument of the add filter command.

Traffic classification conditions that can be specified for the rule list are as follows:

1. IPv4 address : IP address and address mask
2. IPv6 address : IP address and address mask
3. L4 port number : Port number range

Rule lists can be specified for multiple filters repeatedly. Using rule lists reduces the number of filters and lines for configuration.

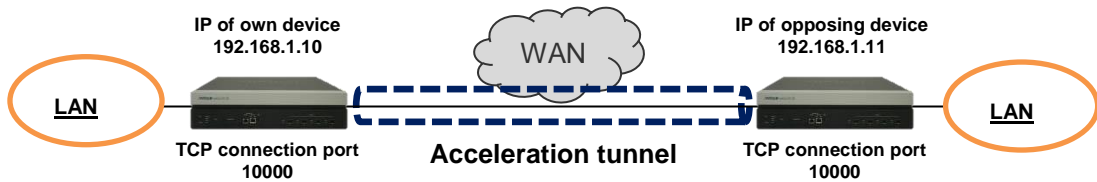


The above figure is a conceptual diagram illustrating the relationship between the rule list settings and the actual traffic control operation. In the figure, multiple TCP/UDP port numbers are registered to the rule list 1, and the list is used as sport parameters (source port number) for filter setting commands for the Base 1 application group virtual circuit and Base 2 application group virtual circuit.

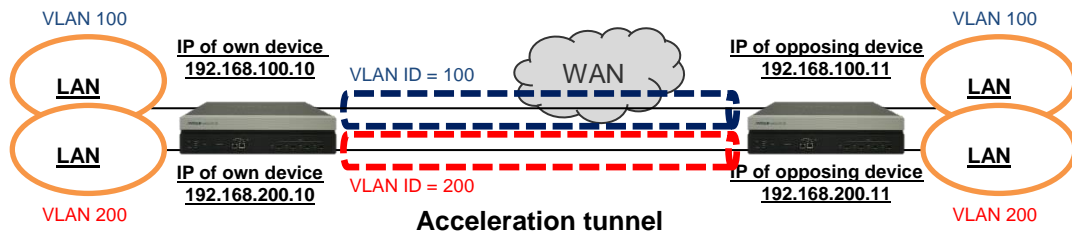
8.8 Acceleration Tunnel

This device performs traffic acceleration when the acceleration mode is selected in the traffic attribute of the scenario.

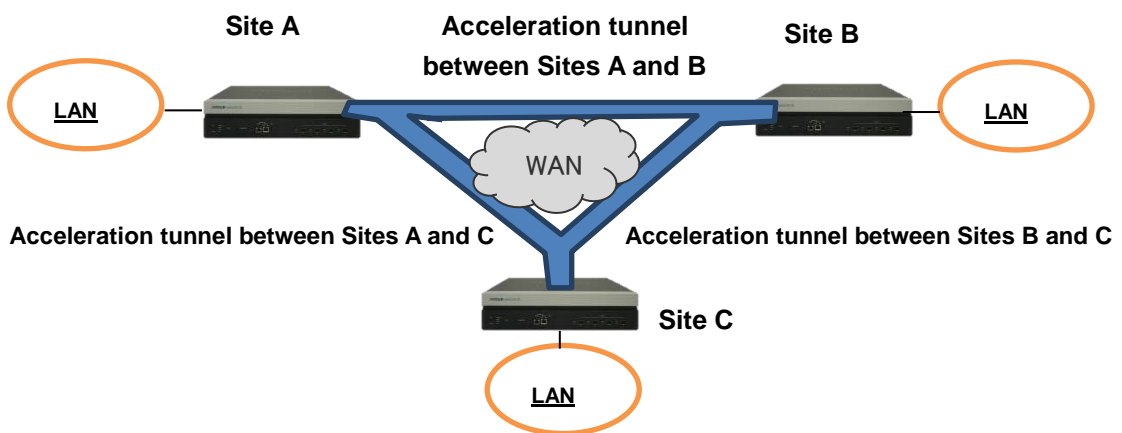
To perform traffic acceleration, an acceleration tunnel must be constructed between opposing devices via the WAN line. To construct an acceleration tunnel, specify the IP address/TCP connection port number/VLAN of the opposing device in the scenario. An acceleration tunnel is constructed between this opposing device and the IP addresses and VLAN of channel interface device that are set in own device.



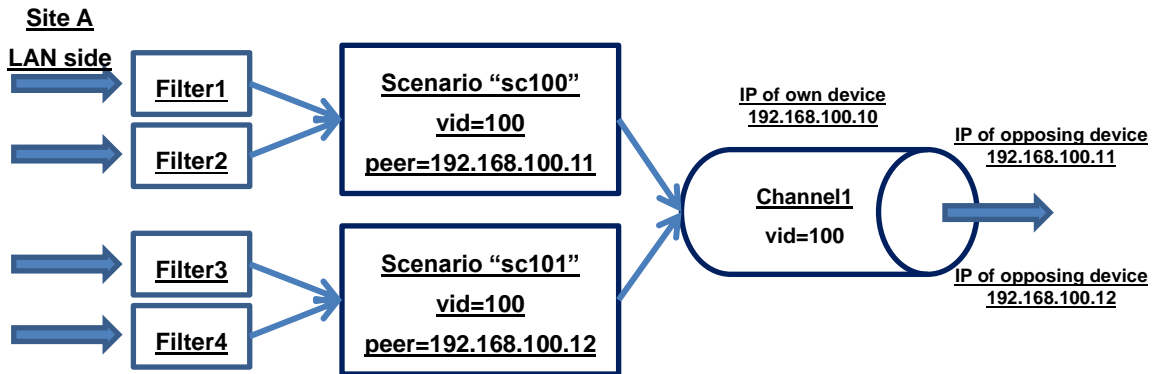
For the wide area Ethernet line in which the WAN line uses VLANs, traffic acceleration can be performed for each VLAN network. In this case, the IP addresses of the own device and opposing device are required for each VLAN.



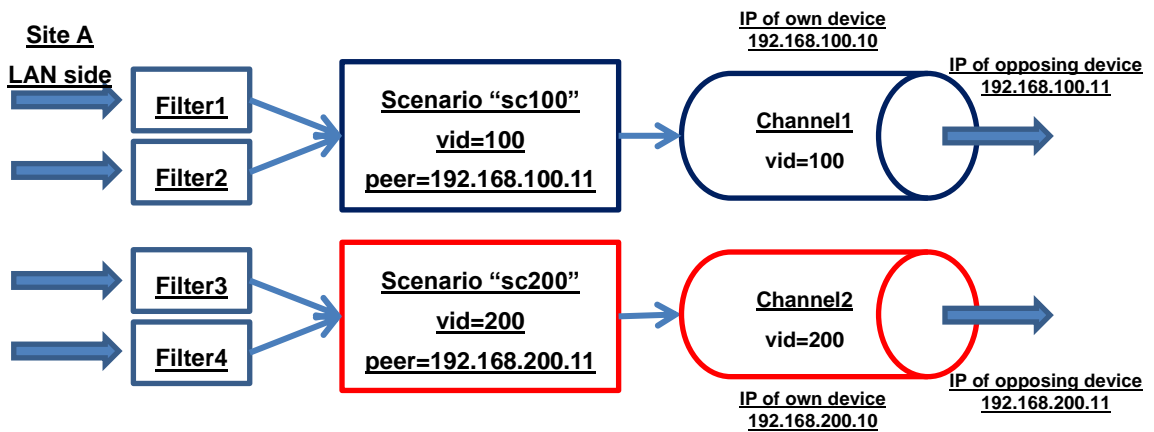
In addition, this device can be installed at multiple sites to perform traffic acceleration between multiple sites. In this case, create a scenario for each destination, and specify the IP address of the opposing device in the scenario.



The acceleration tunnel is constructed with a combination of the IP address/VLAN (channel interface) of the own device and the IP address/VLAN (Scenario) of the opposing device. If the channel interface and the VLANs of scenario for own device and opposing device are different, acceleration tunnel cannot be constructed.

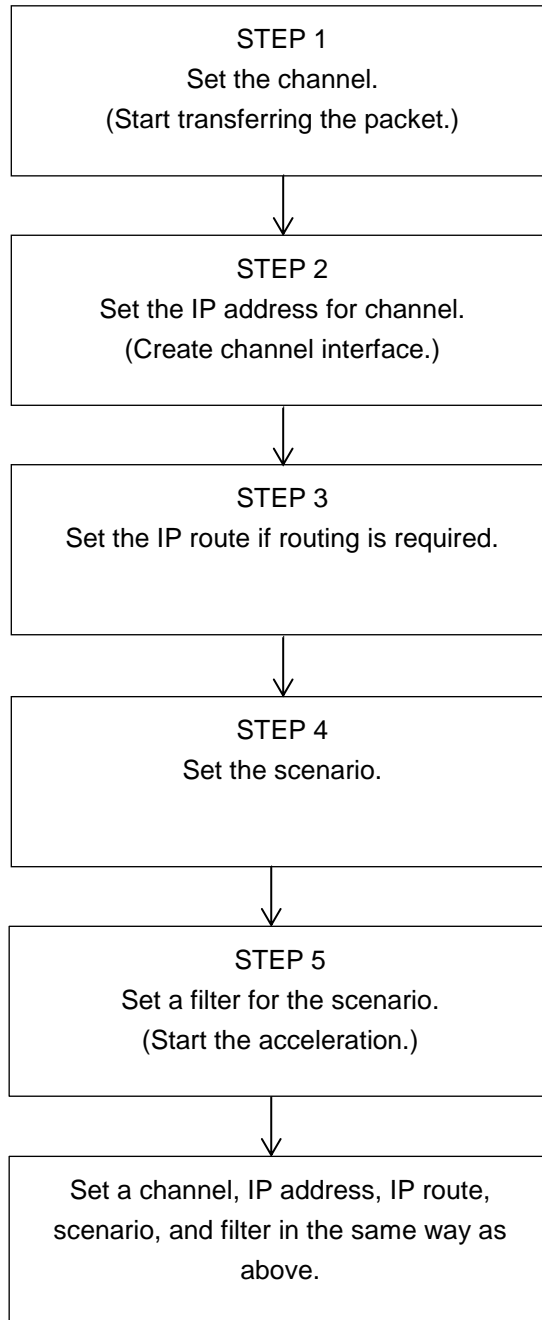


When constructing tunnels by using several VLANs, the channels are determined by VLANs of scenario.



8.9 Setting Procedure

The figure below describes the setting procedure.



Details of each step are described below.

STEP 1: Set the channel

This device specifies the LAN-side Network port, WAN-side Network port, and VLAN according to the channel registration to perform the traffic acceleration. This setting is required to activate the traffic acceleration.

In addition, for the wide area Ethernet line in which the WAN line uses VLANs, each VLAN needs a separate channel. There are two types of channel: a normal channel that transfers applicable flow to VLAN, and a default channel that transfers non-applicable flow to the normal channel.

The following parameters can be specified for channel registrations.

Parameter	Setting range	Optional/required	Description
Channel name (channel_name)	“xxxxxxxx”	Required	Specifies the channel name. The setting range is from 1 to 32 characters.
LAN-side port (slot/port, group_name)	1/1,1/2,1/3,1/4 “xxxxxxxx”	Required	Specifies the LAN side Network port. The slot position is fixed to 1. Specifies the group name registered in the port group setting when using the link aggregation (Note1).
WAN-side port (slot/port, group_name)	1/1,1/2,1/3,1/4 “xxxxxxxx”	Required	Specifies the WAN side Network port. The slot position is fixed to 1. Specifies the group name registered in the port group setting when using the link aggregation (Note1).
VLAN ID (VID, none)	1 to 4094 none	Required	Specifies VLAN ID of channel. Specifies “none” for the traffic acceleration of the frame without the VLAN tag.
TPID (tpid)	0x8100, 0x88a8, 0x9100, 0x9200, 0x9300	Optional	Specifies the TPID (Tag Protocol Identifier) of a channel.
Inner-VLAN ID (VID, none)	1 to 4094 none	Optional	Specifies the Inner-VLAN ID of a channel. Specifies "none" for the traffic acceleration of the frame without the Inner-VLAN tag.
Inner-TPID (tpid)	0x8100, 0x88a8, 0x9100, 0x9200, 0x9300	Optional	Specifies the Inner-TPID (Tag Protocol Identifier) of a channel.
MTU (mtu)	300 to 10200 [Byte]	Optional	Specifies the MTU (Maximum Transmission Unit) of a channel. This parameter applies both of the LAN-side and WAN-side Network ports.
Channel type (default)	default	Optional	Specify “default” when registering default channel.

Note 1:

Currently, port groups are not supported.

The CLI commands related to the channel settings are shown below:

<pre>add channel <channel_name> lan {<slot/port <group_name>} wan {<slot/port <group_name>} vid {<VID> none} [tpid <tpid>] [inner-vid {<VID> none}] [inner-tpid <tpid>] [mtu <mtu>]</pre>	<p>Registers a normal channel.</p> <p>Used when registering the channel for each VLAN for the traffic acceleration.</p>
<pre>add channel <channel_name> lan {<slot/port <group_name>} wan {<slot/port <group_name>} default</pre>	<p>Registers a default channel.</p> <p>Used when transferring the flow that is not applicable to the normal channel.</p>
<pre>delete channel all</pre>	<p>Deletes all channels.</p>
<pre>delete channel <channel_name></pre>	<p>Deletes the specified channel.</p>
<pre>show channel all</pre>	<p>Displays information on all channels.</p>
<pre>show channel name <channel_name> [next]</pre>	<p>Displays information on the specified channel.</p>

Examples of the channel settings are described below:

Sample 1) When connecting Network port 1/1 on the LAN side and Network port 1/2 on the WAN side, naming the frame without the VLAN Tag as channel name "ch1", and then performing the traffic acceleration

```
PureFlow(A)> add channel "ch1" lan 1/1 wan 1/2 vid none
```

Sample 2) When connecting Network port 1/1 on the LAN side and Network port 1/2 on the WAN side, naming the traffic whose VLAN ID is 100 as channel name "ch2", and then performing the traffic acceleration

```
PureFlow(A)> add channel "ch2" lan 1/1 wan 1/2 vid 100
```

Sample 3) When connecting port Network 1/1 on the LAN side and Network port 1/2 on the WAN side, and registering the default channel of the traffic

```
PureFlow(A)> add channel "default" lan 1/1 wan 1/2 default
```

This device can also register the port group of the link aggregation port (Note 2).
The following parameters can be specified for port group registrations.

Parameter	Setting range	Optional/required	Description
Port group name (group_name)	“xxxxxxxx”	Required	Specifies the port group name. The setting range is from 1 to 32 characters.
Link aggregation port (slot/port)	1/1, 1/2, 1/3, 1/4	Required	Specifies two Network ports. The slot position is fixed to 1.

Note 2:

Currently, port groups are not supported.

The CLI commands related to the port group settings are shown below (Note 3):

add port group <group_name> port <slot/port>, <slot/port>	Registers the port group.
delete port group all	Deletes all port groups.
delete port group <group_name>	Deletes the specified port group.
show port group all	Displays the information of all port groups.
show port group <group_name>	Displays the information of the specified port group name.

Note 3:

Currently, this command is not supported.

STEP 2: Set the IP address of the channel.

An acceleration tunnel (Port number for the TCP connection: 10000) is constructed between the IP address of the channel interface of own device and the IP address of the channel interface of the opposing device. This setting is required to activate the traffic acceleration. This setting is not required to activate the traffic shaping only.

Parameters that can be set by the channel interface setting are shown below:

Parameter	Setting range	Optional/required	Description
Channel name (channel_name)	“xxxxxxxx”	Required	Specifies the channel name. The default channel name cannot be specified.
IP address (IP_address)	IPv4 and IPv6 address	Required	Specifies the IPv4/IPv6 address of the channel interface. The IPv4 and IPv6 addresses can be set simultaneously to a single channel.
Subnet mask/Prefix length (netmask)	xxx.xxx.xxx.xxx/ 0 to 128	Required	Specifies a subnet mask when specifying the IPv4 address for the channel interface. Specifies a prefix length when specifying the IPv6 address for the channel interface.

The CLI commands related to the channel interface are shown below:

set ip channel <channel_name> <IP_address> netmask <netmask>	Sets the IP network interface of a channel (channel interface).
unset ip channel all	Releases all the channel interface settings.
unset ip channel <channel_name> [{ipv4 ipv6}]	Releases the channel interface setting of the specified channel/IP version.
show ip channel all	Displays information on all the channel interfaces.
show ip channel name <channel_name> [next]	Displays information on the channel interface of the specified channel.

Examples of the channel interface settings are described below:

Sample 1) Set the channel interface whose IPv4 address is 20.1.5.9 and subnet mask is 255.255.255.0 to the channel “ch1”.

```
PureFlow (A) > set ip channel "ch1" 20.1.5.9 netmask 255.255.255.0
```

Sample 2) Release the settings for the channel interface of the channel “ch1”.

```
PureFlow (A) > unset ip channel "ch1"
```

STEP 3: Set the IP route.

This device registers the static paths (default and target paths) of the channel interface and determines the traffic transfer destination according to the IP route registration. This setting is required to activate the traffic acceleration.

Parameters that can be set by the static path registration are shown below:

Parameter	Setting range	Optional/required	Description
IP address (IP_address)	xxx.xxx.xxx.xxx	Required	Specifies IPv4/IPv6 address of the destination network.
Subnet mask/ Prefix length (netmask)	xxx.xxx.xxx.xxx/ 0 to 128	Required	Specifies a subnet mask when specifying the IPv4 address for the destination network. Specifies a prefix length when specifying the IPv6 address for the destination network.
Gateway address (gateway)	xxx.xxx.xxx.xxx	Required	Specifies the gateway IPv4/IPv6 address.
Channel name (channel_name)	“xxxxxxxx”	Required	Specifies the channel name. The default channel name cannot be specified.
LAN side path/ WAN side path (lan, wan)	lan wan	Required	Specifies "lan" and "wan" when registering the static path of the LAN-side and WAN-side, respectively.

The CLI commands related to the static path settings are shown below:

add route default gateway <IP_address> channel <channel_name> {lan wan}	Registers the static path (default path) of the channel interface.
add route target <IP_address> netmask <netmask> gateway <gateway> channel <channel_name> {lan wan}	Registers the static path (target path) of the channel interface.
delete route all	Deletes all static paths.
delete route target <IP_address> netmask <netmask> gateway <gateway> channel <channel_name> {lan wan}	Deletes the specified destination network static path.
show route all	Displays all static path information.
show route channel <channel_name>	Displays the static path information of the specified channels.
show route target <IP_address> netmask <netmask> gateway <gateway> channel <channel_name> {lan wan} [next]	Displays information on the static path of the specified destination network.

Examples of the static path settings are described below:

Sample 1) When registering the static path whose destination IPv4 Network address is 30.2.1.0/24 and gateway address is 20.1.5.1 as a WAN-side static path of the channel "ch1"

```
PureFlow(A)> add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.1  
channel "ch1" wan
```

Sample 2) When deleting the static path whose destination IPv4 Network address is 30.2.1.0/24 and gateway address is 20.1.5.1 in the WAN-side static path of channel "ch1"

```
PureFlow (A) > delete route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.1  
channel "ch1" wan
```

Sample 3) Delete all static paths.

```
PureFlow (A) > delete route all
```

STEP 4: Set the scenario.

This device assigns a traffic attribute for each virtual circuit according to the scenario registration. This setting is required to activate the traffic acceleration or traffic shaping.

The following parameters can be set for Level 2 and lower scenarios:

Parameter	Setting range	Optional/required	Description
Scenario name (scenario_name)	"/port1/xxxx" (Level 2) "/port2/xxxx" (Level 2) "/port3/xxxx" (Level 2) "/port4/xxxx" (Level 2) "/group1/xxxx" (Level 2) "/group2/xxxx" (Level 2) "/port1/xxxx/xxxx" (Level 3) "/port2/xxxx/xxxx" (Level 3) "/port3/xxxx/xxxx" (Level 3) "/port4/xxxx/xxxx" (Level 3) "/group1/xxxx/xxxx" (Level 3) "/group2/xxxx/xxxx" (Level 3) And so forth (to Level 8)	Required	Cannot be omitted or changed in update command. For the first level, specify the port number such as "/port1" for the Network port number, and then specify a scenario name to be registered for the second level and lower. When the port group was registered, specify the port group name as "/group1" (Note1). Valid values are from 1 to 128 characters for all levels (/port1, /port2/port3, /port4).
Action mode	wan-accel: Acceleration mode Accelerates the traffic of the TCP traffic that matches the filter. aggregate: Aggregate queue mode Controls the traffic of all traffic that match the filter by one queue. individual: Individual queue mode Controls the traffic that matches the filter by individual queue. discard: Discard mode Discards the traffic that matches the filter.	Required	Cannot be omitted or changed in update command.
Class (class)	1 to 8	Optional	When omitted: 2 1 (high) ⇔ (low) 8 Enabled for the aggregate and individual modes.
Minimum bandwidth (min_bandwidth)	0, 10 k[bit/s] to 10 G[bit/s] (Setting unit: 1k[bit/s])	Optional	When omitted: Minimum bandwidth is not guaranteed. Enabled for the aggregate, individual and wan-accel modes.

Parameter	Setting range	Optional/required	Description
Maximum bandwidth (peak_bandwidth)	10 k[bit/s] to 10 G[bit/s] (Setting unit: 1k[bit/s])	Optional	When omitted: Maximum bandwidth is not limited. Enabled for the aggregate, individual and wan-accel modes.
Buffer size (bufsize)	2 k[Byte] to 1G[Byte] (Setting unit: 1k[Byte])	Optional	When omitted: 15 MByte Enabled for the aggregate, individual and wan-accel modes.
Scenario index (scenario_id)	1 to 40000	Optional	Cannot be changed in update command. When omitted: Auto assignment Enabled for all action modes.
Maximum number of queues (maxquenum)	1 to 4096	Optional	When omitted: 4096 Enabled for the individual mode.
Queue division target (quedivision)	<p>default: A combination of 5 tuple (sip, dip, proto, sport, dport) is used to divide queues.</p> <p>vlan: Divides queues based on VLAN ID.</p> <p>ethertype: Divides queues based on Ethernet Type/Length.</p> <p>sip: Divides queues based on the source IP address.</p> <p>dip: Divides queues based on the destination IP address.</p> <p>proto: Divides queues based on the protocol number.</p> <p>sport: Divides queues based on the source port number.</p> <p>dport: Divides queues based on the destination port number.</p>	Optional	When omitted: default Enabled for the individual mode.

Parameter	Setting range	Optional/required	Description
Action when maximum queue count reached (failaction)	discard: Performs discard. forwardbesteffort : Performs best effort transfer (class 8). forwardattribute : Transfers a specified traffic attribute.	Optional	When omitted: forwardbesteffort Specifies the actions applied to the flows (ARP, etc.) other than IP when either of 5 tuple (sip, dip, tos, proto, sport, and dport) is specified as the queue division target. Enabled for the individual mode.
Minimum bandwidth when the maximum number of queues is exceeded (fail_min_bw)	0, 10 k[bit/s] to 10 G[bit/s]	Optional	When omitted: Minimum bandwidth is not guaranteed. Available when "forwardattribute" is specified in the individual mode
Maximum bandwidth when the maximum number of queues is exceeded (fail_peak_bw)	10 k[bit/s] to 10 G[bit/s]	Optional	When omitted: Maximum bandwidth is not limited. Available when "forwardattribute" is specified in the individual mode
Class when the maximum number of queues is exceeded. (fail_class)	1 to 8	Optional	When omitted: 8 1 (high) ⇔ (low) 8 Available only when "forwardattribute" is specified in the individual queue mode
Primary IP address of opposing device (peer)	IPv4 or IPv6 address	Required	Cannot be changed in the update command Specifies the Primary IP address of the opposing device to construct acceleration tunnel. Enabled only for wan-accel mode.

Parameter	Setting range	Optional/ required	Description
Secondary IP address of opposing device (second-peer)	IPv4 or IPv6 address	Optional	Cannot be changed in the update command Specifies the Secondary IP address of the opposing device to construct acceleration tunnel. Can be specified in up to 100 acceleration mode scenarios. Enabled only for wan-accel mode. (For details, see “8.14.6 Traffic acceleration redundancy”.)
TCP connection port number (dport)	10001 to 20000	Optional	Cannot be changed in the update command When omitted: 10000 Specifies the Secondary IP address of the opposing device that constructs the acceleration tunnel. Specify an identical value for dport of the scenario that is set to the own device and opposing device. Enabled only for wan-accel mode.
VLAN ID (VID)	1 to 4094	Optional	When omitted: none Specifies VLAN ID of the channel for the traffic acceleration. Enabled only for wan-accel mode.
Inner-VLAN ID (VID)	1 to 4094	Optional	When omitted: none Specifies Inner-VLAN ID of the channel for the traffic acceleration. Enabled only for wan-accel mode.
CoS (through, user_priority)	through 0 to 7	Optional	When omitted: through Set the CoS overwrite value. Enabled for the aggregate, individual, and wan-accel modes. Cannot be changed with the update command in the wan-accel mode only.
Inner-CoS (through, user_priority)	through 0 to 7	Optional	When omitted: through Set the Inner-CoS overwrite value. Enabled for the aggregate, individual, and wan-accel modes. Cannot be changed with the update command in the wan-accel mode only.
DSCP (through, user_priority)	through 0 to 63	Optional	When omitted: through Set the DSCP overwrite value. Enabled for the aggregate, individual, and wan-accel modes. Cannot be changed with the update command in the wan-accel mode only.

Parameter	Setting range	Optional/ required	Description
Enabling/Disabling compression function (compression)	enable disable	Optional	When omitted: Enabled Specifies enabling/disabling compression of the TCP data for traffic acceleration. Enabled only for wan-accel mode.
TCP buffer size (tcp-mem)	auto 64 k[Byte] to 200 M[Byte] (Setting unit: 1k[Byte])	Optional	When omitted: auto Specifies TCP buffer size. Enabled only for wan-accel mode.
Congestion control mode (cc-mode)	normal semi-fast fast	Optional	When omitted: normal Specifies congestion control mode of traffic acceleration. Enabled only for wan-accel mode.
Auto-bypass RTT threshold value for traffic acceleration (bypass-thresh)	0 to 10000 millisecond	Optional	When omitted: 0 Specifies the RTT (Round Trip Time) threshold of the auto bypass function of traffic acceleration in millisecond. Enabled only for wan-accel mode. (For details, see “8.14.5 Traffic acceleration bypass”.)
Enabling/disabling the Keep Alive monitoring of the auto-bypass function for the traffic acceleration (bypass-keepalive)	enable disable	Optional	When omitted: Disable Specifies enabling/disabling the Keep Alive monitoring of auto-bypass function for traffic acceleration. Can be specified in up to 100 acceleration mode scenarios. Enabled only for wan-accel mode. (For details, see “8.14.5 Traffic acceleration bypass”.)
Enabling/Disabling TCP-FEC function (fec)	enable disable	Optional	When omitted: Disable Specifies enabling/disabling the TCP-FEC function. Enabled only for wan-accel mode. (For details, see “8.14.7 TCP-FEC function”.)
FEC block size (block-size)	2 k[Byte] to 50 k[Byte] (Setting unit: 1k[Byte])	Optional	When omitted: 2 k Specifies the FEC block size of TCP-FEC function. Enabled only for wan-accel mode. (For details, see “8.14.7 TCP-FEC function”.)

Parameter	Setting range	Optional/ required	Description
Data block size (data-block-size)	2 k[Byte] to 200 k[Byte] (Setting unit: 1k[Byte])	Optional	When omitted: 20 k Specifies the data block size of TCP-FEC function. Enabled only for wan-accel mode. (For details, see “8.14.7 TCP-FEC function”.)
FEC session count (fec-session)	0 to 1000 session	Optional	When omitted: 1000 Specifies the TCP sessions (FEC sessions) count that uses the TCP-FEC function Limits the FEC session count available for the scenarios by using this parameter. Enabled only for wan-accel mode. (For details, see “8.14.7 TCP-FEC function”.)

Note 1:

Currently, port groups are not supported.

The following CLI commands are for Level 2 and lower scenarios settings:

<pre>add scenario <scenario_name> action discard [scenario <scenario_id>]</pre>	<p>Registers a discard mode scenario. A scenario index is automatically assigned, and normally need not be specified.</p>
<pre>add scenario <scenario_name> action aggregate [cos {through <user_priority>} [inner-cos {through <user_priority>} [dscp {through <user_priority>} [min_bw <min_bandwidth>} [peak_bw <peak_bandwidth>} [class <class>] [bufsize <bufsize>} [scenario <scenario_id>]</pre>	<p>Registers a scenario in the aggregate queue mode. Traffic attributes such as the bandwidth and buffer size are specified. A scenario index is automatically assigned, and normally need not be set.</p>
<pre>add scenario <scenario_name> action individual [cos {through <user_priority>} [inner-cos {through <user_priority>} [dscp {through <user_priority>} [min_bw <min_bandwidth>} [peak_bw <peak_bandwidth>} [class <class>] [bufsize <bufsize>} [scenario <scenario_id>} [maxquenum <quenum>} [quedivision <field>} [failaction {discard forwardbesteffort forwardattribute}] [fail_min_bw <min_bandwidth>} [fail_peak_bw <peak_bandwidth>} [fail_class <class>]</pre>	<p>Registers a scenario in the individual queue mode. Traffic attributes such as the bandwidth and buffer size are set. Also, the maximum number of individual queues, queue division target, and action when the maximum number of queues is exceeded are specified. A scenario index is automatically assigned, and normally need not be set.</p>
<pre>add scenario <scenario_name> action wan-accel peer <IP_address> second-peer <IP_address> [dport <port>} [vid <vid>] [inner-vid <VID>} [cos {through <user_priority>} [inner-cos {through <user_priority>} [dscp {through <user_priority>} [compression {enable disable}] [tcp-mem {auto <size>} [cc-mode {normal semi-fast fast}] [bypass-thresh <rtt>} [bypass-keepalive {enable disable}] [fec {enable disable}] [block-size <size>] [data-block-size <size>} [fec-session <session>} [min_bw <min_bandwidth>} [peak_bw <peak_bandwidth>} [bufsize <bufsize>} [scenario <scenario_id>]</pre>	<p>Registers a scenario in the acceleration mode. Traffic attributes of traffic acceleration, such as the IP address, TCP connection port number, VLAN of the opposing device are set. A scenario index is automatically assigned, and normally need not be set. The function is enabled as soon as the IP address of opposing device of "Secondary" is specified.</p>

<pre>update scenario <scenario_name> action aggregate [cos {through <user_priority>} [inner-cos {through <user_priority>} [dscp {through <user_priority>} [min_bw <min_bandwidth>} [peak_bw <peak_bandwidth>} [class <class>] [bufsize <bufsize>]</pre>	<p>Changes a scenario in aggregate queue mode.</p> <p>This command allows you to change a traffic attribute while traffic is being controlled. Each of the parameters can be omitted but you cannot omit all the parameters. Specify at least one parameter that you want to change.</p> <p>The scenario name, action mode, and scenario index cannot be changed.</p>
<pre>update scenario <scenario_name> action individual [cos {through <user_priority>} [inner-cos {through <user_priority>} [dscp {through <user_priority>} [min_bw <min_bandwidth>} [peak_bw <peak_bandwidth>} [class <class>] [bufsize <bufsize>} [maxquenum <quenum>} [quedivision <field>} [failaction {discard forwardbesteffort forwardattribute}] [fail_min_bw <min_bandwidth>} [fail_peak_bw <peak_bandwidth>} [fail_class <class>]</pre>	<p>Changes a scenario in individual queue mode.</p> <p>This command allows you to change a traffic attribute while traffic is being controlled. Each of the parameters can be omitted but you cannot omit all the parameters. Specify at least one parameter that you want to change.</p> <p>The scenario name, action mode, and scenario index cannot be changed.</p>
<pre>update scenario <scenario_name> action wan-accel [vid <vid>] [inner-vid <VID>} [compression {enable disable}] [tcp-mem {auto <size>} [cc-mode {normal semi-fast fast}] [bypass-thresh <rtt>} [bypass-keepalive {enable disable}] [fec {enable disable}] [block-size <size>] [data-block-size <size>} [fec-session <session>} [min_bw <min_bandwidth>} [peak_bw <peak_bandwidth>} [bufsize <bufsize>]</pre>	<p>Changes a scenario in acceleration mode.</p> <p>This command allows you to change a traffic attribute while traffic is being controlled. Each of the parameters can be omitted but you cannot omit all the parameters. Specify at least one parameter that you want to change.</p> <p>The scenario name, action mode, IP addresses of the opposing device, TCP connection port number of the opposing device, VLAN and scenario index cannot be changed.</p>
<pre>delete scenario all</pre>	<p>Deletes all scenarios.</p>
<pre>delete scenario <scenario_name> [recursive]</pre>	<p>Deletes the scenario of the specified scenario name.</p> <p>Deletes the scenarios lower than the specified scenario when recursive is specified.</p> <p>Cannot delete the scenarios lower than the specified scenario when recursive is not specified.</p>

show scenario all	Displays the information of all scenarios.
show scenario name <scenario_name> [summary] [next]	Displays the scenario information of the specified scenarios name. Does not display the filter information when summary is specified. Displays the next scenario information when next is specified.
set scenario tree mode {inbound outbound}	Sets the tree mode of the traffic attributes (scenario)(input/output side). The scenario tree mode specifies whether the scenario and filter classifications are applied to the input traffic to the Network port and the output traffic from the Network port, respectively.
show scenario tree	Displays the scenario tree information of all scenarios.

An example of the Level 2 scenario is shown below.

Sample 1) Register the scenario of the maximum bandwidth to 3 Gbit/s for the aggregate queue mode scenario of the “Tokyo” base received from Network port1/1.

```
PureFlow (A) > add scenario "/port1/Tokyo" action aggregate peak_bw 3G
```

Sample 2) Register the scenario of the maximum bandwidth to 500 kbit/s and the maximum number of queues to 20 for the individual queue mode scenario of the “Osaka” base received from Network port1/1.

```
PureFlow (A) > add scenario "/port1/Osaka" action individual peak_bw 500k maxquenum 20
```

Sample 3) Register the scenario of the IP address of the opposing device to 20.1.2.9 for the acceleration mode scenario for the “Nagoya” base received from Network port1/1.

```
PureFlow (A) > add scenario "/port1/Nagoya" action wan-accel peer 20.1.2.9
```

Sample 4) Register the scenario of the IP address of the opposing device to 20.1.2.9 for the acceleration mode scenario for the “Nagoya” base received from port-grouped “group1”.

```
PureFlow (A) > add scenario "/group1/Nagoya" action wan-accel peer 20.1.2.9
```


The level 3 and lower scenarios can also be specified by a scenario name indicating the upper level scenarios and the hierarchy.

Sample 5) Register the “Shinjuku” area under the “Tokyo” base as an aggregate queue mode scenario, and register the scenario of the maximum bandwidth to 100 Mbit/s.

```
PureFlow (A) > add scenario "/port1/Tokyo/Shinjuku" action aggregate peak_bw 100M
```

An example of deleting a scenario is shown below:

Sample 6) Delete the scenarios under the “Tokyo” base.

```
PureFlow (A) > delete scenario "/port1/Tokyo" recursive
```

STEP 5: Set the filter

This device uses filters to identify the Bridge-ctrl frame, Ethernet frame, IPv4 packet, and IPv6 packet traffic. This setting is required to activate the traffic acceleration or traffic shaping.

The following parameters can be set for the Level 2 and lower filters:

Parameter		Setting range	Optional/required
Filter name (filter name)		1 to 48 characters	Required
Scenario name (scenario name)		Total of 1 to 128 characters for all levels (register by the “add scenario” command)	Required
Filter type		bridge-ctrl, ethernet, ipv4, ipv6	Required
Ethertype (ethertype)		Specifies the Type field in the Ethernet header. 0x0000 to 0xFFFF	Optional Enabled only for Ethernet filters
VLAN ID (VID)		Specifies IEEE802.1Q VLAN ID. 0 to 4094 (range specification available), none (without VLAN tag)	Optional
Inner-VLAN ID (VID)		Specifies Inner-VLAN ID in QinQ. 0 to 4094 (range specification available), none (without VLAN tag)	Optional
Source IP address (sip)	IPv4	0.0.0.0 to 255.255.255.255 (The range “start-end” can be specified.) Rule list name	Optional Enabled only for IP filters
	IPv6	0::0 to FFFF:...:FFFF (The range “start-end” can be specified in lowercase.) Rule list name	Optional Enabled only for IP filters
Destination IP address (dip)	IPv4	0.0.0.0 to 255.255.255.255 (The range “start-end” can be specified.) Rule list name	Optional Enabled only for IP filters
	IPv6	0::0 to FFFF:...:FFFF (The range “start-end” can be specified in lowercase.) Rule list name	Optional Enabled only for IP filters
Protocol number (proto)		0 to 255 (The range “start-end” can be specified.) (A string can be specified for tcp, udp, and icmp.)	Optional Enabled only for IP filters
Source port number (sport)		0 to 65535 (The range “start-end” can be specified.) Rule list name	Optional Enabled only for IP filters
Destination port number (dport)		0 to 65535 (The range “start-end” can be specified.) Rule list name	Optional Enabled only for IP filters
Filter priority (priority)		1 to 40000	Optional When omitted: 20000

The following CLI commands are for Level 2 or lower filters:

<pre>add filter scenario <scenario_name> filter <filter_name> bridge-ctrl [priority <filter_pri>]</pre>	<p>Identifies frames with destination MAC addresses 01-80-C2-00-00-00 to 01-80-C2-00-00-FF (including spanning tree protocol, link aggregation, EAPoL (authentication protocol)).</p>
<pre>add filter scenario <scenario_name> filter <filter_name> ethernet [vid {<VID> none}] [inner-vid {<VID> none}] [ethertype <type>] [priority <filter_pri>]</pre>	<p>Identifies frames based on the length/type field of the Ethernet header. This can also be specified for VLAN ID in the VLAN tag.</p> <p>Each of the parameters can be omitted but you cannot omit all the parameters. Specify at least one parameter other than “priority”.</p>
<pre>add filter scenario <scenario_name> filter <filter_name> ipv4 [vid {<VID> none}] [inner-vid {<VID> none}] [sip [list] {<src_IP_address> <list_name>}] [dip [list] {<dst_IP_address> <list_name>}] [proto <protocol>] [sport [list] {<sport> <list_name>}] [dport [list] {<dport> <list_name>}] [priority <filter_pri>]</pre>	<p>Identifies IPv4 packets based on the IP address, protocol number, port number, etc. This can also be specified for VLAN ID.</p> <p>Each parameter can be omitted. If all parameters are omitted, all IPv4 packets are targeted.</p>
<pre>add filter scenario <scenario_name> filter <filter_name> ipv6 [vid {<VID> none}] [inner-vid {<VID> none}] [sip [list] {<src_IP_address> <list_name>}] [dip [list] {<dst_IP_address> <list_name>}] [proto <protocol>] [sport [list] {<sport> <list_name>}] [dport [list] {<dport> <list_name>}] [priority <filter_pri>]</pre>	<p>Identifies IPv6 packets based on the IP address, protocol number, port number, etc. This can also be specified for VLAN ID.</p> <p>Each parameter can be omitted. If all parameters are omitted, all IPv6 packets are targeted.</p>
<pre>delete filter scenario <scenario_name> filter <filter_name></pre>	<p>Deletes the specified filter of the specified scenario.</p>
<pre>delete filter scenario <scenario_name></pre>	<p>Deletes all filters in the specified scenario.</p>
<pre>delete filter all</pre>	<p>Deletes all filters.</p>
<pre>show filter scenario <scenario_name> [filter <filter_name>] [summary] [next]</pre>	<p>Displays the filter information of the specified scenarios.</p> <p>Displays the filter names only when summary is specified.</p> <p>Displays the next scenario information when next is specified.</p>
<pre>show filter all</pre>	<p>Displays all filter settings of all scenarios.</p>

An example of a Level 2 filter is shown below.

Sample 1) Register BPDU filter as a filter for the Level 2 scenario “/port1/bpdu”.

```
PureFlow (A) > add filter scenario "/port1/bpdu" filter "bpdu" bridge-ctrl priority 1
```

Sample 2) Register ARP filter as the filter for the Level 2 scenario “/port1/arp”.

```
PureFlow (A) > add filter scenario "/port1/arp" filter "arp" ethernet ethertype 0x0806
```

Sample 3) Register a filter whose VLAN ID for IPv4 is “10” as a filter for the Level 2 scenario “/port1/Tokyo”.

```
PureFlow (A) > add filter scenario "/port1/Tokyo" filter "Tokyo" ipv4 vid 10
```

Sample 4) Register a filter whose VLAN ID for IPv6 is “20” as a filter for the Level 2 scenario “/port1/Osaka”.

```
PureFlow (A) > add filter scenario "/port1/Osaka" filter "Osaka" ipv6 vid 20
```

In the same way, specify a scenario to set a filter for Level 3 and lower scenarios.

Sample 5) Register a filter whose source IP address for IPv4 is in the range of “192.168.10.0 to 192.168.10.255” as a filter for the Level 3 scenario “/port1/Tokyo/Shinjuku”.

```
PureFlow (A) > add filter scenario "/port1/Tokyo/Shinjuku" filter "Shinjuku" ipv4  
sip 192.168.10.0-192.168.10.255
```

8.10 How to Set a Rule List

This chapter describes how to set a rule list.

To use a rule list, perform the following procedure:

Step 1: Register the rule list.

Step 2: Register a rule list entry to the rule list.

Step 3: Specify the rule list for the add filter command.

Parameters for the rule list and rule list entry are as follows:

Parameters for the rule list

Parameter	Setting range
Rule list name	1 to 32 characters
Rule list type	ipv4, ipv6, l4port

Parameters for rule list entries

Parameter	Setting range	
Rule list name	Specify a registered rule list name.	
Rule list type	ipv4, ipv6, l4port	
Conditions for traffic division	IPv4 address	0.0.0.0 - 255.255.255.255
	IPv6 address	0::0 to FFFF:...:FFFF (lowercase letters available)
	TCP/UDP port number	0 to 65535 (range specification available)

The CLI commands for setting a rule list are displayed as follows:

add rulelist group <list_name> {ipv4 ipv6 l4port}	Registers a rule list. Either ipv4 or ipv6 or l4port is targeted.
add rulelist entry <list_name> ipv4 <IP_address>	Registers the rule list entry of the IPv4 address.
add rulelist entry <list_name> ipv6 <IP_address>	Registers the rule list entry of the IPv6 address.
add rulelist entry <list_name> l4port <port>	Registers the rule list entry of the ITCP/UDP port number.
delete rulelist group {<list_name> all}	Deletes a rule list.
delete rulelist entry <list_name> ipv4 <IP_address>	Deletes the rule list entry of the IPv4 address.
delete rulelist entry <list_name> ipv6 <IP_address>	Deletes the rule list entry of the IPv6 address.

delete rulelist entry <list_name> l4port <port>	Deletes the rule list entry of the ITCP/UDP port number.
show rulelist all	Displays information on all rule lists.
show rulelist [<list_name>]	Displays the rule list information on the specified rule list.

The rule list needs to be set according to the following rules:

- (1) Specify a rule list name that is unique in the device.
- (2) The “delete rulelist group” command can be used only for rule lists not registered to filters.
- (3) “all” cannot be specified for the rule list name.

A sample setting a rule list is shown below.

Step 1) Register the rule list “TVCservers”.

```
PureFlow (A) > add rulelist group “TVCservers” ipv4
```

Step 2) Register a rule list entry to the rule list “TVCservers”.

```
PureFlow (A) > add rulelist entry “TVCservers” ipv4 172.16.111.11
PureFlow (A) > add rulelist entry “TVCservers” ipv4 172.16.112.11
      .
      . (Add a host IP to be listed.)
      .
```

Step 3) Register the rule list name “TVCservers” for “sip” of the add filter registration command.

```
PureFlow (A) > add filter scenario “/port1/Tokyo/TVC” filter “TVC” ipv4 sip list
“TVCservers”
```

8.11 Channel interface communication

IPv4 and IPv6 can be simultaneously used in the channel interface communication.

Function	IPv4	IPv6
PING	○	○
TRACEROUTE	○	○

The network communication/path can be confirmed in the channel interface by using the following commands:

ping <IP_address> channel <channel_name> {lan wan} [<send_count>]	Transmits the ICMP ECHO_REQUEST packet to the specified IP address. (IPv4/IPv6)
tracert <IP_address> channel <channel_name> {lan wan}	Displays a path that reaches the specified IP address.
arp -a channel <channel_name> <IP_Address>	Displays an ARP entry. (For IPv4 only)
arp -d <IP_address> channel <channel_name>	Deletes an ARP entry. (For IPv4 only)
delete ndp neighbor <IP_address> [channel <random_str>]	Deletes an NDP entry. (For IPv6 only)
show ndp neighbor [channel {<channel_name> all}] [<IP_address>]	Displays an NDP entry. (For IPv6 only)

Executes the commands described below when confirming the communication with the IPv4 address 192.168.10.100 from the WAN-side port set for channel.

```
PureFlow(A)> ping 192.168.10.100 channel "channel1" wan
PING 192.168.10.100 0(28) bytes of data.
 8 byte from 192.168.10.100: icmp_req=1 time=200.208 ms
 8 byte from 192.168.10.100: icmp_req=2 time=200.206 ms
 8 byte from 192.168.10.100: icmp_req=3 time=200.184 ms
--- 192.168.10.100 ping statistics ---
 3 packets transmitted, 3 received, 0% packet loss
 rtt min/avg/max = 200.184/200.199/200.208 ms
PureFlow(A)>
```

Displays the following information upon failure in the communication confirmation. Check the channel interface setting and network connection.

```
PureFlow(A)> ping 192.168.10.101 channel channel1 wan
PING 192.168.10.101 0(28) bytes of data.
from 192.168.10.101: icmp_req=1 Destination Host Unreachable
from 192.168.10.101: icmp_req=2 Destination Host Unreachable
from 192.168.10.101: icmp_req=3 Destination Host Unreachable
--- 192.168.10.101 ping statistics ---
 3 packets transmitted, 0 received, 100% packet loss
PureFlow(A)>
```

Executes the commands described below when deleting the ARP entry of IPv4 address 192.168.10.101.

```
PureFlow(A)> arp -d 192.168.10.100 channel "channel1"  
PureFlow(A)> arp -a channel "channel1" 192.168.10.100  
IP address      MAC address      type  
-----
```

```
PureFlow(A)>
```

Executes the commands described below when confirming the communication with the IPv6 address 2001:DB8::1 from the WAN-side port set for channel.

```
PureFlow(A)> ping 2001:db8::1 channel "channel1" wan  
PING 2001:db8::1 (2001:db8::1) 56(84) bytes of data.  
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=0.372 ms
```

```
--- 2001:db8::1 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 0.372/0.372/0.372/0.000 ms  
PureFlow(A)> show ndp neighbor channel "channel1" 2001:db8::1
```

```
IP address      MAC address      type  
-----  
2001:db8::1    00-00-91-01-23-45    reachable
```

```
PureFlow(A)>
```

Displays the following information upon failure in the communication confirmation. Check the channel interface setting and network connection.

```
PureFlow(A)> ping 2001:db8::10 channel "channel1" wan  
PING 2001:db8::10 (2001:db8::10) 56(84) bytes of data.
```

```
--- 2001:db8::10 ping statistics ---  
1 packets transmitted, 0 received, 100% packet loss, time 100ms  
PureFlow(A)>
```

Executes the commands described below when deleting the NDP entry of IPv6 address 2001:db8::10:

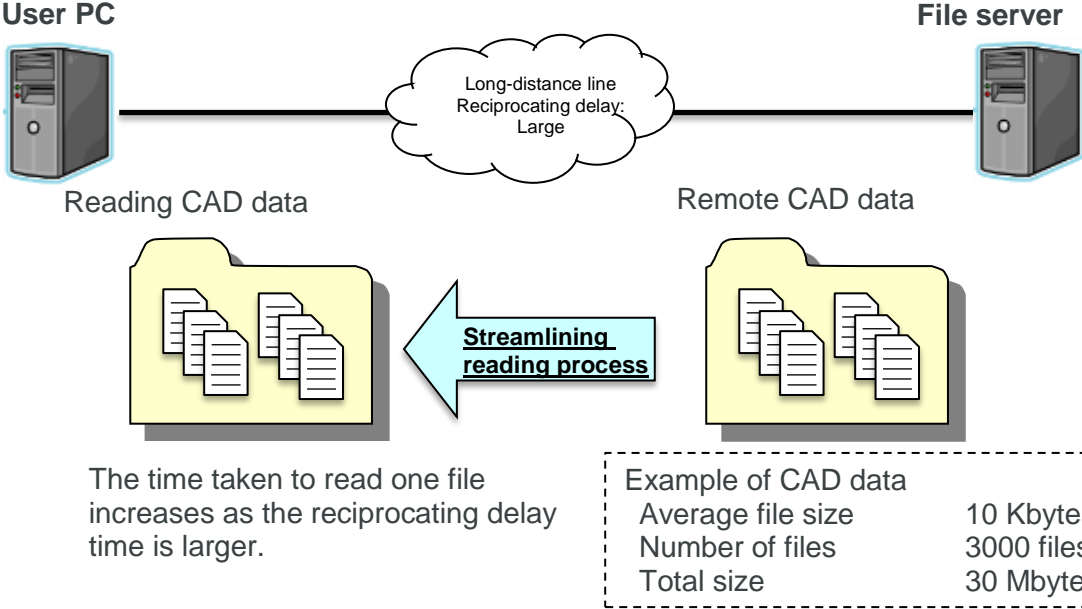
```
PureFlow(A)> delete ndp neighbor 2001:db8::10 channel "channel1"  
PureFlow(A)> show ndp neighbor channel "channel1" 2001:db8::10  
IP address      MAC address      type  
-----
```

```
PureFlow(A)>
```

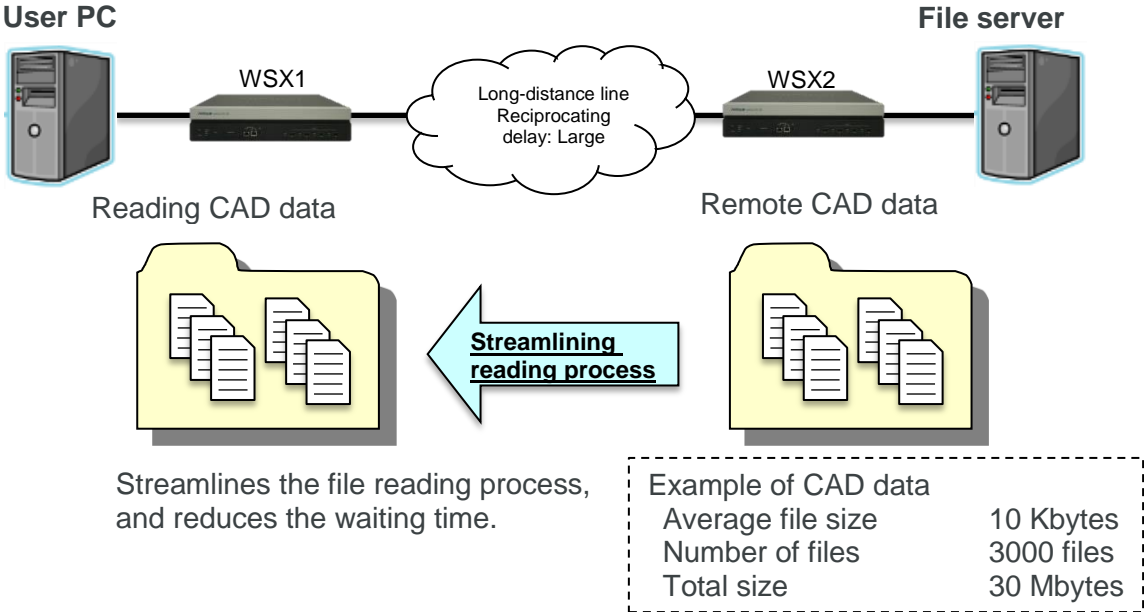

8.12 Application Acceleration Function

The application acceleration function streamlines the traffic of commands in the application protocol and accelerates the application data transfer. For example, when downloading a file from the remote file server connected in the file share protocol while the SMB protocol acceleration function is enabled, time taken for transferring the file can be reduced.

Application acceleration disabled



Application acceleration enabled



8.12.1 SMB protocol acceleration function

SMB (Server Message Block) protocol is used when setting share folders or network drive to share files in the file server in the Windows® server network. The SMB protocol acceleration function streamlines the protocol communication and accelerates reading and writing files.

In the file reading operation, the SMB protocol acceleration function optimizes the SMB command (SMB2_QUERY_INFO command) that reads the file attribution and the command (SMB2_READ command) that reads the file data, and reduces the entire time taken for reading. In the file writing operation, this function optimizes the command (SMB2_QUERY_INFO command) that reads the file attribution before writing the file data, and reduces the entire time taken for writing.

This function is effective if the version of the SMB protocol used by the SMB client and SMB server is SMB2.0 or higher and less than SMB3.0. If the SMB client supports the SMB protocol version SMB3.0 or higher, the corresponding SMB protocol is not accelerated. The following table lists combinations of the SMB servers and SMB clients.

SMB client \ SMB server	SMB 2.0.2	SMB 2.1	SMB 3.0 SMB 3.0.2 SMB 3.1.1
SMB 2.0.2	TCP acceleration SMB acceleration	TCP acceleration SMB acceleration	TCP acceleration SMB acceleration
SMB 2.1	TCP acceleration SMB acceleration	TCP acceleration SMB acceleration	TCP acceleration SMB acceleration
SMB 3.0 SMB 3.0.2 SMB 3.1.1	TCP acceleration	TCP acceleration	TCP acceleration

This function is available only by specifying the SMB protocol acceleration as shown in execution example <1>. Execution example <2> shows the settings to limit the accelerated SMB protocol only to the protocol whose TCP port number 445.

Execution example <1>: Accelerating TCP port number 139 and 445 as the SMB protocol

```
PureFlow (A) > add apl-accel scenario /port1/woc1 protocol smb
```

Execution example <2>: Limiting the TCP port number of the SMB protocol only to 445

```
PureFlow (A) > update apl-accel scenario /port1/woc1 protocol smb tcp 445
```

Execution example <3>: Deleting the SMB acceleration setting

```
PureFlow (A) > delete apl-accel scenario /port1/woc1 protocol smb
```

Specify the following parameters when using the parameters other than the default value.

Command	Parameter	Description
add apl-accel scenario update apl-accel scenario	[tcp <port>]	Specifies the TCO port number of the SMB protocol. For the SMB protocol, TCP 139 and 445 are used as the standard. If the TCP port number of the SMB protocol was changed, set the TCP port number that was changed in this parameter. Up to 16 ports can be specified by delimiting them with commas (.).
	[smb-session <session>]	Specifies the number of TCP sessions (SMB sessions) that use the Windows® file share acceleration function. This parameter restricts the number of the SMB sessions used in each scenario. Up to 10,000 SMB sessions are available for the entire device. The counts of the SMB sessions are not assured in this parameter.
	[read-attr {enable disable}]	The default value must be used normally. Specifies "enable" to enable the substitute response for the SMB2 QUERY_INFO command in the reading operation of the SMB protocol, while "disable" to disable it.

Command	Parameter	Description
add apl-accel scenario update apl-accel scenario (Continued)	read-operation {enable disable}	The default value must be used normally. Specifies "enable" to enable the substitute response for the SMB2 READ command in the reading operation of the SMB protocol, while "disable" to disable it.
	[read-cache-size <size>]	The default value must be used normally. Specifies the cache size of the substitute response for the SMB2 READ command in the reading operation of the SMB protocol. The setting range is from 64 k [Byte] to 60 M [Byte]. The minimum unit is 1 k [Byte]. Specify the unit (k, M).
	[write-attr {enable disable}]	The default value must be used normally. Specifies "enable" to enable the substitute response for the SMB2 QUERY_INFO command in the writing operation of the SMB protocol, while "disable" to disable it.
	[write-attr-1st {enable disable}]	The default value must be used normally. Specifies "enable" to enable the substitute response for the SMB2 SET_INFO command before the writing operation of the SMB protocol, while "disable" to disable it.
	[write-attr-2nd {enable disable}]	Specifies "enable" to enable the substitute response for the SMB2 SET_INFO command after the writing operation of the SMB protocol, while "disable" to disable it.
	[write-operation {enable disable}]	The default value must be used normally. Specifies "enable" to enable the substitute response for the SMB2 WRITE command in the writing operation of the SMB protocol, while "disable" to disable it.
show scenario	name <scenario_name>	Displays the scenario information (parameters related to the application acceleration function) of the specified scenario name.

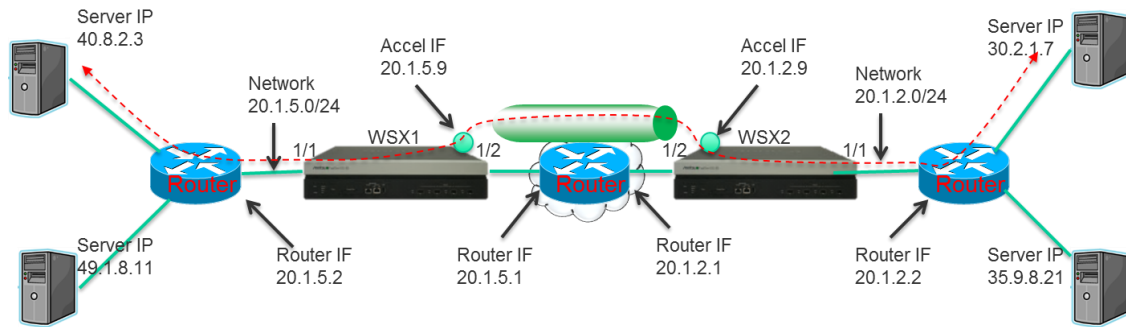
8.12.2 Precautions for SMB protocol acceleration function

- 1) If the digital signature of the SMB packet is always valid, this function accelerates the TCP communication instead of the SMB protocol. For example, if the SMB server (file share server) is the domain controller of Active Directory®, the digital signature is executed for the communication between the SMB server and SMB client (user PC). In this case, this function accelerates the TCP communication instead of the SMB protocol. If the file share server is the main server of Active Directory®, this function accelerates the SMB protocol.
- 2) If the SMB client (user PC) uses a server OS such as WindowsServer® 2008 when transferring a large file by using SMB protocol acceleration, the file transfer time can be shortened by accelerating the traffic with the SMB disabled.
- 3) When resources for the SMB protocol acceleration function are short, system logs related to the Appli-Accel Sessions and Appli-Accel Buffer are displayed. When these system logs are displayed, specify the smb-session parameter and read-cache-size parameter to limit the number of sessions and buffer size.

8.13 Configuration Example

An example of configuration for setting the following network traffic environment is shown below:

[Case 1] Accelerate the normal network that is not bound by VLAN.



WSX1 setting

Execute the following commands:

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/2 vid none
```

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/2 default
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 20.1.5.9 netmask 255.255.255.0
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 20.1.2.0 netmask 255.255.255.0 gateway 20.1.5.1 channel "ch1" wan
```

```
PureFlow (A) > add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.1 channel "ch1" wan
```

```
PureFlow (A) > add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.5.1 channel "ch1" wan
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" lan
```

```
PureFlow (A) > add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" lan
```

<Scenario setting>

```
PureFlow (A) > add scenario "/port1/woc1" action wan-accel peer 20.1.2.9
```

<Setting of filter targeted for acceleration on the LAN side>

```
PureFlow (A) > add filter scenario "/port1/woc1" filter "F1" ipv4 sip 40.8.2.3
```

```
PureFlow (A) > add filter scenario "/port1/woc1" filter "F2" ipv4 sip 49.1.8.11
```

WSX2 setting

Execute the following commands:

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/2 vid none
```

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/2 default
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 20.1.2.9 netmask 255.255.255.0
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 20.1.5.0 netmask 255.255.255.0 gateway 20.1.2.1 channel  
"ch1" wan
```

```
PureFlow (A) > add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.2.1 channel  
"ch1" wan
```

```
PureFlow (A) > add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.2.1 channel  
"ch1" wan
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.2.2 channel  
"ch1" lan
```

```
PureFlow (A) > add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.2.2 channel  
"ch1" lan
```

<Scenario setting>

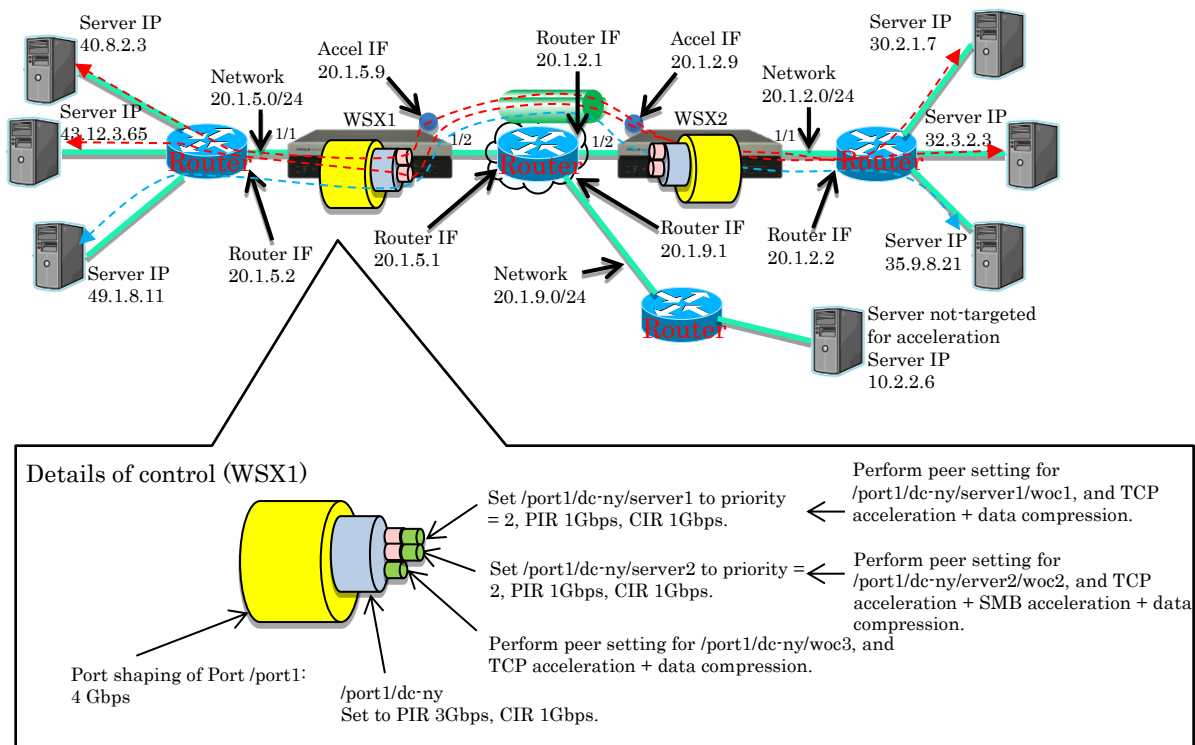
```
PureFlow (A) > add scenario "/port1/woc1" action wan-accel peer 20.1.5.9
```

<Setting of filter targeted for acceleration on the LAN side>

```
PureFlow (A) > add filter scenario "/port1/woc1" filter "F1" ipv4 sip 30.2.1.7
```

```
PureFlow (A) > add filter scenario "/port1/woc1" filter "F2" ipv4 sip 35.9.8.21
```

[Case 2] Single network (QoS is available.)



WSX1 setting

Execute the following commands:

<Channel setting>

PureFlow (A) > add channel "ch1" lan 1/1 wan 1/2 vid none

<Default channel setting>

PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/2 default

<Network interface IP address setting>

PureFlow (A) > set ip interface "ch1" 20.1.5.9 netmask 255.255.255.0

<Route setting on the WAN side>

PureFlow (A) > add route target 20.1.2.0 netmask 255.255.255.0 gateway 20.1.5.1 channel "ch1" wan

PureFlow (A) > add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.1 channel "ch1" wan

PureFlow (A) > add route target 32.3.2.0 netmask 255.255.255.0 gateway 20.1.5.1 channel "ch1" wan

PureFlow (A) > add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.5.1 channel "ch1" wan

<Route setting on the LAN side>

```
PureFlow (A) > add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.5.2 channel
"ch1" lan
```

```
PureFlow (A) > add route target 43.12.3.0 netmask 255.255.255.0 gateway 20.1.5.2 channel
"ch1" lan
```

```
PureFlow (A) > add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.5.2 channel
"ch1" lan
```

<Scenario setting>

```
PureFlow (A) > update scenario "/port1" action aggregate peak_bw 4G
```

```
PureFlow (A) > add scenario "/port1/dc-ny" action aggregate min_bw 1G peak_bw 3G
```

```
PureFlow (A) > add scenario "/port1/dc-ny/server1" action aggregate min_bw 1G peak_bw 1G
class 2
```

```
PureFlow (A) > add scenario "/port1/dc-ny/server2" action aggregate min_bw 1G peak_bw 1G
class 2
```

```
PureFlow (A) > add scenario "/port1/dc-ny/server1/woc1" action wan-accel peer 20.1.2.9
```

```
PureFlow (A) > add scenario "/port1/dc-ny/server2/woc2" action wan-accel peer 20.1.2.9
```

```
PureFlow (A) > add scenario "/port1/dc-ny/woc3" action wan-accel peer 20.1.2.9
```

<Setting of acceleration target on the LAN side and QoS filter>

```
PureFlow (A) > add filter scenario "/port1/dc-ny" filter "F0" ipv4
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/server1" filter "F1-2" ipv4 sip 40.8.2.3
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/server2" filter "F2-2" ipv4 sip 43.12.3.65
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/server1/woc1" filter "F1-3" ipv4 sip 40.8.2.3
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/server2/woc2" filter "F2-3" ipv4 sip 43.12.3.65
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/woc3" filter "F3-2" ipv4 sip 49.1.8.11
```

<Application to be accelerated: SMB>

```
PureFlow(A)> add apl-accel scenario "/port1/dc-ny/server2/woc2" protocol smb
```

WSX2 setting

Execute the following commands:

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/2 vid none
```

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/2 default
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 20.1.2.9 netmask 255.255.255.0
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 20.1.5.0 netmask 255.255.255.0 gateway 20.1.2.1 channel "ch1" wan
```

```
PureFlow (A) > add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.2.1 channel "ch1" wan
```

```
PureFlow (A) > add route target 43.12.3.0 netmask 255.255.255.0 gateway 20.1.2.1 channel "ch1" wan
```

```
PureFlow (A) > add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.2.1 channel "ch1" wan
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.2.2 channel "ch1" lan
```

```
PureFlow (A) > add route target 32.3.2.0 netmask 255.255.255.0 gateway 20.1.2.2 channel "ch1" lan
```

```
PureFlow (A) > add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.2.2 channel "ch1" lan
```

<Scenario setting>

```
PureFlow (A) > update scenario "/port1" action aggregate peak_bw 4G
```

```
PureFlow (A) > add scenario "/port1/dc-ny" action aggregate min_bw 1G peak_bw 3G
```

```
PureFlow (A) > add scenario "/port1/dc-ny/server1" action aggregate min_bw 1G peak_bw 1G class 2
```

```
PureFlow (A) > add scenario "/port1/dc-ny/server2" action aggregate min_bw 1G peak_bw 1G class 2
```

```
PureFlow (A) > add scenario "/port1/dc-ny/server1/woc1" action wan-accel peer 20.1.5.9
```

```
PureFlow (A) > add scenario "/port1/dc-ny/server2/woc2" action wan-accel peer 20.1.5.9
```

```
PureFlow (A) > add scenario "/port1/dc-ny/woc3" action wan-accel peer 20.1.5.9
```

<Setting of acceleration target on the LAN side and QoS filter>

```
PureFlow (A) > add filter scenario "/port1/dc-ny" filter "F0" ipv4
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/server1" filter "F1-2" ipv4 sip 30.2.1.7
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/server2" filter "F2-2" ipv4 sip 32.3.2.3
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/server1/woc1" filter "F1-3" ipv4 sip 30.2.1.7
```

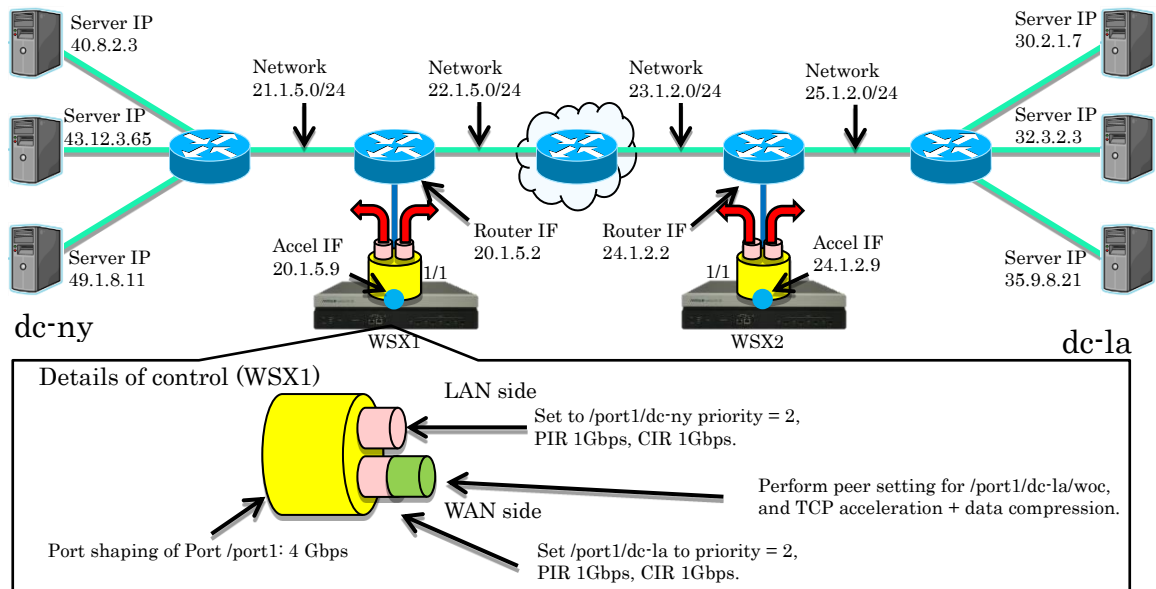
```
PureFlow (A) > add filter scenario "/port1/dc-ny/server2/woc2" filter "F2-3" ipv4 sip 32.3.2.3
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/woc3" filter "F3-2" ipv4 sip 35.9.8.21
```

<Application to be accelerated: SMB>

```
PureFlow (A) > add apl-accel scenario "/port1/dc-ny/server2/woc2" protocol smb
```

[Case 3] Out of Path connection



WSX1 setting

Execute the following commands:

<Scenario tree mode setting>

```
PureFlow (A) > set scenario tree mode outbound
```

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/1 vid none
```

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/1 default
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 20.1.5.9 netmask 255.255.255.0
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 24.1.2.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 30.2.1.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 32.3.2.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 35.9.8.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" wan
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 40.8.2.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" lan
```

```
PureFlow (A) > add route target 43.12.3.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" lan
```

```
PureFlow (A) > add route target 49.1.8.0 netmask 255.255.255.0 gateway 20.1.5.2 channel "ch1" lan
```

<Scenario setting>

```
PureFlow (A) > update scenario "/port1" action aggregate peak_bw 4G
```

```
PureFlow (A) > add scenario "/port1/dc-ny" action aggregate min_bw 1G peak_bw 1G class 2
```

```
PureFlow (A) > add scenario "/port1/dc-la" action aggregate min_bw 1G peak_bw 1G class 2
```

```
PureFlow (A) > add scenario "/port1/dc-la/woc" action wan_accel peer 24.1.2.9
```

<Rule list setting>

```
PureFlow (A) > add rulelist group "ny-serv" ipv4
```

```
PureFlow (A) > add rulelist entry "ny-serv" ipv4 40.8.2.3
```

```
PureFlow (A) > add rulelist entry "ny-serv" ipv4 43.12.3.65
```

```
PureFlow (A) > add rulelist entry "ny-serv" ipv4 49.1.8.11
```

<Setting of acceleration target on the LAN side and QoS filter>

```
PureFlow (A) > add filter scenario "/port1/dc-ny" filter "F1-lan" ipv4 dip list "ny-serv"
```

```
PureFlow (A) > add filter scenario "/port1/dc-la" filter "F2-wan" ipv4 sip list "ny-serv"
```

```
PureFlow (A) > add filter scenario "/port1/dc-la/woc" filter "F2-wan-1" ipv4 sip list "ny-serv"
```

WSX2 setting

Execute the following commands:

<Scenario tree mode setting>

```
PureFlow (A) > set scenario tree mode outbound
```

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/1 vid none
```

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/1 default
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 24.1.2.9 netmask 255.255.255.0
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 20.1.5.0 netmask 255.255.255.0 gateway 24.1.2.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 40.8.2.0 netmask 255.255.255.0 gateway 24.1.2.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 43.12.3.0 netmask 255.255.255.0 gateway 24.1.2.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 49.1.8.0 netmask 255.255.255.0 gateway 24.1.2.2 channel "ch1" wan
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 30.2.1.0 netmask 255.255.255.0 gateway 24.1.2.2 channel "ch1" lan
```

```
PureFlow (A) > add route target 32.3.2.0 netmask 255.255.255.0 gateway 24.1.2.2 channel "ch1" lan
```

```
PureFlow (A) > add route target 35.9.8.0 netmask 255.255.255.0 gateway 24.1.2.2 channel "ch1" lan
```

<Scenario setting>

```
PureFlow (A) > update scenario "/port1" action aggregate peak_bw 4G
```

```
PureFlow (A) > add scenario "/port1/dc-la" action aggregate min_bw 1G peak_bw 1G class 2
```

```
PureFlow (A) > add scenario "/port1/dc-ny" action aggregate min_bw 1G peak_bw 1G class 2
```

```
PureFlow (A) > add scenario "/port1/dc-ny/woc" action wan_accel peer 20.1.5.9
```

<Rule list setting>

```
PureFlow (A) > add rulelist group "la-serv" ipv4
```

```
PureFlow (A) > add rulelist entry "la-serv" ipv4 30.2.1.7
```

```
PureFlow (A) > add rulelist entry "la-serv" ipv4 32.3.2.3
```

```
PureFlow (A) > add rulelist entry "la-serv" ipv4 35.9.8.21
```

<Setting of acceleration target on the LAN side and QoS filter>

```
PureFlow (A) > add filter scenario "/port1/dc-la" filter "F1-lan" ipv4 dip list "la-serv"
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny" filter "F2-wan" ipv4 sip list "la-serv"
```

```
PureFlow (A) > add filter scenario "/port1/dc-ny/woc" filter "F2-wan-1" ipv4 sip list "la-serv"
```

8.14 Advanced Settings

This device provides the following advanced settings:

- Flow
- Queue
- Communication gap mode
- Traffic acceleration bypass
- Traffic acceleration redundancy
- TCP-FEC function
- TCP congestion control function
- Remarking function

8.14.1 Flow

A flow is the minimum identifiable unit in the device. Traffic is considered as a group consisting of multiple flows.

This device registers a flow to transfer a packet when it receives the packet. The registered flow stores the packet in the queue according to the operation set in the filter, and controls the traffic.

There are four types of flows: BridgeControl flow, EthernetType flow, IPv4 flow, and IPv6 flow.

(1) BridgeControl flow

The BridgeControl flow uses the Bridge-ctrl filter for identification. It aggregates frames of which destination MAC address is within the range of 01-80-C2-00-00-00 to 01-80-C2-00-00-FF into one flow for each input port.

(2) EthernetType flow

The EthernetType flow uses the Ethernet filter for identification. It identifies flows based on the following Ethernet fields:

- VLAN ID (whether the VLAN Tag is added is also identified)
- Ethernet Type.

(3) IPv4/IPv6 flow

The IPv4/IPv6 flow is identified by the IPv4/IPv6 filter. It identifies flows based on the following IP packet fields:

- VLAN ID (whether the VLAN Tag is added or not is also identified)
- Source IP address (SIP)
- Destination IP address (DIP)
- Protocol number
- Source port number (Sport)
- Destination port number (Dport)

Notes:

1. A maximum of 1,280,000 flows (a total of BridgeControl flows, EthernetType flows, and IPv4/IPv6 flows) can be created in the device and used for bandwidth control.
2. BridgeControl flows are created on a one-per-port basis.
3. For fragmented packets, ensure that all fragmented packets go through this device. If there is no first packet of the fragmented packets, the subsequent packets are not forwarded because its flow is not identified.

8.14.2 Queues

This device assigns a queue to each flow, and stores a received packet in the assigned queue. The packet stored in the queue is scheduled and transferred for traffic control.

(1) Default queue

In a level n scenario, this queue is used for transferring flows not corresponding to a lower level n scenario under it. The default queue is the best effort class (class 8).

Flows that match a certain level filter but do not match a lower level filter under it are assigned to the default queue to control the traffic.

For example, when the guaranteed bandwidth is set to 100 Mbit/s in a Level 2 scenario, the operation will be as follows:

Assuming that the following filters are registered to this device:

- Level 2 filter
Source IP address: 192.168.0.0 - 192.168.255.255
Destination IP address: 192.168.0.0 - 192.168.255.255
- Level 3 filter
Source IP address: 192.168.10.0 - 192.168.10.255
Destination IP address: 192.168.10.0 - 192.168.10.255

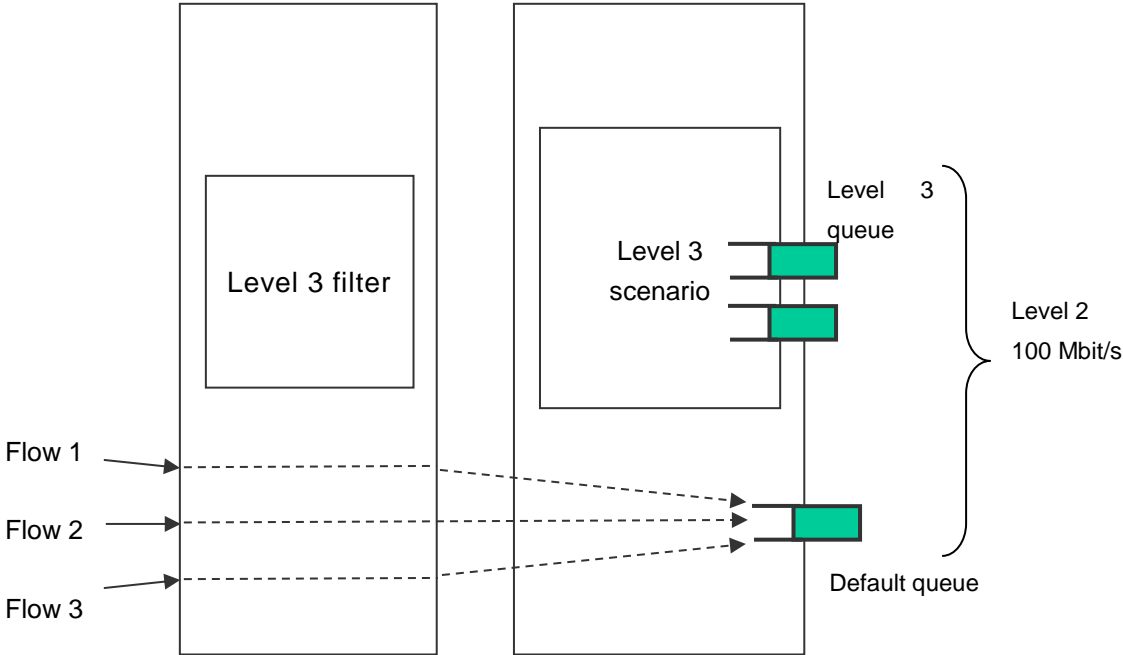
Also, assume the following three types of traffic were input:

- Traffic from 192.168.1.1 to 192.168.1.100 (flow 1)
- Traffic from 192.168.1.1 to 192.168.1.150 (flow 2)
- Traffic from 192.168.1.1 to 192.168.1.200 (flow 3)

These flows match the level 2 filter but not the level 3 filter, and therefore packets are stored in the default queue.

- Total of 100 Mbit/s for flows 1 to 3

A total bandwidth of 100 Mbit/s is guaranteed as the level 2 scenario.



Note that when a flow assigned to a level 3 queue of a high priority class is active, 100 Mbit/s bandwidth is not guaranteed for the total of all flows assigned to the default queue.

(2) Aggregate queue (level n queue)

The level n scenario in Aggregate queue mode is a method to aggregate multiple flows that match the level n filter into the level n queue.

All flows that match the level n filter and the lower level n filter under it are assigned to the same level n queue to control the traffic.

For example, when the source IP address is 192.168.10.1, the destination IP addresses are 192.168.10.100, 192.168.10.150, and 192.168.10.200, and the maximum bandwidth of the level n scenario aggregate queue is set to 10 Mbit/s, the operation is as follows:

Assuming that the following filters are registered to this device:

- Level 2 filter
Source IP address: 192.168.0.0 - 192.168.255.255
Destination IP address: 192.168.0.0 - 192.168.255.255
- Level 3 filter
Source IP address: 192.168.10.0 - 192.168.10.255
Destination IP address: 192.168.10.0 - 192.168.10.255

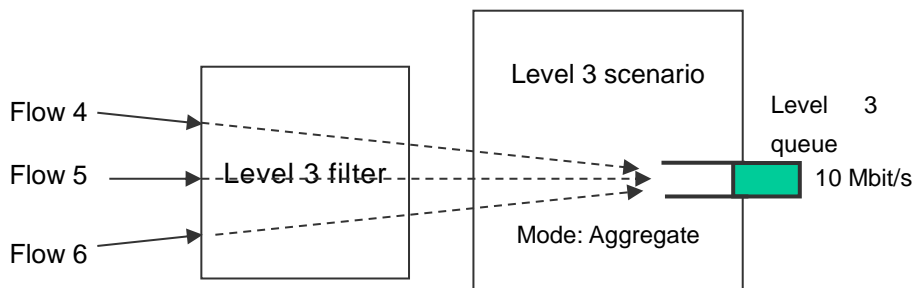
Also, assume the following three types of traffic were input:

- Traffic from 192.168.10.1 to 192.168.10.100 (flow 4)
- Traffic from 192.168.10.1 to 192.168.10.150 (flow 5)
- Traffic from 192.168.10.1 to 192.168.10.200 (flow 6)

These flows match the level 2 filter and the level 3 filter, and therefore packets are stored in the Level 3 (aggregate) queue.

- Total of 10 Mbit/s for flows 4 to 6

A total bandwidth of 10 Mbit/s is used as the level 3 scenario.



(3) Individual queue (level n queue)

The level n scenario in Individual queue mode is a method to assign individual level n queues to multiple flows that match the level n filter.

All flows that match the level n filter are separately assigned to individual level n queues to control the traffic. A lower level scenario can be registered but flows are not assigned to lower level scenarios of the individual scenario.

For example, when the source IP address is 192.168.20.1, the destination IP addresses are 192.168.20.100, 192.168.20.150, and 192.168.20.200, and the maximum bandwidth of each level n scenario individual queue is set to 10 Mbit/s, the operation is as follows:

Assuming that the following filters are registered to this device:

- Level 2 filter
Source IP address: 192.168.0.0 - 192.168.255.255
Destination IP address: 192.168.0.0 - 192.168.255.255
- Level 3 filter
Source IP address: 192.168.20.0 - 192.168.20.255
Destination IP address: 192.168.20.0 - 192.168.20.255

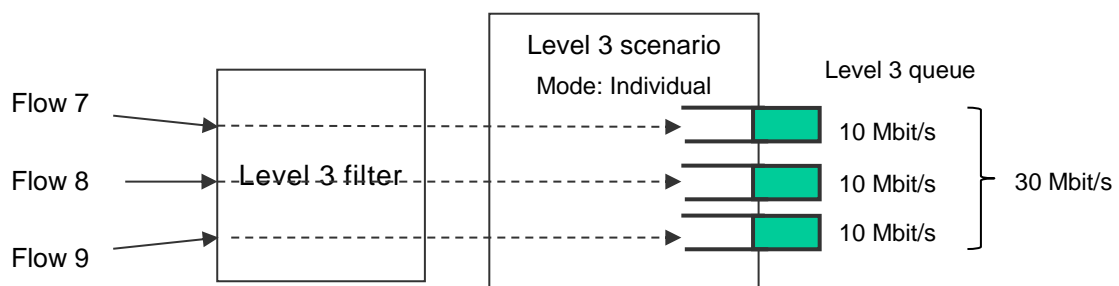
Also, assume the following three types of traffic were input:

- Traffic from 192.168.20.1 to 192.168.20.100 (flow 7)
- Traffic from 192.168.20.1 to 192.168.20.150 (flow 8)
- Traffic from 192.168.20.1 to 192.168.20.200 (flow 9)

These flows match the level 2 filter and the level 3 filter, and therefore packets are stored in the Level 3 (individual) queues.

- Flow 7 is 10 Mbit/s
- Flow 8 is 10 Mbit/s
- Flow 9 is 10 Mbit/s

A total bandwidth of 30 Mbit/s is used as the level 3 scenario.

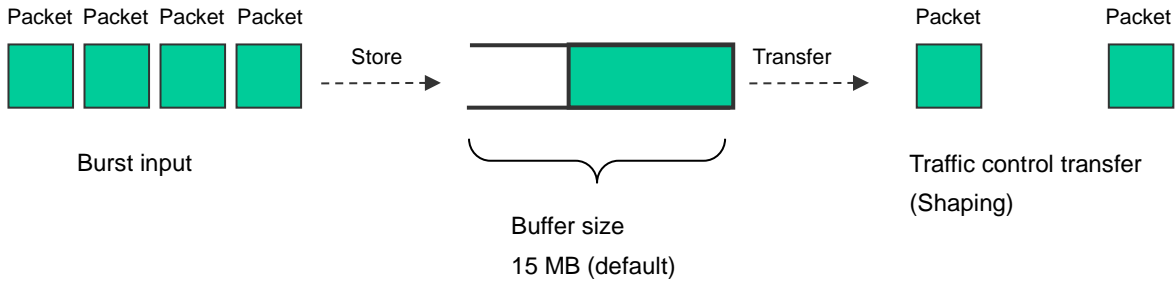
**Note:**

In monitoring manager 2, the scenario of the individual queue mode is displayed as one queue in the same way as the aggregate queue mode. The individual queue is not displayed.

(4) Buffer size

The buffer size can be set to the level n queue.

Buffer size is the allowable input burst length for the queue. It is the number of bytes that can be stored in the queue when receiving burst packets.



Stores packets up to 15 MB.
If 15 MB is exceeded, the packets are discarded.

When the input burst length exceeds the buffer size, packets are discarded. If packets are discarded due to a small buffer size, set the buffer size for the level n scenario (traffic attribute).

To check whether the packets have been discarded, see the queue statistics. (For details, see “Chapter 12 Statistics”.)

Specify the buffer sizes (bytes) of the default queue and the level n queue assigned in the level n scenario.

The following commands change the buffer size of the level n queue assigned in the level n scenario:

Sample 1) Changing the buffer size for the existing Level 2 scenario to 5 MB

```
PureFlow (A) > update scenario "/port1/Tokyo" action aggregate bufsize 5M
```

Sample 2) Changing the buffer size for the existing Level 3 scenario to 2 MB

```
PureFlow (A) > update scenario "/port1/Tokyo/Shinjuku" action aggregate bufsize 2M
```

(5) Class

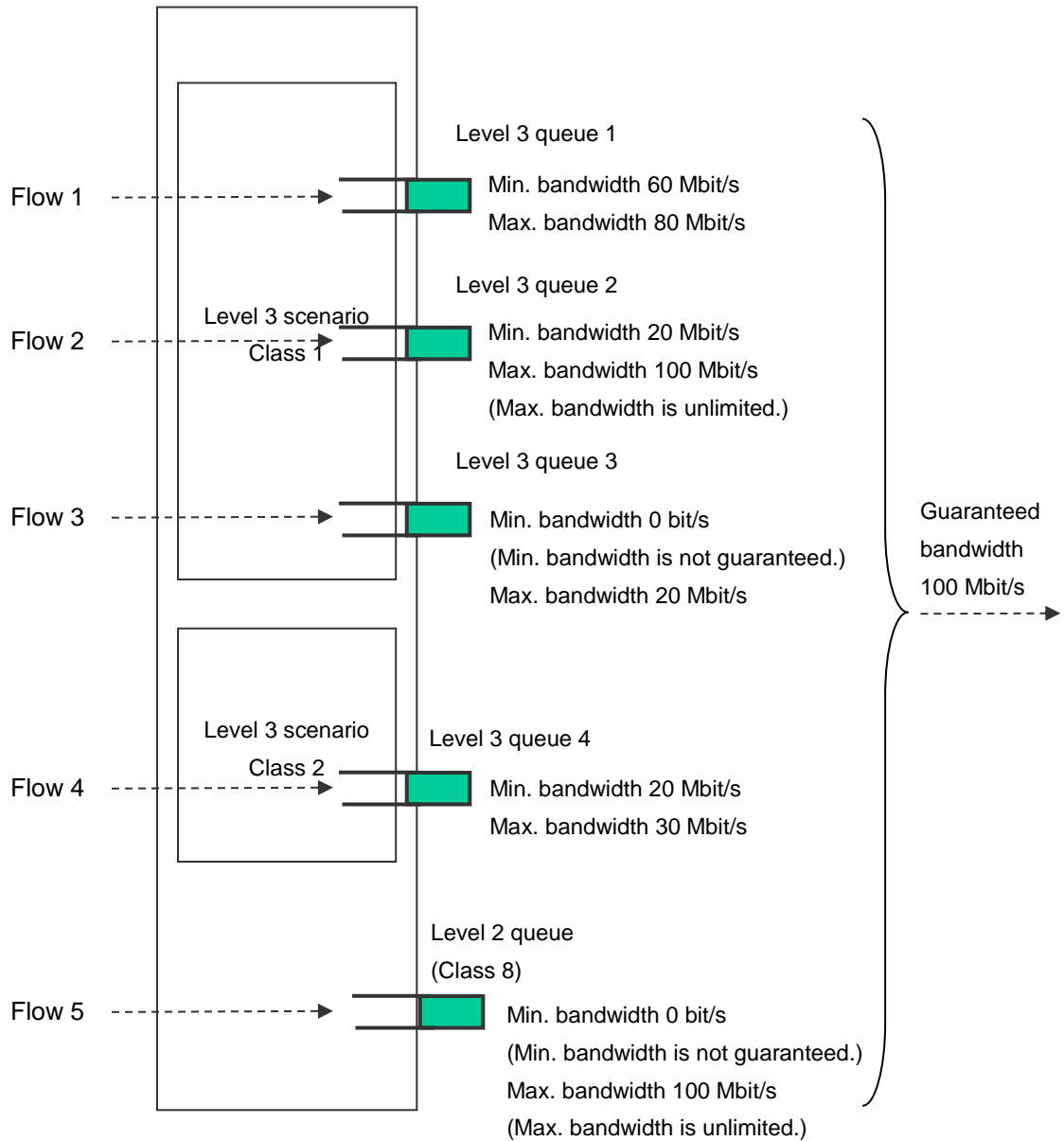
A class (queue priority) can be specified for Level 2 or lower queues.

This device uses a traffic control method in which queues of 8 classes (Class 1 to 8) are output in order of priority (Strict Priority).

The Strict Priority operation is as follows:

Assuming the Level 2 and 3 queues are assigned to this device:

- Level 2 queue (class 8, guaranteed bandwidth 100 Mbit/s)
- Level 3 queue 1 (class 1, minimum bandwidth 60 Mbit/s, maximum bandwidth 80 Mbit/s)
- Level 3 queue 2 (class 1, minimum bandwidth 20 Mbit/s, maximum bandwidth unlimited)
- Level 3 queue 3 (class 1, minimum bandwidth not guaranteed, maximum bandwidth 20 Mbit/s)
- Level 3 queue 4 (class 2, minimum bandwidth 20 Mbit/s, maximum bandwidth 30 Mbit/s)



- a) For the Level 2 scenario, the bandwidth is guaranteed.

For example, 100 Mbit/s is guaranteed for flows in the Level 2 scenario even when there are 990 Mbit/s flows in other scenarios.

However, if the total of guaranteed bandwidths assigned to Level 2 scenarios exceeds the Level 1 scenario bandwidth, the Level 2 scenario bandwidth is not guaranteed.

- b) For flows assigned to the Level 3 queue with the minimum bandwidth guaranteed, the minimum bandwidth is guaranteed.

For example, even when Flow 3 (100 Mbit/s) is active, Flow 1 (60 Mbit/s) and Flow 2 (20 Mbit/s) are controlled as 60 Mbit/s and 20 Mbit/s traffic, respectively.

However, if the total of minimum bandwidths assigned to Level 3 scenarios exceeds the Level 2 scenario guaranteed bandwidth, the Level 3 scenario minimum bandwidth is not guaranteed.

- c) If multiple Level 3 queues with different classes are assigned to the same Level 2 scenario, the minimum bandwidth is not guaranteed for Level 3 queue flows with lower priority. For Level 3 queues with lower priority classes, the traffic is controlled in the available bandwidth of the higher priority class.

For example, when Flow 1 (60 Mbit/s), Flow 2 (20 Mbit/s), and Flow 3 (15 Mbit/s) (all class 1), and Flow 4 (20 Mbit/s) (class 2) are active, Flow 4 is controlled as 5 Mbit/s traffic.

- d) Flows assigned to the Level 3 queue with the minimum bandwidth limited are controlled within their maximum bandwidth.

For example, when Flow 3 (30 Mbit/s) is active, Flow 3 is controlled as 20 Mbit/s traffic.

Also, when the maximum bandwidth of the Level 3 queue exceeds the Level 2 scenario guaranteed bandwidth, the traffic is controlled in the Level 2 scenario guaranteed bandwidth.

- e) Flows assigned to the Level 3 queue with the maximum bandwidth unlimited are controlled in the Level 2 scenario guaranteed bandwidth.

For example, when Flow 2 (120 Mbit/s) is active, Flow 2 is controlled as 100 Mbit/s traffic.

By prioritizing the Level 3 queues, packets stored in higher priority class queues are transferred on a priority basis, and thus fluctuation is smaller than the lower priority classes. To prioritize Level 3 queues, set the class in the Level 3 scenario.

The following command can change the Level 3 scenario class:

Sample) Setting class 1 for the existing Level 3 scenario

```
PureFlow (A) > update scenario "/port1/Tokyo/Shinjuku" action aggregate class 1
```

Note:

A change of the scenario class by the CLI command, etc. is applied after the target scenario sends 1 packet. If other scenarios with higher priority dominate the bandwidth, the target scenario cannot send packets and thus the class setting is not applied. Change the class when the bandwidth is available (the maximum bandwidth is not reached).

8.14.3 Communication gap mode

For Ethernet, inter-frame gaps and preambles are inserted to continuously transmit frames. When setting the bandwidth for traffic attributes (scenario, Network port), you can select whether to control traffic including the gaps and preambles (the target will include the entire network bandwidth) or to control traffic excluding them (the target will only include frames). This setting is applied to the entire device.

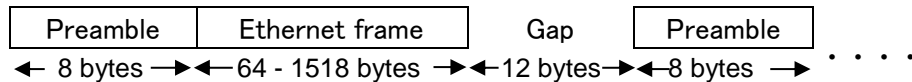


Figure Gaps and preambles of the Ethernet frame

The following CLI commands are available for communication gap mode:

<pre>set bandwidth mode {gap [<size>] no_gap}</pre>	<p>Enables/disables inter-frame gaps and preambles in the communication bandwidth settings. The default value is “gap (enabled)”.</p> <p>If “gap” is specified, inter-frame gaps and preambles can be included in the bandwidth, with the specified size. Valid values for the size are from -100 [bytes] to +100 [bytes]. If the size is set to 0, the behavior is the same as no_gap.</p>
------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The following is a command execution example:

```
PureFlow (A) > set bandwidth mode gap
PureFlow (A) >
```

When communication gap mode is enabled, control by the traffic attribute (scenario, Network port) bandwidth setting value includes inter-frame gaps and preambles. With this setting, the bandwidth setting value is the same as the physical line, which is effective for avoiding congestion in the output WAN line bandwidth and for traffic control on a priority basis.

When communication gap mode is disabled, control by the traffic attribute (scenario, Network port) bandwidth setting value targets only the Ethernet frames as the data rate, and does not include the inter-frame gaps and preambles. This setting is generally effective for controlling the contents rate by performing actions such as smoothing to avoid bursts of audio and video contents that are indicated by a data rate that excludes inter-frame gaps and preambles and controlling the reception rate control for servers.

Note that communication gaps need to be considered for the bandwidth value since the traffic attribute (scenario, Network port) bandwidth value output rate is different from the line bandwidth when the communication gap mode is disabled. For example, if the line bandwidth is 100 Mbit/s, the setting value should be approx. 76 Mbit/s (100 Mbit/s x 64 bytes/84 bytes) to transfer all frames (64 to 1522 bytes) without omission. In this case, all frames are limited to 76 Mbit/s regardless of the length, and the longer frame length results in less effective transfer. To make better use of the line bandwidth, enable the communication gap mode to set the bandwidth including inter-frame gaps.

Note:

This set value of the communication gap mode applies to each packet when receiving the packet. The value does not apply to the packet remaining in the scenario buffer when changing the communication gap mode. Therefore, the changed communication gap mode is reflected after the packet remaining at the change is discharged.

8.14.4 Traffic acceleration bypass

If opposing devices cannot be connected or if RTT (Round Trip Time) between the opposing devices is less than a certain value during the traffic acceleration, the traffic acceleration can be bypassed. The bypass transfer is performed without traffic acceleration.

The following parameters that can be set for traffic acceleration bypass:

Parameter	Setting range	Optional/required	Description
Enabling/Disabling function	enable disable	Required	<p>Specifies enabling/disabling the auto bypass function for the traffic acceleration.</p> <p>When enabled, it enters the bypass transfer state under the following conditions.</p> <ul style="list-style-type: none"> • When the TCP connection error occurs • When the RTT value is less than the threshold value during the connection with TCP • The ICMP communication error occurs upon the Keep Alive monitoring <p>The default value is enable.</p> <p>Applies to all of the scenarios in acceleration mode.</p>
Bypass recovery time	1 to 600 seconds	Required	<p>Sets the bypass recovery time of the auto bypass function for the traffic acceleration.</p> <p>Attempts the traffic acceleration again for a new TCP session after this specified time has passed in the case of the bypass transfer state.</p> <p>The default value is 60 seconds.</p> <p>Applied to all acceleration mode scenarios.</p>
Auto-bypass RTT threshold value for traffic acceleration (bypass-thresh)	0 to 10000 milliseconds	Optional	<p>When omitted: 0</p> <p>Specifies a threshold value of the RTT (Round Trip Time: Reciprocation delay time) of the auto bypass function in milliseconds for the traffic acceleration.</p> <p>The default value is 0 second. When 0 is specified, bypass operation due to RTT bypass transfer is not performed, but bypass operation due to a TCP connection error is performed.</p> <p>This parameter can be specified for each of the acceleration mode scenarios.</p>
Enabling/disabling the Keep Alive monitoring of the auto bypass function for the traffic acceleration (bypass-keepalive)	enable disable	Optional	<p>When omitted: disable</p> <p>Specifies enabling/disabling the Keep Alive monitoring of the auto bypass function for the traffic acceleration.</p> <p>Can be specified in up to 100 acceleration mode scenarios.</p> <p>In the case of enable, the opposing device specified for the applicable scenario is monitored by ICMP via communication. Sets under the bypass transfer state when the communication error occurs. Continues the communication monitoring under the bypass transfer state that is maintained until the communication error is recovered.</p> <p>This parameter can be specified for each of the acceleration mode scenarios.</p>

The CLI commands related to the bypass settings for the traffic acceleration are shown below:

set wan-accel bypass status {enable disable}	Specifies enabling/disabling the auto bypass function for the traffic acceleration.
set wan-accel bypass recoverytime <duration>	Sets the bypass recovery time of the auto bypass for the traffic acceleration.
add scenario <scenario_name> action wan-accel peer <IP_address> second-peer <IP_address> [dport <dport>] [vid <vid>] [inner-vid <VID>] [compression {enable disable}] [tcp-mem {auto <size>}] [cc-mode {normal semi-fast fast}] [bypass-thresh <rtt>] [bypass-keepalive {enable disable}] [fec {enable disable}] [block-size <size>] [data-block-size <size>] [fec-session <session>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [bufsize <bufsize>] [scenario <scenario_id>]	Registers a scenario of the acceleration mode. Specifies a bypass threshold value. Specifies the RTT threshold value (bypass-thresh) and enabling/disabling the Keep Alive monitoring (bypass-keepalive) of the auto bypass for the traffic acceleration.
update scenario <scenario_name> action wan-accel [vid <vid>] [inner-vid <VID>] [compression {enable disable}] [tcp-mem {auto <size>}] [cc-mode {normal semi-fast fast}] [bypass-thresh <rtt>] [bypass-keepalive {enable disable}] [fec {enable disable}] [block-size <size>] [data-block-size <size>] [fec-session <session>] [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [bufsize <bufsize>]	Changes a scenario of the acceleration mode. Specifies the RTT threshold value (bypass-thresh) and enabling/disabling the Keep Alive monitoring (bypass-keepalive) of the auto bypass for the traffic acceleration.
switch wan-accel bypass force {enable disable} all	Specifies enabling/disabling the forcible bypass function for the traffic acceleration. When enabled, this equipment enters the bypass transfer state forcibly..
switch wan-accel bypass force {enable disable} scenario <scenario_name>	Specifies enabling/disabling the forcible bypass function for the traffic acceleration of the specified scenario name. When enabled, this equipment enters the bypass transfer state forcibly.
show wan-accel bypass	Displays the bypass information of traffic acceleration.

The auto bypass RTT threshold value can be set in the action mode "wan-accel" of the "add scenario" command. It performs bypass transfer when the RTT value measured upon TCP connection is less than the specified RTT threshold value. Normally, set this value to 6 milliseconds. High-speed transfer can be performed without applying the traffic acceleration when RTT value is within 6 milliseconds. The traffic acceleration of this device functions effectively when the RTT value exceeds 6 milliseconds.

Command execution examples are shown below:

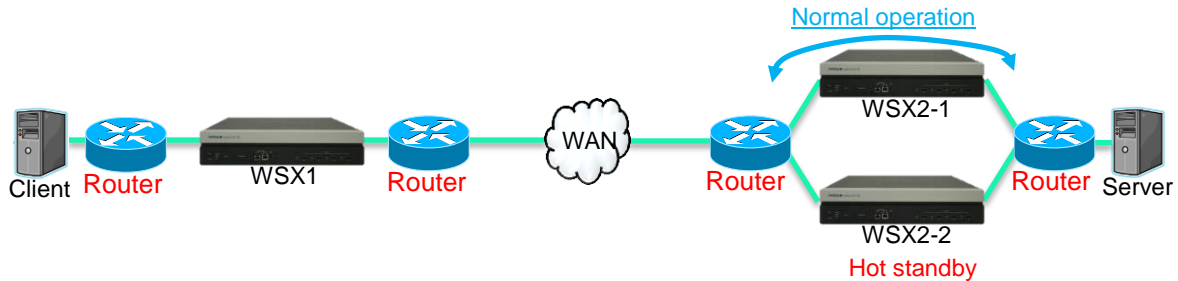
```
PureFlow (A) > set wan-accel bypass status enable  
PureFlow (A) > set wan-accel bypass recoverytime 30
```

The “show scenario info name” command shows the information on the traffic acceleration bypass.

Traffic acceleration bypass Parameter	Displayed contents
Status	Displays the state (enabling/disabling) of the auto bypass function for the traffic acceleration.
Recovery time	Displays the time until the scenario in the bypass transfer state retries traffic acceleration.
State	Displays the current scenario state of the auto bypass function for the traffic acceleration Standby: Waiting for traffic input Measuring: Measuring RTT with the connection connected Acceleration: Applying the traffic acceleration Bypass: Bypass transfer in progress Force Bypass: Forcible transfer in progress
Threshold RTT	Displays the RTT threshold value.
Minimum RTT	Displays the minimum value of the RTT measured values. When this measured value is below the RTT threshold value, traffic acceleration is stopped, and this equipment enters the bypass transfer state.
Low RTT	Displays the detection state for the values that are under the RTT threshold value. not detected: The RTT lower than the RTT threshold value has not been detected. detected: The RTT lower than the RTT threshold value has been detected.
Connection Error	Displays the detection state for the TCP connection error. not detected: The TCP connection error has not been detected. detected: The TCP connection error has been detected.
Keep Alive	Displays the Keep Alive monitoring function state (enabling/disabling) of the auto bypass function for the traffic acceleration.
Keep Alive State	Displays the Keep Alive monitoring state. Alive: Indicates the normality of the communication with Peer. Timeout: Indicates the timeout of the communication with Peer. -----: Indicates that the Keep Alive monitoring is not performed.
Acceleration Trans	Displays the accumulated number of times the traffic acceleration scenario changed to the “Acceleration” status.
Bypass Trans	Displays the accumulated number of times the traffic acceleration scenario changed to the “Bypass” status.

8.14.5 Traffic acceleration redundancy

For the traffic acceleration function, this device can be used in the device redundancy configuration of the hot-standby type. For the redundancy configuration, establish the network of the following configuration.



For WSX1 and WSX2-1, traffic acceleration is performed under the normal operation (the WSX2-1 is referred to as Primary Peer, in this case.) The WSX2-2 waits in the hot standby state (the WSX2-2 is referred to as Secondary Peer, in this case). Normally, the WSX1 configures the acceleration tunnel with Primary Peer that is switched to Secondary Peer when the device for Primary Peer fails or when communication cannot be established with the device for Primary Peer due to error occurred in the WAN-side path.

The Secondary Peer is set for the WSX on the Client side because the WSX on the Client side controls the use of either the Primary or the Secondary. In the case of the configuration above, "second-peer" (IP address for the opposing device of the Secondary Peer) is set in the WSX1 scenario. In addition, it is necessary to set "second-peer" bi-directionally in the case of 2-to-2 configuration with a total of four WSXs.

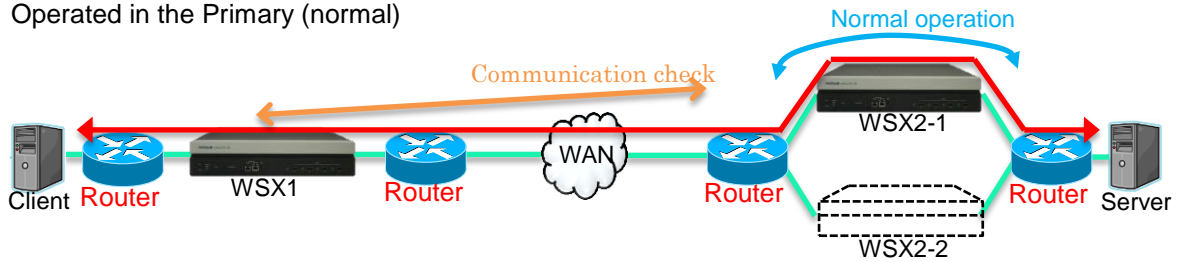
Note:

For the redundancy configuration, please note the following.

- (1) The path to the opposing device (peer) must be controlled on layer 3.
- (2) The WSX on the path used by the routers in normal operation must be specified as the Primary Peer. (For example, specify the path that prioritizes the Primary Peer side by adjusting the OSPF path cost when the routing protocol is OSPF.)
- (3) Must be enable the link down transfer function.(For how to set, see “chapter9 Link-down Transfer”.)

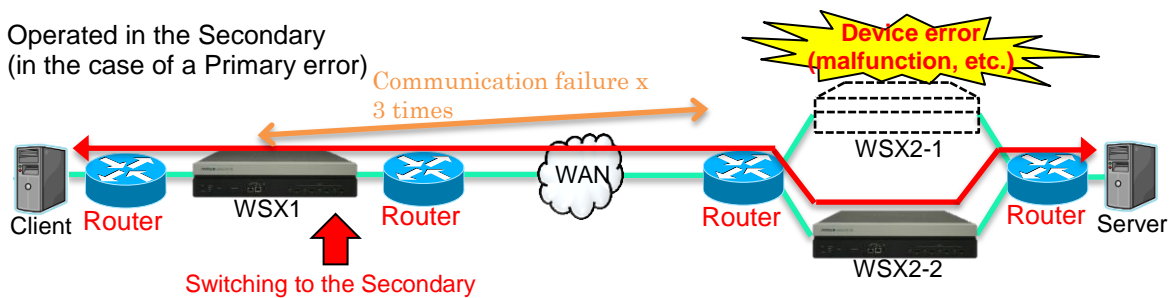
Operation at switching is shown below:

Operated in the Primary (normal)



Normally, the traffic acceleration tunnel is created between the WSX1 and the Primary Peer device. At this time, the WSX1 checks the communication once every 3 seconds by using the Primary Peer device and ICMP.

Operated in the Secondary (in the case of a Primary error)

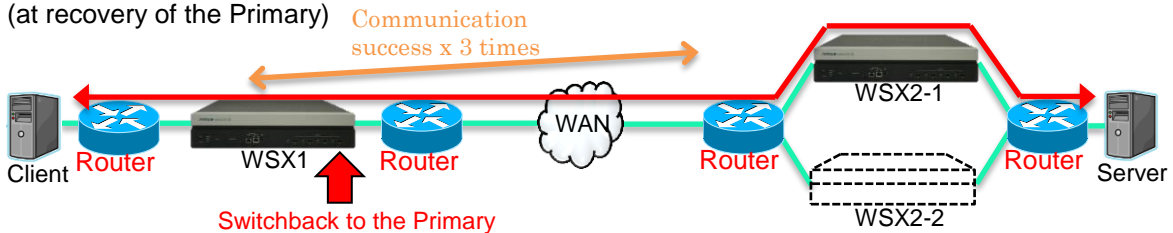


If the communication fails three times in a row, the WSX1 determines that it cannot be communicated to the Primary Peer device, and then connects to the Secondary Peer device. After this, the acceleration tunnel is constructed between the WSX1 and the Secondary Peer device.

Note 1:

The TCP session between the WSX1 and the Primary Peer device during traffic acceleration does not switch to the Secondary Peer device.

Switchback to the Primary device (at recovery of the Primary)



The WSX1 continues to check the communication to the Primary Peer device also while the Secondary Peer device is connected. When the communication can be checked successfully three times in a row after the Primary Peer device recovers from the error (malfunction, etc.), WSX1 judges that it can be connected to Primary Peer device, and the connection is switched back to the Primary Peer device. After that, the Primary Peer device and TCP acceleration tunnel is recovered.

Note 2:

The TCP session between the WSX1 and the Secondary Peer device during traffic acceleration does not recover to the Primary Peer device. The tunnel is connected via the Secondary Peer device until the TCP session via the acceleration tunnel finishes.

Note 3:

When the redundancy configuration and traffic acceleration bypass function are used at the same time, perform the operation according to the following procedure.

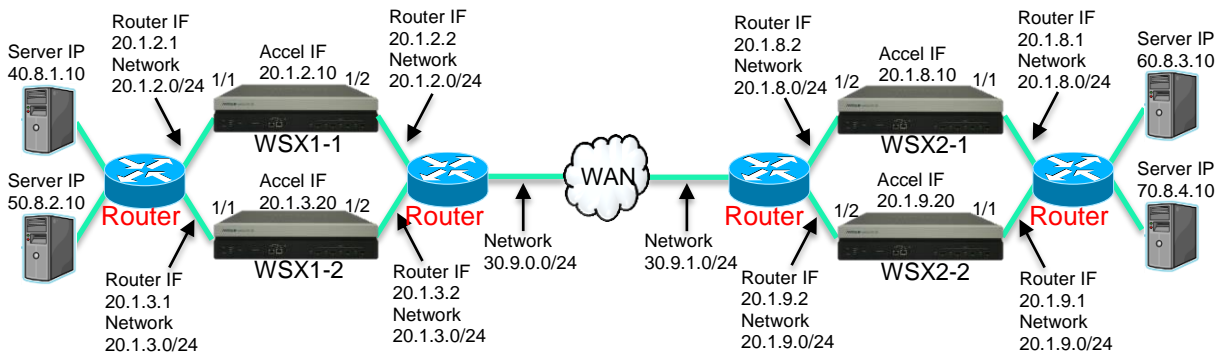
- (1) A failure occurs in the Primary Peer: The redundancy configuration function switches from the Primary to the Secondary Peer.
- (2) A failure also occurs in the Secondary Peer: Auto bypass function switches to TCP bypass transfer status.

In the case of the redundancy configuration, communication is monitored for the Primary Peer by ICMP, but not monitored for the Secondary Peer. Communication monitoring in ICMP for Secondary Peer can be performed by enabling the auto bypass Keep Alive monitoring during the communication with the Secondary Peer

The commands and parameters used for the redundancy configuration are listed below: For details of how to set, see “STEP 4 Set the scenario.” in “8.9 Setting Procedure”.

Command	Parameter	Description
add scenario	second-peer <IP_address>	Specifies the Secondary Peer device IP address.
show scenario info name	<scenario name>	Allows you to check connection of either the Primary or Secondary Peer by execution for the scenario that specified second-peer.
show syslog	None	The Primary and Secondary connection conditions can be checked with the system log. When switching from the Primary Peer to the Secondary Peer, the system log is recorded as shown below: “Wan-accel scenario switched to secondary-peer. [S:#M]” When switchback from the Secondary Peer to the Primary Peer, the system log is recorded as shown below: “Wan-accel scenario switched back to primary-peer. [S:#M]” M is replaced with the name of the target scenario.

As a reference, a setting example for the redundancy configuration is shown below:



Setting of the WSX1-1

Execute the following commands:

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/2 default
```

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/2 vid none
```

<Link-down transfer setting>

```
PureFlow(A)> add lpt pair port 1/1 1/2
```

```
PureFlow(A)> set lpt enable
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 20.1.2.10 netmask 255.255.255.0
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 40.8.1.0 netmask 255.255.255.0 gateway 20.1.2.1 channel "ch1" lan
```

```
PureFlow (A) > add route target 50.8.2.0 netmask 255.255.255.0 gateway 20.1.2.1 channel "ch1" lan
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 20.1.8.0 netmask 255.255.255.0 gateway 20.1.2.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 20.1.9.0 netmask 255.255.255.0 gateway 20.1.2.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 60.8.3.0 netmask 255.255.255.0 gateway 20.1.2.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 70.8.4.0 netmask 255.255.255.0 gateway 20.1.2.2 channel "ch1" wan
```

<Scenario setting>

```
PureFlow (A) > add scenario /port1/woc1-1 action wan_accel peer 20.1.8.10 second-peer 20.1.9.20
```

<Filter setting>

```
PureFlow (A) > add filter scenario /port1/woc1-1 filter "F1" ipv4 sip 40.8.1.10
```

Setting of the WSX1-2

Execute the following commands:

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/2 default
```

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/2 vid none
```

<Link-down transfer setting>

```
PureFlow(A)> add lpt pair port 1/1 1/2
```

```
PureFlow(A)> set lpt enable
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 20.1.3.20 netmask 255.255.255.0
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 40.8.1.0 netmask 255.255.255.0 gateway 20.1.3.1 channel "ch1" lan
```

```
PureFlow (A) > add route target 50.8.2.0 netmask 255.255.255.0 gateway 20.1.3.1 channel "ch1" lan
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 20.1.8.0 netmask 255.255.255.0 gateway 20.1.3.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 20.1.9.0 netmask 255.255.255.0 gateway 20.1.3.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 60.8.3.0 netmask 255.255.255.0 gateway 20.1.3.2 channel "ch1" wan
```

```
PureFlow (A) > add route target 70.8.4.0 netmask 255.255.255.0 gateway 20.1.3.2 channel "ch1" wan
```

<Scenario setting>

```
PureFlow (A) > add scenario /port1/woc1-2 action wan_accel peer 20.1.8.10 second-peer 20.1.9.20
```

<Filter setting>

```
PureFlow (A) > add filter scenario /port1/woc1-2 filter "F1" ipv4 sip 50.8.2.10
```

Setting of the WSX2-1

Execute the following commands:

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/2 default
```

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/2 vid none
```

<Link-down transfer setting>

```
PureFlow(A)> add lpt pair port 1/1 1/2
```

```
PureFlow(A)> set lpt enable
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 20.1.8.10 netmask 255.255.255.0
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 60.8.3.0 netmask 255.255.255.0 gateway 20.1.8.1 channel  
"ch1" lan
```

```
PureFlow (A) > add route target 70.8.4.0 netmask 255.255.255.0 gateway 20.1.8.1 channel  
"ch1" lan
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 20.1.2.0 netmask 255.255.255.0 gateway 20.1.8.2 channel  
"ch1" wan
```

```
PureFlow (A) > add route target 20.1.3.0 netmask 255.255.255.0 gateway 20.1.8.2 channel  
"ch1" wan
```

```
PureFlow (A) > add route target 40.8.1.0 netmask 255.255.255.0 gateway 20.1.8.2 channel  
"ch1" wan
```

```
PureFlow (A) > add route target 50.8.2.0 netmask 255.255.255.0 gateway 20.1.8.2 channel  
"ch1" wan
```

<Scenario setting>

```
PureFlow (A) > add scenario /port1/woc2-1 action wan_accel peer 20.1.2.10 second-peer  
20.1.3.20
```

<Filter setting>

```
PureFlow (A) > add filter scenario /port1/woc2-1 filter "F1" ipv4 sip 60.8.3.10
```

Setting of the WSX2-2

Execute the following commands:

<Default channel setting>

```
PureFlow (A) > add channel "ch10000" lan 1/1 wan 1/2 default
```

<Channel setting>

```
PureFlow (A) > add channel "ch1" lan 1/1 wan 1/2 vid none
```

<Link-down transfer setting>

```
PureFlow(A)> add lpt pair port 1/1 1/2
```

```
PureFlow(A)> set lpt enable
```

<Network interface IP address setting>

```
PureFlow (A) > set ip interface "ch1" 20.1.9.20 netmask 255.255.255.0
```

<Route setting on the LAN side>

```
PureFlow (A) > add route target 60.8.3.0 netmask 255.255.255.0 gateway 20.1.9.1 channel  
"ch1" lan
```

```
PureFlow (A) > add route target 70.8.4.0 netmask 255.255.255.0 gateway 20.1.9.1 channel  
"ch1" lan
```

<Route setting on the WAN side>

```
PureFlow (A) > add route target 20.1.2.0 netmask 255.255.255.0 gateway 20.1.9.2 channel  
"ch1" wan
```

```
PureFlow (A) > add route target 20.1.3.0 netmask 255.255.255.0 gateway 20.1.9.2 channel  
"ch1" wan
```

```
PureFlow (A) > add route target 40.8.1.0 netmask 255.255.255.0 gateway 20.1.9.2 channel  
"ch1" wan
```

```
PureFlow (A) > add route target 50.8.2.0 netmask 255.255.255.0 gateway 20.1.9.2 channel  
"ch1" wan
```

<Scenario setting>

```
PureFlow (A) > add scenario /port1/woc2-2 action wan_accel peer 20.1.2.10 second-peer  
20.1.3.20
```

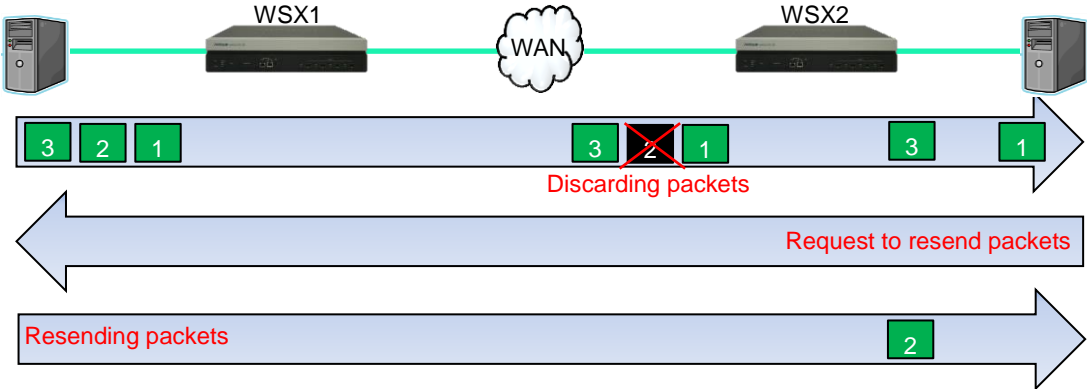
<Filter setting>

```
PureFlow (A) > add filter scenario /port1/woc2-2 filter "F1" ipv4 sip 70.8.4.10
```

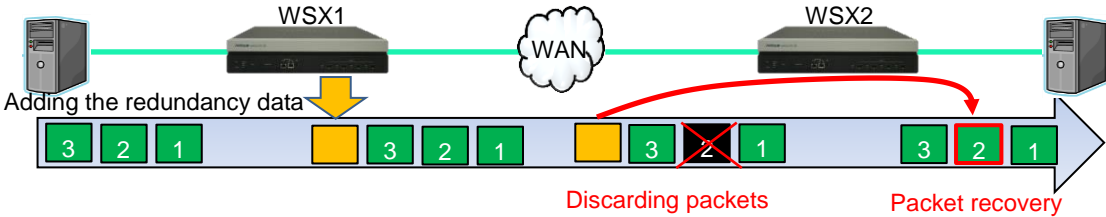
8.14.6 TCP-FEC function

This device is equipped with the TCP-FEC function. This function adds the redundancy data to the TCP packet of the traffic acceleration and performs FEC (Forward Error Correction). TCP resends the data in the case of discarding a packet. The TCP-FEC function enables the data discarded when a packet is discarded to be recovered. Therefore, it is not required to resend the data. As a result, this function sufficiently provides the performance under an environment where packets are frequently discarded. Operation in the case of discarding a packet is shown below:

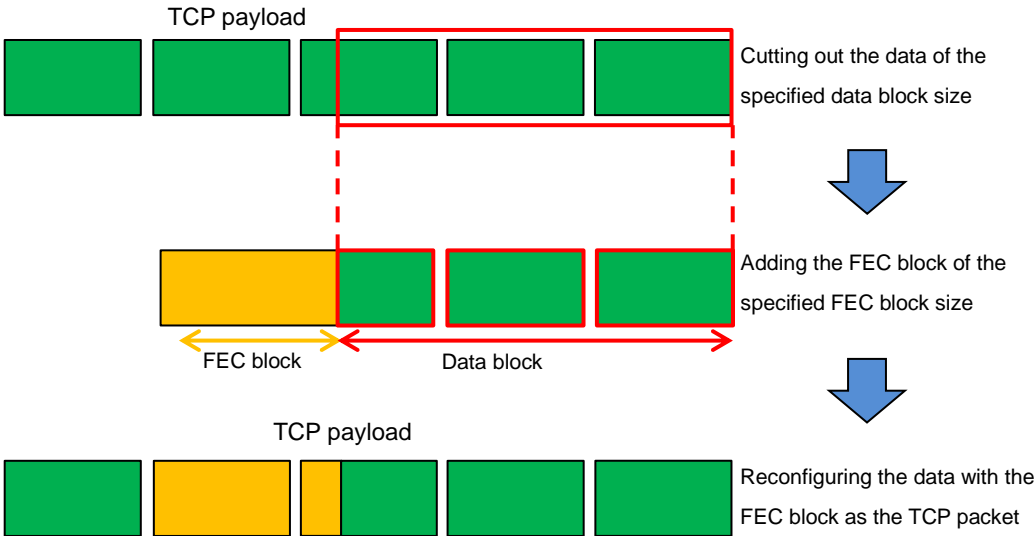
Device NOT equipped with the TCP-FEC function



Device equipped with the TCP-FEC function



To use this function, the data block size and FEC (redundancy) block size must be specified. The relationship between the data block size and FEC block size is shown below:



Note:

The possibility of recovering the data becomes higher as the redundancy ratio (= FEC block size/Data block size) is larger. However, the line efficiency becomes lower.

Set this function for each scenario. To use this function, enable the TCP-FEC function when registering or updating the scenario, and then set the parameter. The parameters related to the TCP-FEC function are shown below: For details of how to set and check the parameter, see “STEP 4 Set the scenario.” in “8.9 Setting Procedure”.

Command	Parameter	Description
add scenario update scenario	fec {enable disable}	Specifies enabling/disabling the TCP-FEC function.
	block-size <size>	Specifies the FEC block size. A value larger than the data block size cannot be specified. Additionally, specify a value that can evenly divide the data block size.
	data-block-size <size>	Specifies the data block size. A value smaller than the FEC block size cannot be specified. Additionally, specify a value that can be divided by the FEC block size.
	fec-session <session>	The TCP session by using the TCP-FEC function is called the FEC function. Limits the counts of the FEC sessions available for the scenarios that configure this parameter. Up to 1000 FEC sessions are available for the entire device. The counts of the FEC sessions are not assured in this parameter.
show scenario	name <scenario_name>	Displays the scenario information of specified scenario (TCP-FEC function related parameter).

A command execution example is shown below:

Execution example <1>: When adding the scenario with the TCP-FEC function

Parameter: FEC block size = 4 Kbytes, Data block size = 8 Kbytes, FEC session limit count: 10

```
PureFlow (A) > add scenario /port1/woc1-fec action wan-accel peer 192.168.100.11 compression
disable fec enable block-size 4k data-block-size 8k fec-session 10
```

Execution example <2>: When updating an already-registered scenario onto the scenario with the TCP-FEC function

Parameter: FEC block size = 8 Kbytes, Data block size = 24 Kbytes, FEC session limit count: 100

```
PureFlow (A) > update scenario /port1/woc1 action wan-accel compression disable fec enable
block-size 8k data-block-size 24k fec-session 100
```

Execution example <3>: When updating an already-registered scenario with the TCP-FEC function onto the normal scenario

Parameter: TCP-FEC function disabled

```
PureFlow (A) > update scenario /port1/woc1-fec action wan-accel compression enable fec disable
```

The measures of setting the TCP-FEC function are listed below. Perform fine adjustment depending on the line environment.

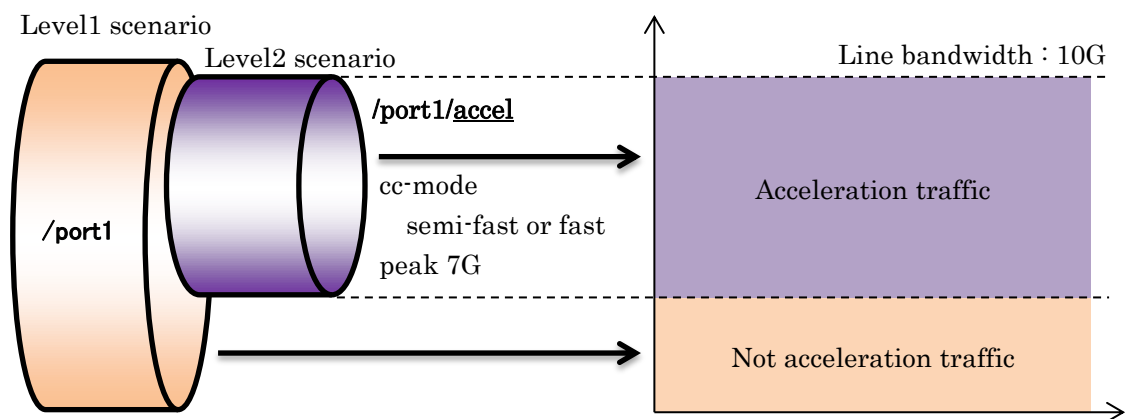
Example of line environment	Measure of setting
The TCP communication data amount is smaller.	Reduce the data block size.
It is required to emphasize the responsiveness.	Reduce the FEC block size.
The packet discarding rate of the WAN line is higher.	Reduce the data block size, and increase the FEC block size.
It is required to prevent the WAN line efficiency from being lowered.	Increase the data block size larger than the FEC block size.
The burst discarding of the WAN line occurs frequently.	Increase the FEC block size.

8.14.7 TCP congestion control function

This device is equipped with the TCP-FEC congestion control function. Congestion control function, for efficient use of communication capacity by avoiding network congestion, and adjust the transmission rate of the TCP communication. By congestion control function, transmission rate reduces when the packet loss is occurred by network congestion and increases gradually when the packet loss is not occurred. As a result, transmit rate is adjusted by following the communication capacity of the network in real time. But there is a problem in the conventional congestion control algorithms, In the network that there are much packet losses, before the communication speed is restored, by the next packet loss, and further reduce the communication speed, the average will become less than half of the communication capacity.

This device can select our own congestion control function for solving the above problems. Please specify the scenario congestion control mode (cc-mode) when using our own congestion control algorithms. Normal is standard congestion control algorithms, Semi-fast and fast is our own congestion control algorithms and communication speed will gradually decrease when a packet loss is occurred. As a result, this function sufficiently provides the performance under an environment where packets are frequently discarded.

If semi-fast or fast is selected, When the acceleration traffic and non-accelerated traffic are mixed on the same line, the rate of non-accelerated traffic will tend to be kept low. If semi-fast or fast is selected, please set the maximum bandwidth of the scenario (peak) in the following line bandwidth, and please set as non-accelerated traffic is properly transferred.



Set the congestion control function for each scenario.

Command	Parameter	Description
add scenario update scenario	cc-mode {normal semi-fast fast}	When omitted: normal Specifies the congestion control mode for the traffic acceleration. Enabled for the wan-accel mode only.
show scenario	name <scenario_name>	Displays information on scenarios (parameters related to the TCP congestion control function) of the specified scenario names.

A command execution example is shown below:

Execution example: Add high-speed scenario in the congestion control mode.

Parameter: Congestion control mode fast

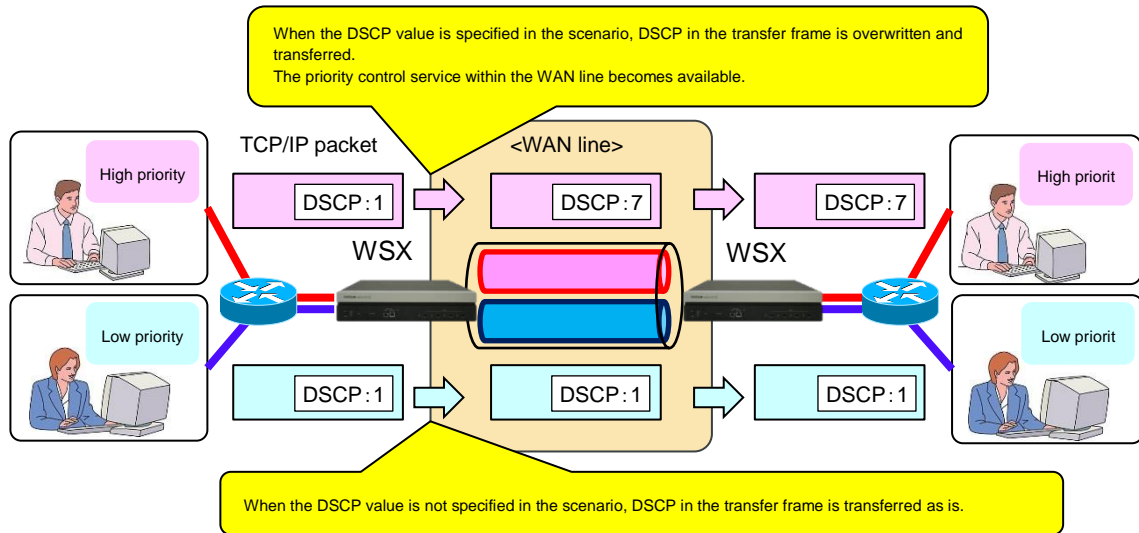
```
PureFlow(A)> add scenario /port1/woc1-fast action wan-accel peer 192.168.100.11
cc-mode fast peak 7G
```

The measures of setting the congestion control function are listed below: Perform fine adjustment depending on the line environment for use.

Example of line environment	Measure of setting
There is no packet discarding in the WAN line.	Set the congestion control mode to "normal".
A small amount of packet discarding occurs in the WAN line.	Set the congestion control mode to "semi-fast".
A large amount of packet discarding occurs in the WAN line.	Set the congestion control mode to "fast".

8.14.8 Remarking function

This equipment has the remarking function that overwrites "User Priority" (user priority: CoS) in the IEEE802.1Q and QinQ VLAN Tag fields and "DiffServ Code Point" (DSCP) in the "Type Of Service" (ToS) field in the IP header with the values specified in the scenario. When CoS and DSCP are overwritten in this equipment, the priority control service within the WAN line becomes available.



The scenario modes for which CoS and DSCP can be overwritten in this equipment are shown below.

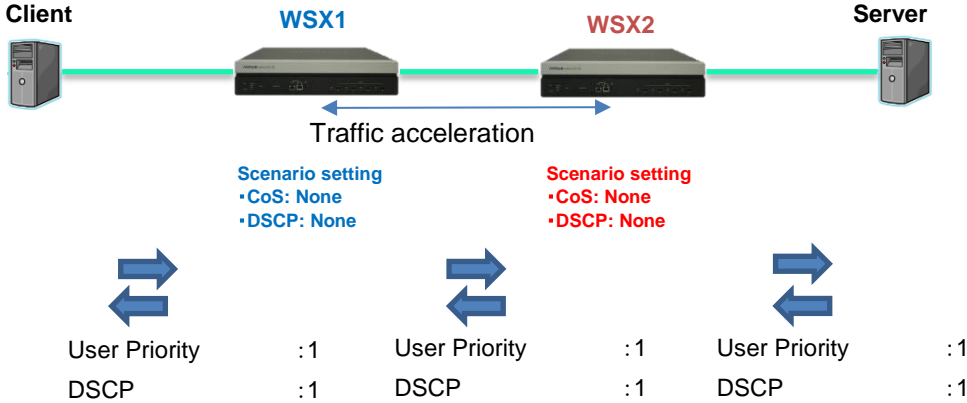
Scenario mode	CoS overwrite	DSCP overwrite
Acceleration mode (Wan-accel mode)	OK	OK
Aggregate queue mode (Aggregate mode)	OK	OK
Individual queue mode (Individual mode)	OK	OK
Discard mode	N/A	N/A

Note:

In the acceleration mode, configure the local device, opposing device, and acceleration tunnel to perform traffic acceleration. For overwriting CoS and DSCP in the acceleration mode, make the scenario settings of the local device and opposing device the same values. In TCP communication between the client and server, CoS and DSCP of the TCP packet sent to the WAN side and LAN side become the same values.

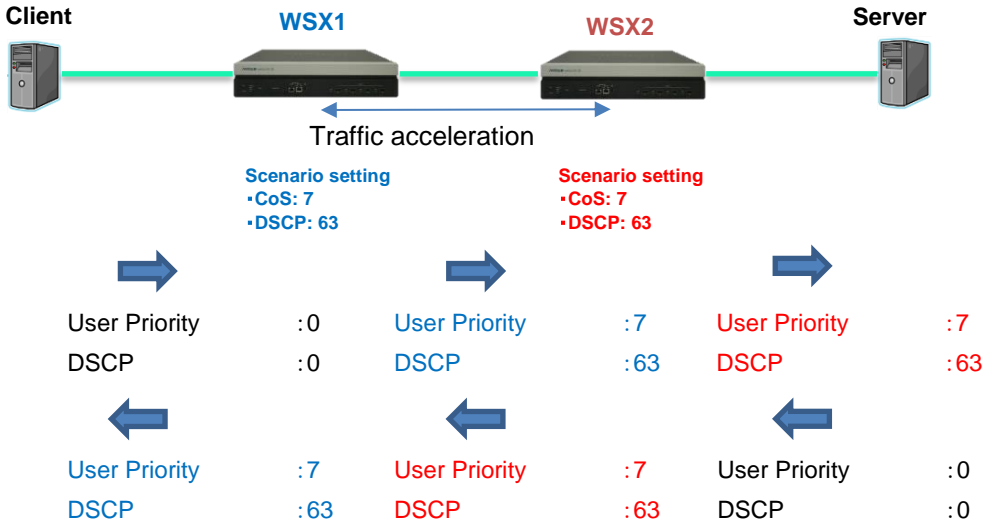
[Case 1] Not setting CoS and DSCP in the acceleration mode scenario

CoS and DSCP of the TCP packet between the client and server are not overwritten.



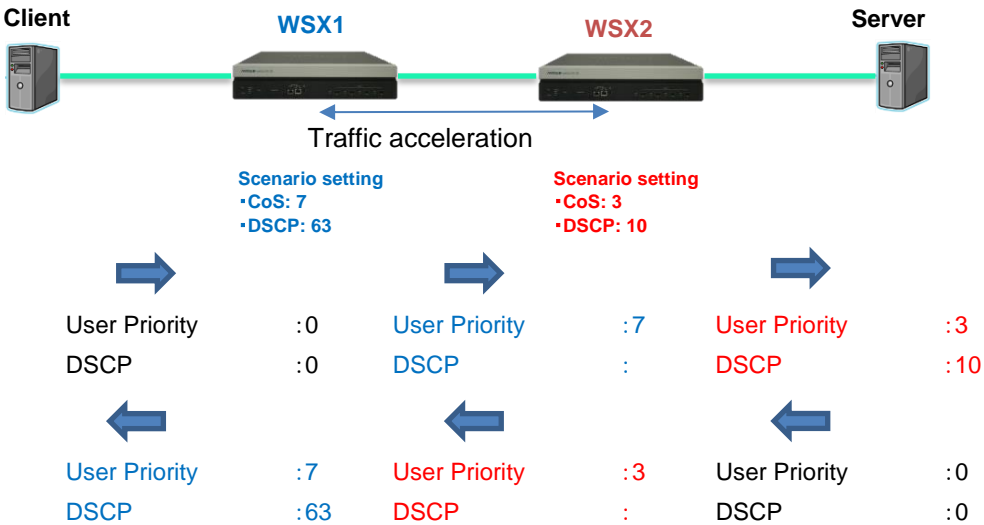
[Case 2] Setting the same CoS and DSCP in the local device and opposing device

CoS and DSCP of the TCP packet between the client and server are overwritten.



[Case 3] Setting different CoS and DSCP in the local device and opposing device

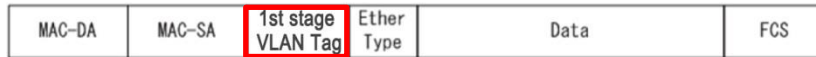
CoS and DSCP of the TCP packet between the client and server are overwritten.



This function can overwrite user priority (CoS) which is the upper 3 bits in the VLAN Tag of the Ethernet frame transferred by this equipment. It can also overwrite DSCP which is the upper 6 bits of the ToS field in the IP header.

The frame format is shown below.

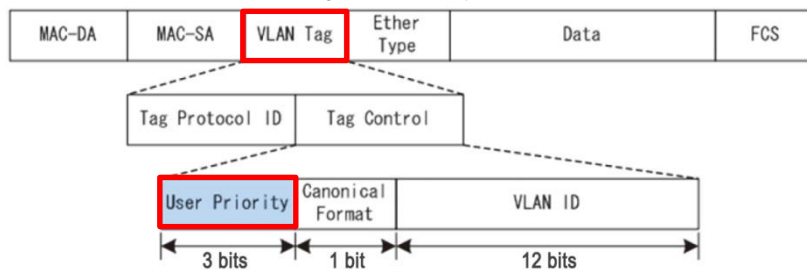
Ethernet frame format of the VLAN Tag



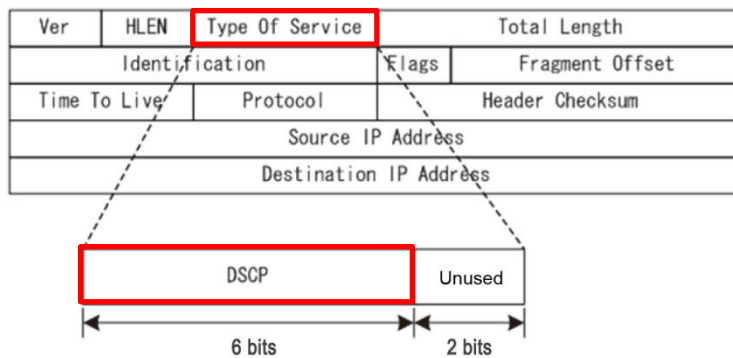
Ethernet frame format of double the VLAN Tag



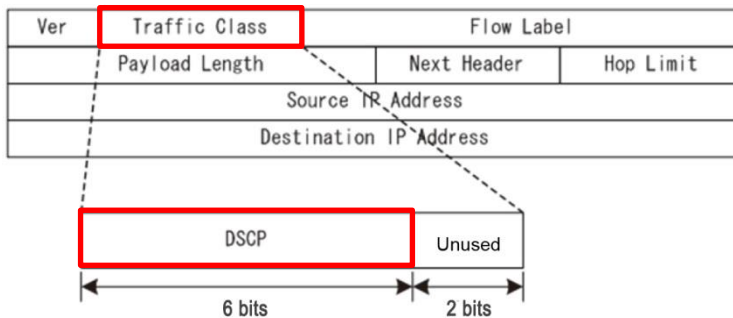
Header format of the VLAN Tag (user priority: CoS)



Header format of IPv4 (DSCP)



Header format of IPv6 (DSCP)



This function is set for each scenario. To use this function, set the parameters when registering or updating the scenario. Scenario update can be performed only in the aggregate mode and individual mode.

The parameters related to the remarking function are shown below. For details of how to set and check the parameters, see "STEP 4 Set the scenario." in "8.9 Setting Procedure".

Command	Parameter	Description
add scenario update scenario	cos {through <user_priority>}	Specifies the CoS overwrite value of a frame with the VLAN Tag.
	inner-cos {through <user_priority>}	Specifies the CoS overwrite value of a frame with the double VLAN Tag.
	dscp {through <dscp>}	Specifies the DSCP overwrite value.
show scenario	name <scenario_name>	Displays the scenario information (parameters related to the remarking function) of the specified scenario name.

A command execution example is shown below:

Execution example <1>: Specifying DSCP in the scenario (CoS not specified)

Parameter: DSCP 5

```
PureFlow(A)> add scenario /port1/woc1 action wan-accel peer 192.168.100.11 dscp 5
PureFlow(A)> add scenario /port1/agg1 action aggregate dscp 5
```

Execution example <2>: Updating CoS and Inner-CoS in a scenario other than the acceleration mode that has been registered

Parameter: CoS 3, Inner-CoS 4

```
PureFlow(A)> update scenario /port1/agg1 action aggregate cos 3 inner-cos 4
```

Execution example <3>: Updating a scenario other than the acceleration mode that has been registered without overwriting CoS and DSCP

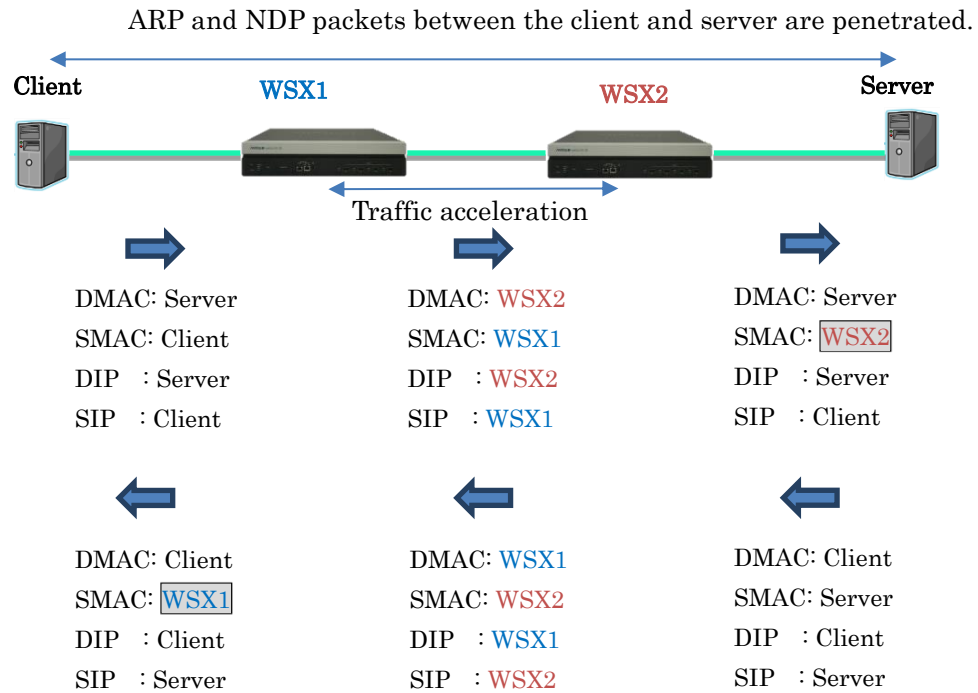
Parameter: CoS, DSCP overwrite disabled

```
PureFlow(A)> update scenario /port1/agg1 action wan-accel cos through dscp through
```

8.15 Address during the traffic acceleration

The IP address and MAC address specified in the channel interface of this device enable the traffic acceleration. The relationship between the IP address and MAC address during the traffic acceleration is shown below:

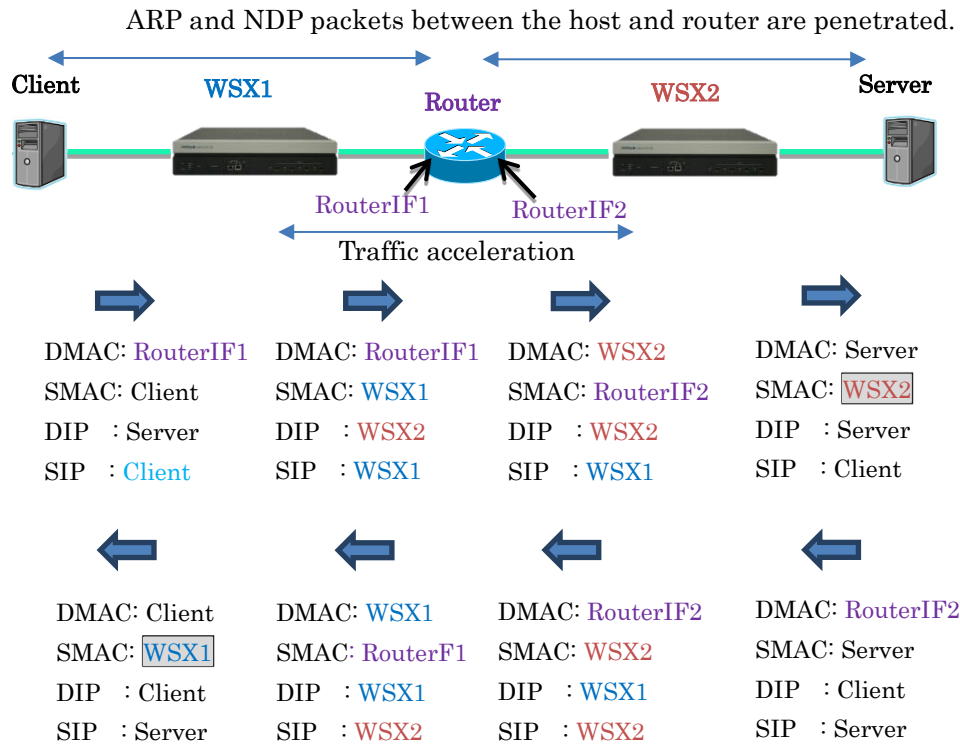
[Case 1] Traffic acceleration in the same subnetwork



ARP and NDP are transmitted and received between the Client and Server, and WSX penetrates them.

The transmission source MAC address of the TCP session that applied the traffic acceleration is replaced with the MAC address of WSX. The MAC address used in this case is the “Forwarding MAC Address” displayed in the “show module” command.

[Case 2] Traffic acceleration performed via the router



ARP and NDP are transmitted and received between the host and Router, and WSX penetrates them.

The transmission source MAC address of the TCP session that applied the traffic acceleration is replaced with the MAC address of WSX. The MAC address used in this case is the “Forwarding MAC Address” displayed in the “show module” command.

(Blank page)

Chapter 9 *Link-down Transfer*

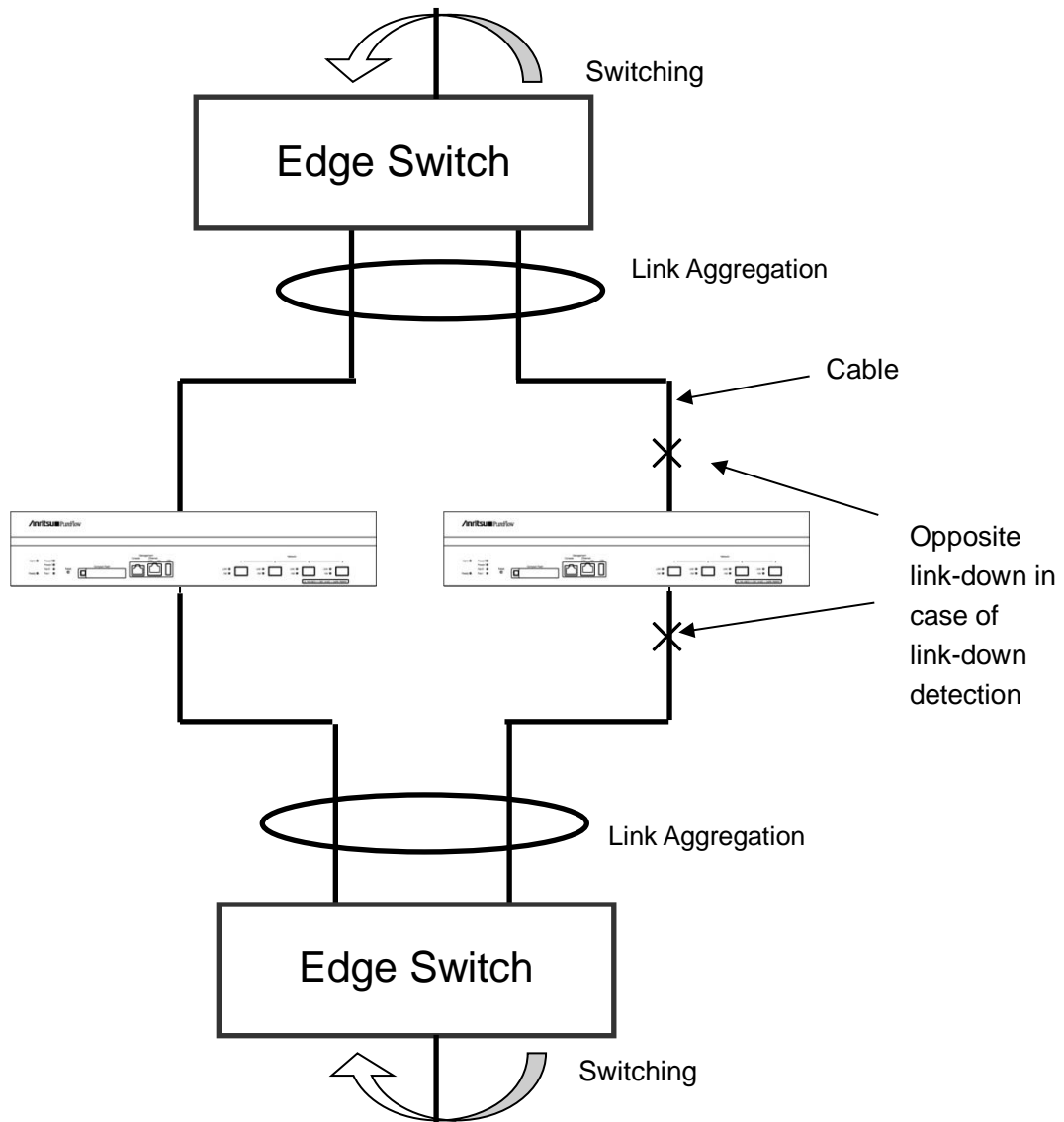
This chapter describes the link down transfer feature.

9.1	Link-down Transfer	9-2
-----	--------------------------	-----

9.1 Link-down Transfer

The link down transfer feature of this device allows coordinated operation without disturbing the line redundancy between the external devices even when the device is inserted between devices using a line redundancy feature such as “IEEE802.3ad Link Aggregation”.

When this device detects a link-down, it transfers an alarm to the communicating device by bringing down the communicating link. The communicating device can switch the line by detecting the link-down.



The link down transfer settings are as follows:

add lpt pair port <slot/port> <slot/port>	Registers the Network port combination of link down transfer function.
delete lpt pair port <slot/port> <slot/port>	Deletes the Network port combination of link down transfer function.
set lpt {enable disable}	Sets enabling/disabling the link-down transfer function.
show lpt	Displays the information related to link down transfer.

The following is a command execution example:

```
PureFlow(A)> add lpt pair port 1/1 1/2
PureFlow(A)> add lpt pair port 1/3 1/4
PureFlow(A)> set lpt enable
PureFlow(A)>
```

(Note 1)

To register or delete the combination of the Network ports, execute the command when the link-down transfer function is disabled.

(Note 2)

A Network port that is registered once cannot be registered in another combination.

(Blank page)

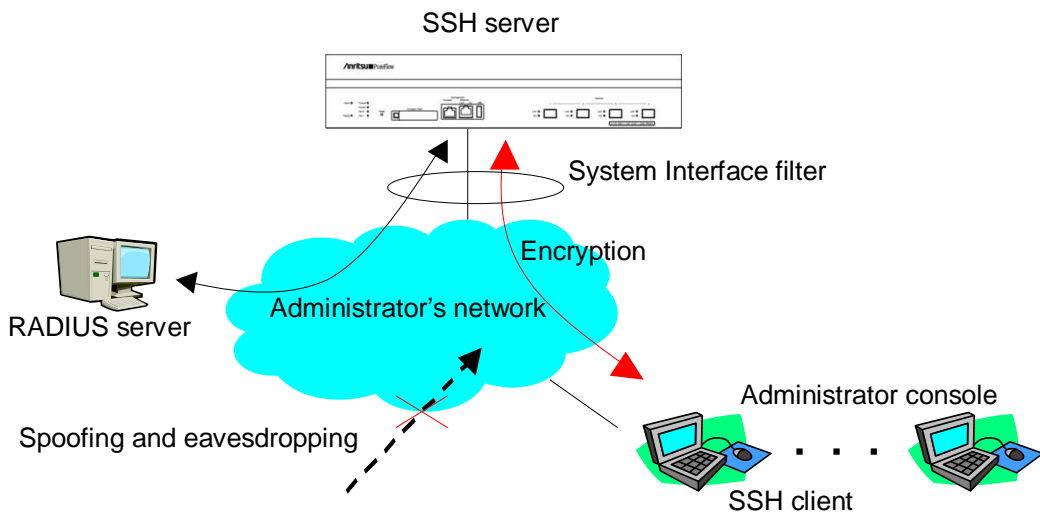
This chapter describes the SSH (Secure Shell) feature.

10.1	Overview.....	10-2
10.2	Specifications.....	10-3
10.3	Using SSH.....	10-4
	10.3.1 Device setting	10-4
	10.3.2 Preparing the SSH client	10-4
	10.3.3 Cautions	10-5

10.1 Overview

This device provides a SSH server feature that complies with the SSH versions 2. The SSH server feature encrypts communication between this device and SSH clients, enabling secure remote operation even via a network where safety is not guaranteed. It also has a powerful server authentication feature to prevent eavesdropping and spoofing by a third party.

When using connection with the SSH server, you can set system interface filters to restrict communication from an indefinite number of terminals to this device. For details, see Chapter 7 “System Interface Settings”. Also, as in Telnet, you can use password authentication of root users set to the local terminal as well as password authentication via the RADIUS server. For details about the RADIUS feature, see Chapter 13 “RADIUS”.



10.2 Specifications

The specifications of the SSH server feature of this device are shown below.

Item	Contents
SSH version	Compliant with SSH Ver 2
User authentication method	Password authentication
Key-exchange algorithm	ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256, diffie-hellman-group-exchange-sha1, diffie-hellman-group14-sha1, diffie-hellman-group1-sha1
Public key algorithm	RSA 2048bit, DSA 1024bit, ECDSA 256bit
Encryption algorithm	aes128-ctr, aes192-ctr, aes256-ctr, arcfour256, arcfour128, aes128-gcm@openssh.com, aes256-gcm@openssh.com, aes128-cbc, blowfish-cbc, cast128-cbc, aes192-cbc, aes256-cbc, arcfour, rijndael-cbc@lysator.liu.se
MAC algorithm	hmac-md5-etm@openssh.com, hmac-sha1-etm@openssh.com, umac-64-etm@openssh.com, umac-128-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-ripemd160-etm@openssh.com, hmac-sha1-96-etm@openssh.com, hmac-md5-96-etm@openssh.com, hmac-md5, hmac-sha1, umac-64@openssh.com, umac-128@openssh.com, hmac-sha2-256, hmac-sha2-512, hmac-ripemd160, hmac-ripemd160@openssh.com, hmac-sha1-96, hmac-md5-96
Connection port number	22
Maximum number of client connections	4 (with Telnet connections)

10.3 Using SSH

10.3.1 Device setting

To use the SSH server function of this device, the following settings are required:

(1) System interface setting

Set the IP address and Gateway of this device. To restrict connected terminals, set up system interface filters. For details, see Chapter 7 “System Interface Settings”.

(2) Public key (host key) generation

The SSH server requires a host key (RSA authentication key or DSA authentication key) to establish connection with a SSH client. A randomly generated host key is set at factory shipment. This host key is saved in the device so that it cannot be referenced from outside of the device. You do not necessarily need to generate a new host key, but you can change it from the serial console as required.

10.3.2 Preparing the SSH client

Prepare an SSH client in compliance with SSH version 2.

10.3.3 Cautions

- (1) Cautions on using SSH connection for the first time
When connecting to a remote host from the SSH client for the first time, server authentication is performed to check if the host can be trusted. The SSH client displays the fingerprint of the authentication key reported by the remote host, and asks for confirmation on whether to connect to the host. In this case, it is recommended to check if the fingerprint of the remote host displayed by the SSH client and the fingerprint of this device match. The fingerprint of the host key of this device can be displayed by using the “show ssh” command.
- (2) Host key generation
The host key used by the SSH server of this device is factory-generated and saved in this device. You can change the host key by using the “set ssh server key” command. However, you can execute this command only when you log in from the serial console.
- (3) SSH connection after regenerating a host key
The SSH client stores the fingerprint of a remote host connected in the past. If the fingerprint reported in the past is different, the SSH client displays a warning and disconnects the SSH connection to the remote host. This operation prevents spoofing of the remote host, and many SSH clients perform a similar operation.
When a host key of this device is regenerated, you need to delete or update the fingerprint of this device from the SSH client from which you connected to this device via SSH. For details, see the manual of the SSH client.
- (4) SSH connection when the RADIUS feature is enabled
When the RADIUS feature of this device is enabled, this device makes an inquiry to the RADIUS server at login authentication. When a new SSH session connection is attempted from the SSH client to this device, the communication between the SSH client and this device is encrypted by the SSH feature, but the communication between the RADIUS server and this device is not encrypted. If communication with the RADIUS server is intercepted, the password is hidden by the RADIUS protocol, but the login name may be deciphered by a third party.

(Blank page)

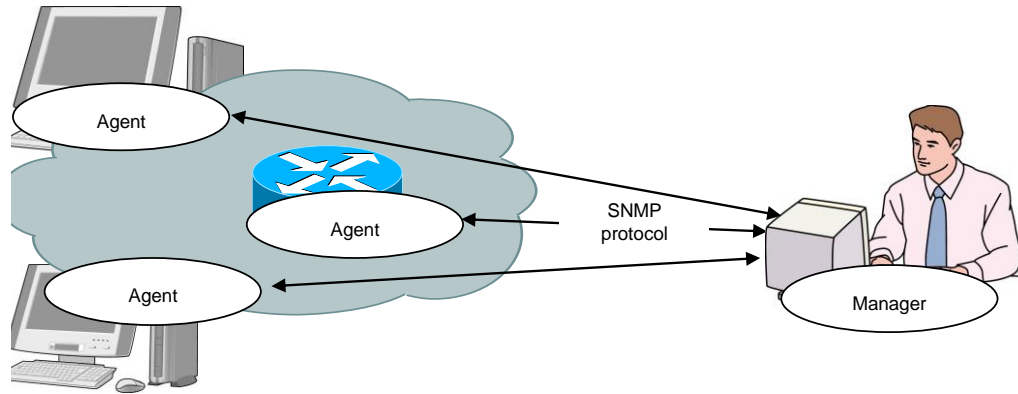
Chapter 11 *SNMP Setting*

This chapter describes the SNMP feature and settings.

11.1	Overview of SNMP.....	11-2
11.2	SNMPv1/SNMPv2c Setting	11-3
11.3	SNMPv3 Setting	11-5
11.4	TRAP Setting.....	11-7

11.1 Overview of SNMP

SNMP is a protocol to remotely manage network devices such as routers and servers over the network. For SNMP, managed routers and servers are called “agent nodes” (or agents), and PCs and EWS on which the management application software is installed are called “management nodes” (or managers). A network administrator uses the management node console for daily network management operations such as detecting errors of network devices (network nodes) and modifying settings.



There are three versions of SNMP: SNMPv1, SNMPv2c, and SNMPv3.

This device supports all three versions, SNMPv1, SNMPv2c, and SNMPv3. Differences between these versions are as follows:

- **SNMPv1:** The simplest protocol consisting of three operations: retrieving, setting, and trapping (for warning) management information. Security is realized by a string called the community name (similar to a password). The community name is included in a packet along with an SNMPv1 data request, and therefore can be monitored and leaked by a network tester or other equipment. The community name is not encrypted, and is not considered safe. It can only be used for an intranet to which external users cannot connect.
- **SNMPv2c:** This protocol supports simultaneous data acquisition called bulk transfer to retrieve management data, which reduces the protocol overhead. Access security is realized via the community name string like SNMPv1, and therefore the security strength is the same as SNMPv1.
- **SNMPv3:** This protocol is latest, and authenticates access by using a user name and an encrypted password. A user name is needed to access the agent. User names are categorized into groups. The scope of management information acquisition and configuration permissions can be set per group. For example, you can set the administrator group, team administrator group, and general user group per corporate group so that permissions are in a hierarchy. This protocol is designed for general applications from a large-sized intranet to the Internet. SNMPv3 security supports encryption but this device does not support encryption.

General management software automatically detects the versions the agent can support, and uses the latest one on a priority basis.

- * If you use OpenFlow function, specify more than two seconds for the time-out value of SNMP manager.

11.2 SNMPv1/SNMPv2c Setting

For both SNMPv1 and SNMPv2c, a character string called a community name (similar to a password) is set to enable access from management nodes.

add snmp community <community_string> [version {v1 v2c}] [view <view_name>] [permission {ro rw}]	Adds an SNMPv1/v2c community.
delete snmp community <community_string>	Deletes a community.
add snmp view <view_name> <oid> {included excluded}	Sets the SNMP View (restriction of management scope). Note: Although the snmpv2 group can be specified by using this command, access via SNMP is not possible.
delete snmp view <view_name> [<oid>]	Deletes the SNMP View (restriction of management scope).
show snmp community [<community_string>]	Shows the set community.
show snmp view [<view_name>]	Shows the set View.

First, set the SNMPv1 community to “netman1”, and SNMPv2c community to “netman2”.

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp community netman1 version v1 permission rw
PureFlow(A)> add snmp community netman2 version v2c permission rw
```

View is a mechanism that determines which MIB Tree of this device can be accessed by the management node accessed via the community name. If View is omitted in the “add snmp community” command, access is permitted for the View name “All”. If you use v2c trap transmission, add the “included” setting for “system” and “snmpmodules” if you specify “private” for the <oid> parameter.

To restrict access to SNMPv1 community netman1 from the interfaces group, run the following commands:

```
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp view myview1 interface included
PureFlow(A)> add snmp community netman1 version v1 view myview1 permission rw
```

To confirm the community name set by the setting command and the content of the View, run the “show snmp community” command and “show snmp view” command.

```
PureFlow> show snmp community
```

```
-----  
Community Name      :netman1  
Version             :v1  
Read View           :myview1  
Write View          :myview1  
-----
```

```
Community Name      :netman2  
Version             :v2c  
Read View           :All  
Write View          :All  
-----
```

```
PureFlow>
```

```
PureFlow> show snmp view
```

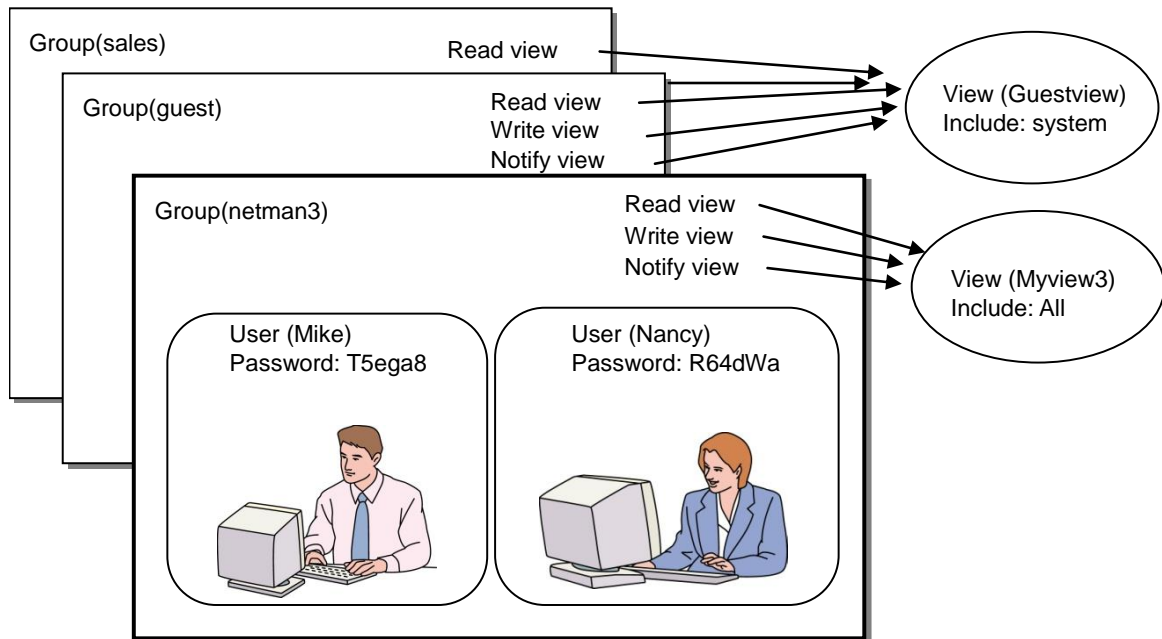
```
-----  
View name           :All  
Subtree             :iso  
Access State        :included  
-----
```

```
View name           :myview1  
Subtree             :interface  
Access State        :Included  
-----
```

```
PureFlow>
```

11.3 SNMPv3 Setting

The SNMPv3 management framework is user-based security in which security is set per user. Each user belongs to a group, and View is set as a group attribute.



To use SNMPv3, the group, user, and View must be set. Run the following commands:

<code>add snmp group <group_name> [auth_type {auth noauth}] [read <readview>] [write <writeview>] [notify <notifyview>]</code>	Adds an SNMPv3 group.
<code>delete snmp group <group_name></code>	Deletes a group.
<code>add snmp user <user_name> <group_name> [auth_type {auth noauth}] [password <auth_password>]</code>	Adds an SNMPv3 user. To set the password, ensure the length is from 8 to 24 characters.
<code>delete snmp user <user_name></code>	Deletes a user.
<code>add snmp view <view_name> <oid> {included excluded}</code>	Sets the SNMP View (restriction of management scope). Note: Although the snmpv2 group can be specified by using this command, access via SNMP is not possible.
<code>delete snmp view <view_name> [<oid>]</code>	Deletes the SNMP View (restriction of management scope).
<code>show snmp group [<group_name>]</code>	Shows the set group.
<code>show snmp user [<user_name>]</code>	Shows the set user.
<code>show snmp view [<view_name>]</code>	Shows the set View.

View is a mechanism that determines which MIB Tree of this device can be accessed by the management node accessed via the group and user names. If View is omitted in the “add snmp group” command, access is permitted for the View name “All”. If you use v3 trap transmission, add the “included” setting for “system” and “snmpmodules” if you specify “private” for the <oid> parameter.

The following commands set the SNMPv3 users Mike and Nancy as members of netman3 group.

```
PureFlow(A)> add snmp view myview3 iso included
PureFlow(A)> add snmp group netman3 auth_type auth read myview3 write myview3
notify myview3
PureFlow(A)> add snmp user Mike netman3 auth_type auth password T5ega8GH
PureFlow(A)> add snmp user Nancy netman3 auth_type auth password R64dWa99
```


11.4 TRAP Setting

SNMP has a feature to notify a management node of a detected status change of the agent node. Specify the View for notification and the management node (host) address so that the TRAP (notification) can be send to the management node.

add snmp view <view_name> <oid> {included excluded}	Sets the SNMP View (restriction of management scope).
add snmp host <host_address> version {v1 v2c v3 [auth_type { auth noauth}] } {user community} <community_string / username> }{trap inform} [udp_port <port_number>] [<notification_type>]	Adds the host indicating the SNMP TRAP (notification) destination.
delete snmp host <host_address>	Deletes the host indicating the TRAP destination.
set snmp traps {authentication linkup linkdown warmstart coldstart modulefailurealarm modulefailurerecovery systemheatalarm systemheatrecovery powerinsert powerextract powerfailure powerrecovery faninsert fanextract fanfailure fanrecovery queuebuffalarm queuebuffrecovery systembuffalarm systembuffrecovery queueallocalarm queueallocalrecovery maxqnumalarm maxqnumrecovery} {enable disable}	Enables/disables SNMP TRAP transmission. This can be set per trap type. For <trapname>, “authentication”, “linkup”, “linkdown”, “coldstart”, “modulefailurealarm”, “modulefailurerecovery”, “systemheatalarm”, “systemheatrecovery”, “powerinsert”, “powerextract”, “powerfailure”, “powerrecovery”, “faninsert”, “fanextract”, “fanfailure”, “fanrecovery”, “queuebuffalarm”, “queuebuffrecovery”, “systembuffalarm”, “systembuffrecovery”, “queueallocalarm”, “queueallocalrecovery”, “maxqnumalarm”, “maxqnumrecovery” can be specified.
show snmp host [<host_address>]	Displays the list of hosts indicating TRAP destinations.

First set the View for SNMP TRAP transmission. SNMP basic TRAP is included in the snmpv2 object, and Enterprise TRAP is included in the private object. Enable access to the snmpv2 object and private object so that TRAP can be sent to the management node.

```
PureFlow(A)> add snmp view All iso included  
PureFlow(A)> add snmp host 192.168.1.10 version v1 community public trap udp_port 162
```

To disable authenticationFailure TRAP transmission, configure as follows:

```
PureFlow(A)> set snmp traps authentication disable
```

To check the content of the host set by the setting command, use the “show snmp system” command.

```
PureFlow(A)> show snmp host  
-----  
Host Address      :192.168.1.10  
Version           :v1  
Security          :No Authentication  
Security Name     :public  
UDP port         :162  
Notification Type :all  
-----  
Host Address      :192.168.1.11  
Version           :v2c  
Security          :No Authentication  
Security Name     :public  
UDP port         :162  
Notification Type :all  
PureFlow(A)>
```

To check the status (enabled/disabled) set by the setting command, use the “show snmp system” command.

```
PureFlow(A)> show snmp system
```

```
-----
System Location           :Not Yet Set
System Contact            :Not Yet Set
System Name               :Not Yet Set
Engine ID                 :00:00:04:7f:00:00:00:a1:c0:a8:01:01
```

Traps

```
authentication           :disable
linkup                   :enable
linkdown                 :enable
warmstart                :enable
coldstart                :enable
modulefailurealarm      :enable
modulefailurerecovery   :enable
systemheatalarm         :enable
systemheatrecovery      :enable
powerinsert              :enable
powerextract             :enable
powerfailure             :enable
powerrecovery            :enable
faninsert                :enable
fanextract               :enable
fanfailure               :enable
fanrecovery              :enable
queuebuffalarm          :enable
queuebuffrecovery       :enable
systembuffalarm         :enable
systembuffrecovery      :enable
queueallocalarm         :enable
queueallocrecovery      :enable
maxqnumalarm            :enable
maxqnumrecovery         :enable
```

```
-----
PureFlow(A)>
```

(Blank page)

Chapter 12 Statistics

This chapter describes the statistics.

This device provides statistics on ports and scenarios.

12.1	Port Statistics.....	12-2
12.1.1	Port counter.....	12-2
12.2	Scenario Statistics	12-3
12.2.1	Scenario counter	12-3
12.2.2	Scenario operation information	12-4
12.2.3	Rate measurement.....	12-5
12.2.4	Determining the scenario parameters	12-6

12.1 Port Statistics

The port statistics contain the Network port counter and system interface counter.

This information is statistical information about the system interface for each Network port.

12.1.1 Port counter

This is the system interface counter per Network port.

The port counter displays the following:

- Number of received bytes
- Number of received packets
- Number of received broadcast packets
- Number of received multicast packets
- Number of transmitted bytes
- Number of transmitted packets
- Number of transmitted broadcast packets
- Number of transmitted multicast packets
- Number of reception error packets
- Number of packet collisions
- Number of discarded packets
- Average rate of received packets (kbit/s)
- Average rate of transmitted packets (kbit/s)

The system interface counter shows the following:

- Number of received bytes
- Number of received packets
- Number of transmitted bytes
- Number of transmitted packets

The following CLI commands can be used for the port counter:

<code>show counter [brief]</code>	Shows the counter for all Network ports and system interface. Specify “brief” to show an overview.
<code>show counter {<slot/port> system}</code>	Displays the counter of the specified Network port or system interface.
<code>clear counter [<slot/port> system]</code>	Clears the counter of the specified Network port or system interface.

12.2 Scenario Statistics

The scenario statistics contain the scenario counter, scenario operation information, and rate measurement.

This information is the statistics for each scenario.

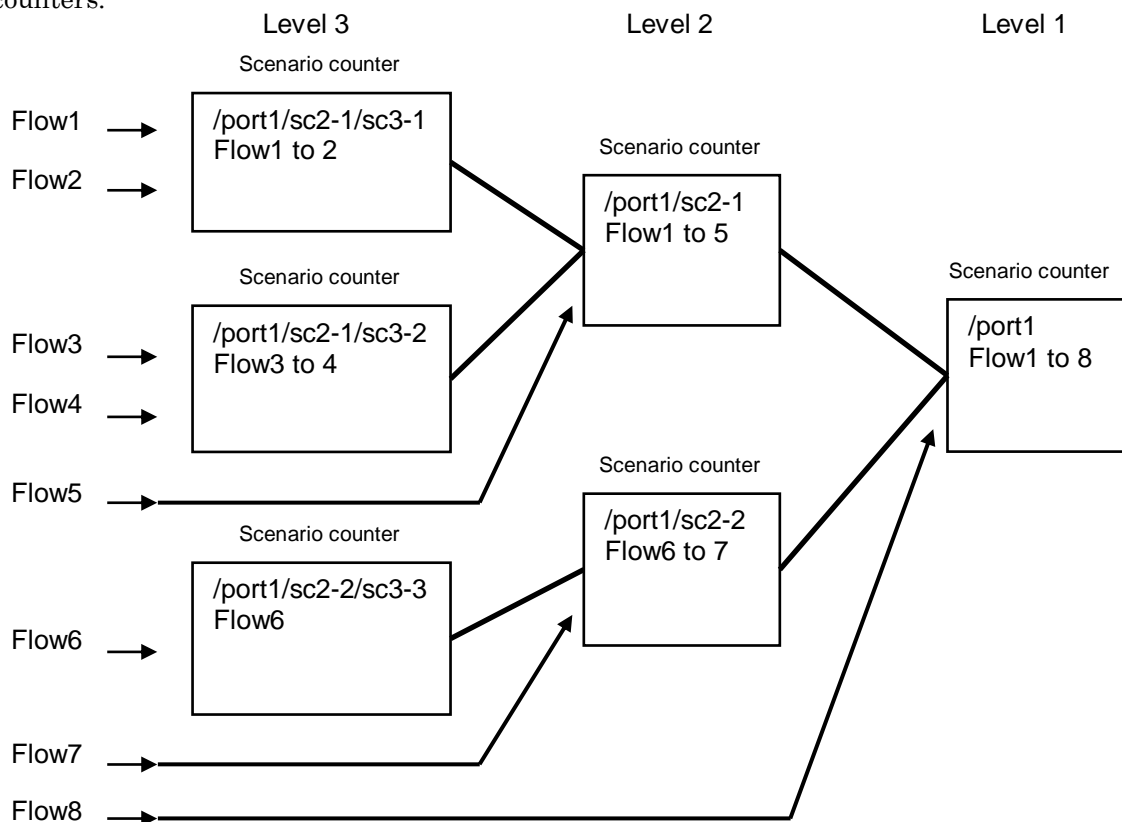
12.2.1 Scenario counter

This is the counter per scenario.

The scenario counter shows the following:

- Number of received bytes, number of received packets
- Number of transmitted bytes, number of transmitted packets
- Number of discarded bytes, number of discarded packets

The scenario counter shows the total number including the related lower level scenario counters.



The following CLI commands can be used for the scenario counter:

show scenario counter name <scenario_name>	Shows a scenario counter.
show scenario counter summary	Displays a list of scenario counters.
clear scenario counter name <scenario_name>	Deletes a scenario counter.
clear scenario counter all	Clears all scenario counters.

For <scenario_name>, specify the scenario set by the “add scenario” command.

12.2.2 Scenario operation information

This is the operation information per scenario.

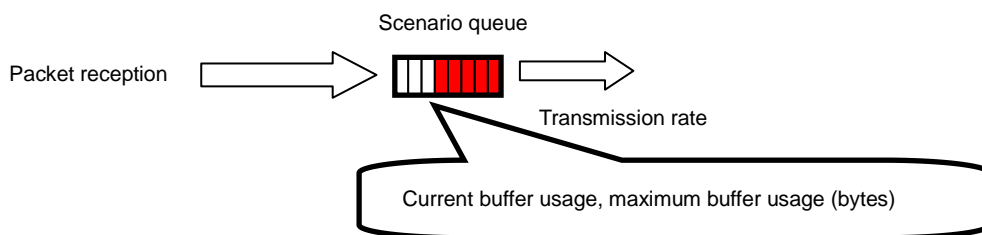
The scenario operation information shows the following:

<Information on the default queue of the scenario>

- Buffer usage and use rate
- Buffer peak hold (the maximum value of buffer usage)
- Number of flows

<Information on the scenario transmission rate>

- Peak transmission rate (the peak transmission rate over the last 1 minute)
- Average transmission rate (the average transmission rate over the last 1 minute)



<Information on the traffic acceleration>

- The TCP session count that apply the traffic acceleration
- Opposing device that configures the acceleration tunnel (PRIMARY/SECIBDARY)
- The setting contents and operation states related to the bypass function of the traffic acceleration (For details about the bypass function, see "8.13.4 Traffic acceleration bypass".)

The following CLI commands can be used for the scenario operation information:

show scenario info name <scenario_name>	Shows the operation information for the scenario.
show scenario info summary	Displays a list of operation information for the scenario.
clear scenario peakhold buffer name <scenario_name>	Clears the maximum buffer usage for the scenario.
clear scenario peakhold buffer all	Clears the maximum buffer usage of all scenarios.

For <scenario_name>, specify the scenario set by the “add scenario” command.

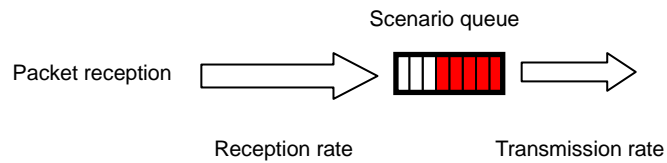
Note:

For the individual queue mode scenario, the buffer information of the individual queue is not displayed. It displays the buffer information of the queue to be forwarded if the queue upper limit is exceeded.

12.2.3 Rate measurement

Measure the transmission and reception rates of the scenario. The transmission and reception rates are measured every minute, and shown the specified number of times.

A value to the third decimal place is shown in kbit/s units. Measurement of transmission and reception rates only targets packets, and does not include gaps and preambles between frames.



The following CLI commands can be used for rate measurement:

<code>monitor rate <scenario_name> [<num>]</code>	Measures the transmission and reception rates of the scenario.
---------------------------------------------------------------	----------------------------------------------------------------

The following is a command execution example:

```
PureFlow(A)> monitor rate /port1/Tokyo 3
```

```
Scenario Name : "/port1/Tokyo"
```

Times[s]	Rcv Rate[kbps]	Trs Rate[kbps]
1	3587.562	1254.531
2	3482.826	1198.426
3	3624.692	1217.879
Average	3565.026	1223.612

```
PureFlow(A)>
```

Note:

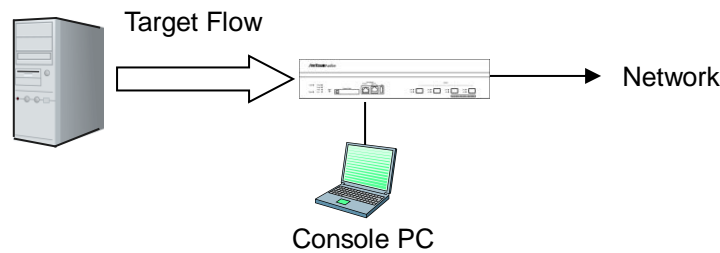
“bps” in CLI means bits per second.

12.2.4 Determining the scenario parameters

The scenario statistics provide the average rate of the scenario and the burst size for reference for determining parameters. This section describes how to determine the parameters.

STEP 1 Measuring the average rate using the rate measurement feature

To measure the rate, assign a scenario. Set the scenario and filter for the flow to be measured.



First set the measuring scenario to Level 2, and the buffer size to 100 MB (the maximum valid value). For the scenario to be measured, set a filter that matches the target flow.

Setting example:

```
PureFlow(A)> add scenario /port1/meassscenario action aggregate bufsize 100M
PureFlow(A)> add filter scenario /port1/meassscenario filter measflow ipv4 sip 192.168.10.9
```

Start the flow, and measure the rate for the target scenario.

```
PureFlow(A)> monitor rate /port1/meassscenario 3
Scenario Name : "/port1/meassscenario"
```

Times[s]	Rcv Rate[kbps]	Trs Rate[kbps]
1	3587.562	3587.562
2	3482.826	3482.826
3	3624.692	3624.692
<hr style="border-top: 1px dashed black;"/>		
Average	3565.026	3565.026

```
PureFlow(A)>
```

Note:

“bps” in CLI means bits per second.

The rate measurement result indicates that the average reception rate is approx. 3.6 Mbit/s.

STEP 2 Using buffer peak hold to measure the maximum buffer usage

Measure the burst size to determine the buffer size. Add a 10% margin to the average reception rate retrieved in STEP 1, and reset the traffic attribute to this value.

In the example below, the traffic attribute is reset to the rate of 4 Mbit/s.

```
PureFlow(A)> update scenario /port1/measscenario action aggregate peak_bw 4M
```

Then start the flow, and clear the maximum buffer usage.

```
PureFlow(A)> clear scenario peakhold buffer name /port1/measscenario
```

The maximum buffer usage is recorded again in this state. For normal video traffic, it takes approx. 1 minute to record the burst size of the video as the maximum buffer usage.

The recorded maximum buffer usage as follows:

```
PureFlow(A)> show scenario info name /port1/measscenario
Scenario 1: "/port1/measscenario"
  Rate Control Unit:
    Create Mode      :Aggregate
    Class            :2
    Min Bandwidth    :-----
    Peak Bandwidth   :4M[bps]
  Default Queue:
    Class            :8
    Buf Size         :100M[Bytes]

  *Attached Filters:
    "measflow"

  Scenario Rate Information
    Recent interval Tx peak      :0[bps]
    Recent interval Tx average   :0[bps]

  Default Queue Information
  Buffer Utilization
    Current              :105384(10%)[Bytes(%)
    Peak Hold           :149504(14%)[Bytes(%)
  Related Flow
    Flow Num            :1[flows]
PureFlow(A)>
```

The result indicates that the maximum buffer usage is 149504 bytes. Add a safety factor of 2 to the measured maximum buffer usage so that the buffer size is 300000 bytes.

```
PureFlow(A)> update scenario /port1/measscenario action aggregate
    bufsize 300000
```

This sets the traffic attributes of the target flow to the following values:

PeakBandwidth: 4 Mbit/s

BufSize: 300000 bytes

Note:

Use an appropriate safety factor according to the network environment and traffic.

(Blank page)

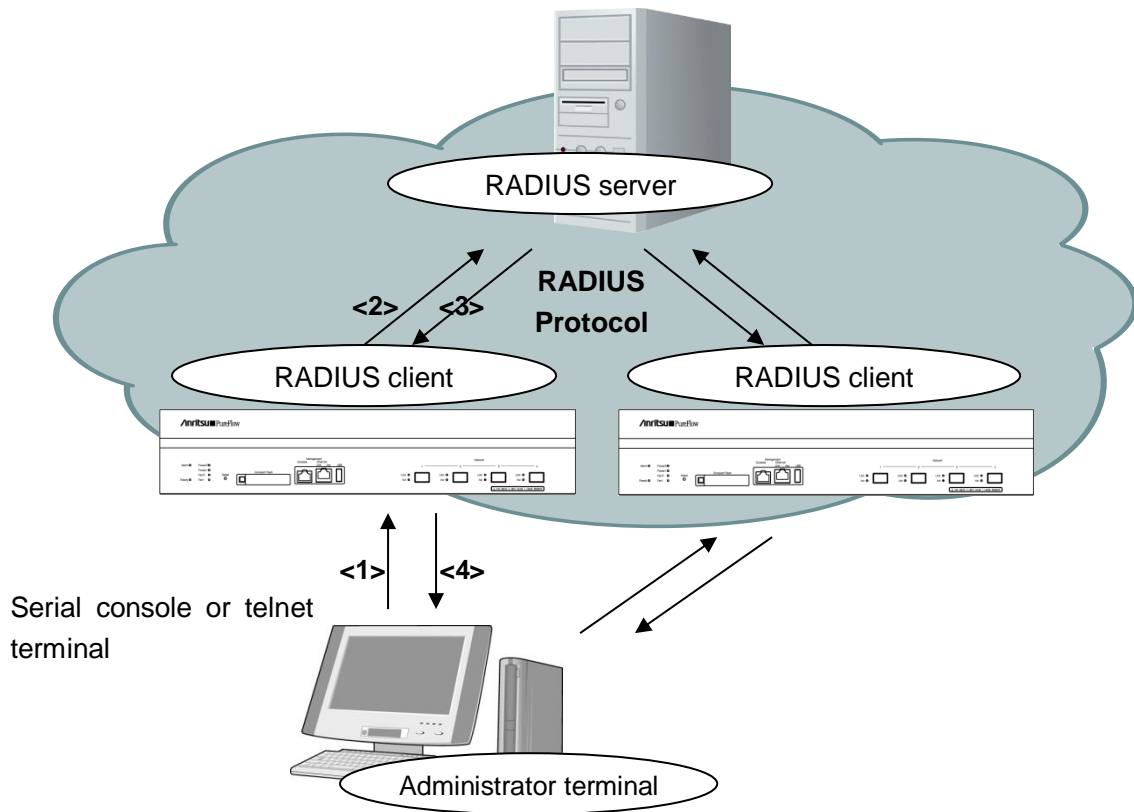
Chapter 13 RADIUS

This chapter describes the RADIUS (Remote Authentication Dial In User Service) feature.

13.1	Overview	13-2
13.2	Controlling Login Authentication	13-3
13.3	Controlling Login Mode	13-3
13.4	Setting Up the RADIUS Feature	13-4
13.5	RADIUS Server Settings	13-6

13.1 Overview

The RADIUS feature performs user authentication by using RADIUS (RFC2865) when a user logs into Telnet, SSH, or the serial console. This device operates as a RADIUS client to provide user authentication based on user information in the external RADIUS server.



- <1> The user enters their user name and password on the administrator terminal.
- <2> An authentication request is sent from the RADIUS client of this device to the RADIUS server.
- <3> An authentication response is sent from the RADIUS server to the RADIUS client.
- <4> This device permits connection from the administrator terminal based on the received authentication response.

13.2 Controlling Login Authentication

This section describes how to control login authentication with the RADIUS feature enabled. Control of login authentication when the RADIUS feature is enabled and disabled is shown below.

Login authentication procedure when RADIUS authentication is enabled	Login authentication procedure when RADIUS authentication is disabled
(1) Login authentication is performed based on the user name and login password set in the device. (2) If the login authentication is rejected, login authentication is performed based on the user name and login password set in the RADIUS server.	(1) Login authentication is performed based on the user name and login password set in the device.

13.3 Controlling Login Mode

This device switches the login mode when the user logs in according to the service type of the user set in the RADIUS server. The service types supported by this device are as follows:

Service type	Login mode
Login-User(1)	Normal mode
Administrative-User(6)	Administrator mode

If a service type other than the above is specified from the RADIUS server, the normal mode is used for logging in.

13.4 Setting Up the RADIUS Feature

User authentication as a RADIUS client can be performed by specifying the information of the RADIUS authentication server and authentication parameters.

set radius auth { enable disable }	Enables or disables RADIUS authentication.
set radius auth timeout <timeout>	Specifies the reception timeout value for the RADIUS authentication response packet. The setting range is 1 to 30 [seconds]. The default value is 5 [seconds].
set radius auth retransmit <retry>	Specifies the retransmission count for the RADIUS authentication request packet. The setting range is 0 to 10 [times]. The default value is 3 [times].
set radius auth method {PAP CHAP default}	Specifies the RADIUS authentication method.
add radius auth server <IP_address> [port <port>] key <string > [Primary]	Adds RADIUS authentication servers.
update radius auth server <IP_address> [port <port>] [key <string>] [Primary]	Changes the settings of the existing RADIUS authentication server.
delete radius auth server <IP_address>	Deletes the settings of the RADIUS authentication server.
show radius	Displays the RADIUS setting information.

An example of set up the RADIUS feature is shown below.

1. Specify the RADIUS authentication method. In the example, the PAP authentication method is specified.

```
PureFlow(A)> set radius auth method PAP
```

2. Add RADIUS authentication servers. In the example, two servers are registered. One server is set with the server IP address 192.168.1.10 and the RADIUS shared key “testing123”. Another server is set with the server IP address 192.168.1.11 and the RADIUS shared key “testing789”. Primary specification is set to the RADIUS server to which login authentication is sent first. If no Primary specification exists, login authentication is sent in the order of registration of the RADIUS servers.

```
PureFlow(A)> add radius auth server 192.168.1.10 key testing123 Primary
```

```
PureFlow(A)> add radius auth server 192.168.1.11 key testing789
```

3. Enable the RADIUS feature.

```
PureFlow(A)> set radius auth enable
```


4. Check the settings.

```
PureFlow(A)> show radius
```

```
RADIUS Authentication : Enable
```

```
RADIUS method       : PAP
```

```
RADIUS server entries : 2
```

```
Retry retransmit     : 5
```

```
Retry timeout        : 3
```

```
Type Pri Server      Port  key
-----
auth  * 192.168.1.10  1812 "testing123"
auth   192.168.1.11  1812 "testing789"
PureFlow(A)>
```

13.5 RADIUS Server Settings

This section describes how to set up the RADIUS server. Set the following user information to the RADIUS server.

RADIUS shared key

Specify the same string as the RADIUS shared key set for this device.

User ID

Set the user ID.

Authentication method

Specify the same authentication method (CHAP or PAP) as the authentication method set for this device.

Password

Set the login password.

Service type

Specify this parameter as required. If the RADIUS server does not give notification of any service type, this device allows the user to log into the normal mode. If the RADIUS server gives notification of a service type and if it is Administrative-User, this device allows the user to log into the administrator mode.

This document assumes that FreeRADIUS version 1 is used as the RADIUS server. The actual setting may vary depending on the type and version of your RADIUS server. FreeRADIUS can be integrated with various types of user information such as LDAP (Lightweight Directory Access Protocol), SQL Server, and UNIX system user information, and it can be used for management, authentication, and authorization of many users within a corporation.

Note:

It is assumed that FreeRADIUS is installed in Linux. For details on how to set up and use FreeRADIUS, see the manual of the installed software.

FreeRADIUS version 1 setting

1. Setting the RADIUS shared key

Specify the IP address of the device to be registered as a RADIUS client and the RADIUS shared key in the following format in the RADIUS server.

Open the `clients.conf` file under `/usr/local/etc/raddb` in the RADIUS server, and add the following setting in the appropriate section:

```
client 192.168.37.10 {
    secret = testing123
    shortname = wsx
}
```

2. Setting a user

Specify the user information for allowing login to this device in the RADIUS server. Specify a user ID, authentication method, password, and service type for each user.

Open the `/usr/local/etc/raddb/users` file in the RADIUS server, and add the following setting in the appropriate section.

1) Using CHAP as the authentication method

Setting a user for which login in the normal mode is allowed

```
user1 Cleartext-Password:=" user1passwd "
    Auth-Type:=CHAP,
    Service-Type=Login-User
```

Setting a user for which login in the administrator mode is allowed

```
user2 Cleartext-Password:=" user2passwd "
    Auth-Type:=CHAP,
    Service-Type= Administrative-User
```

2) Using PAP as the authentication method

Setting a user for which login in the normal mode is allowed

```
user3 Cleartext-Password:=" user3passwd "
    Auth-Type:=PAP,
    Service-Type=Login-User
```

Setting a user for which login in the administrator mode is allowed

```
user4 Cleartext-Password:=" user4passwd "
    Auth-Type:=PAP,
    Service-Type=Administrative-User
```

(Blank page)

Chapter 14 Downloading and Uploading Data

This chapter describes how to download or upload software and configuration data.

14.1	Downloading/Uploading Software.....	14-2
14.1.1	Downloading software from a CF card.....	14-2
14.1.2	Uploading software to a CF card	14-3
14.1.3	Downloading software from a USB flash drive	14-3
14.1.4	Uploading software to a USB flash drive.....	14-3
14.1.5	Downloading software via TFTP.....	14-4
14.1.6	Downloading software via FTP	14-5
14.1.7	Downloading software via WebGUI	14-5
14.2	Downloading the Software Update Patch	14-6
14.2.1	Downloading software Update Patch from a CF card.....	14-6
14.2.2	Downloading software Update Patch from a USB flash drive	14-6
14.3	Downloading/Uploading Configuration Data	14-7
14.3.1	Downloading configuration data from a CF card.....	14-7
14.3.2	Uploading configuration data to the CF card.....	14-7
14.3.3	Downloading configuration data from a USB flash drive	14-8
14.3.4	Uploading configuration data to a USB flash drive	14-8
14.3.5	Downloading configuration data via TFTP ..	14-9
14.3.6	Uploading configuration data via TFTP	14-9
14.3.7	Downloading configuration data via FTP.....	14-10
14.3.8	Uploading configuration data via FTP	14-10
14.4	Restarting the Software.....	14-11

To download or upload software and configuration data, use a Compact Flash card (hereafter referred to as “CF card”) or USB flash drive. FAT16/FAT32 are supported as the file format. For downloading software and downloading/uploading configuration data, you can also use TFTP or FTP from the system interface. To use the system interface, provide your PC with TFTP or FTP server functionality.

When using a CF card, use the CF card recommended by Anritsu Networks. Operation with a CF card other than a recommended one is not guaranteed. For details of validated USB flash drives, see the Operation Manual.

For loading software and configuration data, the Command Line Interface (CLI) is used. For CLI, see Chapter 3 “Configuring Settings”.

14.1 Downloading/Uploading Software

Cautions on downloading software

If any object file other than the proper object file specified by Anritsu Networks (file name: nf7600.bin) is downloaded, the device may not start up. Be careful not to download a file other than the proper object file by using the download command above. If the wrong object file is downloaded, insert a CF card or USB flash drive containing the proper object file and start the device. After that, download the proper object file again. For more information on how to obtain the proper object file, contact the supplier.

14.1.1 Downloading software from a CF card

Insert a CF card with the new software object into the CF card slot to download the new software to this device. The downloaded software is automatically saved to the internal flash memory. The software of the old version is saved to a different area, and the new software is written. During version upgrade, be careful not to remove the CF card or turn off the power of the device. If the CF card is removed or the power of the device is turned off during the operation, the device will load the old software saved in a different area. In this case, restart the device and try the download operation again.

```
PureFlow(A)> download cf obj nf7600.bin
Download “nf7600.bin” from Flash Memory Card (y/n)? y
Loading .....
creating Backup from Master file.....completed.
Done.
PureFlow(A)>
```

The new software is not reflected immediately when download is complete. Restart the device after the download is complete.

14.1.2 Uploading software to a CF card

Insert a CF card into the CF card slot to upload the software to the CF card. The uploaded software is saved to the inserted CF card.

```
PureFlow(A)> upload cf obj nf7600.bin
Upload as "nf7600.bin" to Flash Memory Card (y/n)? y
Loading .....
Done.
PureFlow(A)>
```

14.1.3 Downloading software from a USB flash drive

Insert a USB flash drive with the new software object into the USB port to download the new software to this device. The downloaded software is automatically saved to the internal flash memory. The software of the old version is saved to a different area, and the new software is written. During version upgrade, be careful not to remove the USB flash drive or turn off the power of the device. If the USB flash drive is removed or the power of the device is turned off during the operation, the device will load the old software saved in a different area. In this case, restart the device and try the download operation again.

```
PureFlow(A)> download usb obj nf7600.bin
Download "nf7600.bin" from USB Memory (y/n)? y
Loading .....
creating Backup from Master file.....completed.
Done.
PureFlow(A)>
```

The new software is not reflected immediately when download is complete. Restart the device after the download is complete.

14.1.4 Uploading software to a USB flash drive

Insert a USB flash drive into the USB port to upload the software to the USB flash drive. The uploaded software is saved to the inserted USB flash drive

```
PureFlow(A)> upload usb obj nf7600.bin
Upload as "nf7600.bin" to USB Memory (y/n)? y
Loading .....
Done.
PureFlow(A)>
```

14.1.5 Downloading software via TFTP

The software can be downloaded to the device via TFTP. The downloaded software is automatically saved to the internal flash memory. The software of the old version is saved to a different area, and the new software is written. During version upgrade, be careful not to turn off the power of the device. If the power of the device is turned off during the operation, the device will load the old software saved in a different area. In this case, restart the device and try the download operation again. If the communication is discontinued during download, start the downloading work again.

To download the software to the device, use the following command. Specify the correct IP address for the system interface in advance to enable communication with the TFTP server. For more information on system interface settings, see Chapter 7 “System Interface Settings”.

Because the file size of the software is more than 32MByte, please use the TFTP server that supports the tsize option that are specified in RFC2349.

```
PureFlow(A)> download tftp obj 192.168.100.40 nf7600.bin
Download “nf7600.bin” from 192.168.100.40 (y/n)? y
Loading ...
creating Backup from Master file.....completed.
Done.
PureFlow(A)>
```

The new software is not reflected immediately when download is complete. Restart the device after the download is complete.

14.1.6 Downloading software via FTP

The software can be downloaded to the device via FTP. The downloaded software is automatically saved to the internal flash memory. The software of the old version is saved to a different area, and the new software is written. During version upgrade, be careful not to turn off the power of the device. If the power of the device is turned off during the operation, the device will load the old software saved in a different area. In this case, restart the device and try the download operation again. If the communication is discontinued during download, start the downloading work again.

To download the software to the device, use the following command. Specify the correct IP address for the system interface in advance to enable communication with the FTP server. For more information on system interface settings, see Chapter 7 “System Interface Settings”. Provide a user name and password for the FTP server used for downloading.

```
PureFlow(A)> download ftp obj 192.168.100.40 nf7600.bin
Name:ftpuer (Input a user name.)
Password: (Input a password.)
Download “nf7600.bin” from 192.168.100.40 (y/n)? y
Loading ...
creating Backup from Master file.....completed.
Done.
PureFlow(A)>
```

The new software is not reflected immediately when download is complete. Restart the device after the download is complete.

14.1.7 Downloading software via WebGUI

The software can be downloaded to the device via WebGUI. The downloaded software is automatically saved to the internal flash memory. The software of the old version is saved to a different area, and the new software is written. During version upgrade, be careful not to turn off the power of the device. If the power of the device is turned off during the operation, the device will load the old software saved in a different area. In this case, restart the device and try the download operation again. If the communication is discontinued during download, start the downloading work again.

For more information about WebGUI, see WebGUI Operating Manual (NF7600-W014E). Specify the correct IP address for the system interface in advance to enable communication with WebGUI. For more information on system interface settings, see Chapter 7 “System Interface Settings”.

The new software is not reflected immediately when download is complete. Restart the device after the download is complete.

14.2 Downloading the Software Update Patch

The software of this device can be updated to the new version by downloading a software update patch. The size of the software update patch is small, which reduces the download time. The procedure is the same as for downloading software. When the software update patch is downloaded, it is automatically implemented. When the download finishes, restart the device to apply the new software.

For information on how to obtain a software update patch, contact an Anritsu Networks sales engineer.

14.2.1 Downloading software Update Patch from a CF card

Insert a CF card with the software update patch into the CF card slot to download the software update patch to this device. During version upgrade, be careful not to remove the CF card or turn off the power of the device. If the CF card is removed or the power of the device is turned off during the operation, the device will load the old software saved in a different area. In this case, restart the device and try the download operation again.

```
PureFlow(A)> download cf patch
Apply patch from Flash Memory Card (y/n)? y
Applying file system patch ..... done
Applying apps patch ..... done
Applying fcpu patch ..... done
creating Backup from Master file.....completed.
Done.
PureFlow(A)>
```

The new software is not reflected immediately when download is complete. Restart the device after the download is complete.

14.2.2 Downloading software Update Patch from a USB flash drive

Insert a USB flash drive with the software update patch into the USB port to download the software update patch to this device. During version upgrade, be careful not to remove the USB flash drive or turn off the power of the device. If the USB flash drive is removed or the power of the device is turned off during the operation, the device will load the old software saved in a different area. In this case, restart the device and try the download operation again.

```
PureFlow(A)> download usb patch
Apply patch from USB Memory (y/n)? y
Applying file system patch ..... done
Applying apps patch ..... done
Applying fcpu patch ..... done
creating Backup from Master file.....completed.
Done.
PureFlow(A)>
```

The new software is not reflected immediately when download is complete. Restart the device after the download is complete.

14.3 Downloading/Uploading Configuration Data

Caution on downloading configuration data

Download the configuration file uploaded to the CF card, USB memory, TFTP server, and FTP server with the upload command described above. If any configuration file other than proper configuration file is downloaded, the device may not start up. If an incorrect configuration file was downloaded, insert the CF card or USB memory containing the proper configuration file (file name: extcnf.txt), and start the equipment.

After that, save the setting contents by using the save command.

14.3.1 Downloading configuration data from a CF card

Insert a CF card into the CF card slot to download the new configuration file to the device. The downloaded configuration file is automatically saved to the internal flash memory. The configuration file of the old version is saved to a different area, and the new configuration file is written. The new configuration is not reflected immediately when download is complete. Restart the device after the download is complete. During the download, be careful not to remove the CF card or turn off the power of the device. If the CF card is removed or the power of the device is turned off during the operation, the device will load the old configuration file saved in a different area. In this case, restart the device and try the download operation again.

```
PureFlow(A)> download cf conf config.txt
Download "config.txt" from Flash Memory Card (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

The new configuration is not reflected immediately when download is complete. Restart the device after the download is complete.

14.3.2 Uploading configuration data to the CF card

Insert a CF card into the CF card slot to upload the configuration file to the CF card. The uploaded configuration file is saved to the inserted CF card.

```
PureFlow(A)> upload cf conf config.txt
Upload as "config.txt" to Flash Memory Card (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

The configuration information saved to the internal flash memory rather than the operating configuration information is uploaded. The configuration information is saved to the internal flash memory when the save config command is executed.

14.3.3 Downloading configuration data from a USB flash drive

Insert a USB flash drive into the USB port to download the new configuration file to the device. The downloaded configuration file is automatically saved to the internal flash memory. The configuration file of the old version is saved to a different area, and the new configuration file is written. The new configuration is not reflected immediately when download is complete. Restart the device after the download is complete. During the download, be careful not to remove the USB flash drive or turn off the power of the device. If the USB flash drive is removed or the power of the device is turned off during the operation, the device will load the old configuration file saved in a different area. In this case, restart the device and try the download operation again.

```
PureFlow(A)> download usb conf config.txt
Download "config.txt" from USB Memory (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

The new configuration is not reflected immediately when download is complete. Restart the device after the download is complete.

14.3.4 Uploading configuration data to a USB flash drive

Insert a USB flash drive into the USB port to upload the configuration file to the USB flash drive. The uploaded configuration file is saved to the inserted USB flash drive.

```
PureFlow(A)> upload usb conf config.txt
Upload as "config.txt" to USB Memory (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

The configuration information saved to the internal flash memory rather than the operating configuration information is uploaded. The configuration information is saved to the internal flash memory when the save config command is executed.

14.3.5 Downloading configuration data via TFTP

The configuration file can be downloaded to this device via TFTP. The downloaded configuration file is automatically saved to the internal flash memory. The configuration file of the old version is saved to a different area, and the new configuration file is written. The new configuration is not reflected immediately when download is complete. Restart the device after the download is complete. During the download, be careful not to turn off the power of the device. If the power of the device is turned off during the operation, the device will load the old configuration file saved in a different area. In this case, restart the device and try the download operation again. If the communication is discontinued during download, start the downloading work again.

To download the configuration file to the device, use the following command. Specify the correct IP address for the system interface in advance to enable communication with the TFTP server. For more information on system interface settings, see Chapter 7 “System Interface Settings”.

```
PureFlow(A)> download tftp conf 192.168.100.40 config.txt
Download “config.txt” from 192.168.100.40 (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

The new configuration is not reflected immediately when download is complete. Restart the device after the download is complete.

14.3.6 Uploading configuration data via TFTP

The configuration file can be uploaded to the TFTP server via TFTP. The uploaded configuration file is saved in the TFTP server.

```
PureFlow(A)> upload tftp conf 192.168.100.40 config.txt
Upload as “config.txt” to 192.168.100.40 (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

The configuration information saved to the internal flash memory rather than the operating configuration information is uploaded.

14.3.7 Downloading configuration data via FTP

The configuration file can be downloaded to the device via FTP. The downloaded configuration file is automatically saved to the internal flash memory. The configuration file of the old version is saved to a different area, and the new configuration file is written. The new configuration is not reflected immediately when download is complete. Restart the device after the download is complete. During the download, be careful not to turn off the power of the device. If the power of the device is turned off during the operation, the device will load the old configuration file saved in a different area. In this case, restart the device and try the download operation again. If the communication is discontinued during download, start the downloading work again.

To download the configuration file to the device, use the following command. Specify the correct IP address for the system interface in advance to enable communication with the FTP server. For more information on system interface settings, see Chapter 7 “System Interface Settings”. Provide a user name and password for the FTP server used for downloading.

```
PureFlow(A)> download ftp conf 192.168.100.40 config.txt
Name:ftpuser (Input a user name.)
Password: (Input a password.)
Download “config.txt” from 192.168.100.40 (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

The new configuration is not reflected immediately when download is complete. Restart the device after the download is complete.

14.3.8 Uploading configuration data via FTP

The configuration file can be uploaded to the FTP server via FTP. The uploaded configuration file is saved in the FTP server.

```
PureFlow(A)> upload ftp conf 192.168.100.40 config.txt
Name:ftpuser (Input a user name.)
Password: (Input a password.)
Upload as “config.txt” to 192.168.100.40 (y/n)? y
Loading ...
Done.
PureFlow(A)>
```

The configuration information saved to the internal flash memory rather than the operating configuration information is uploaded.

14.4 Restarting the Software

After the download is complete, restart the device with the new software.

- (1) Restart the device

To restart the device, turn off the power and turn it on again, or use the following command:

```
PureFlow(A)> reboot system
Rebooting the system, ok(y/n)? y
```

- (2) Confirm the start-up file

If the baud rate of the serial console is set to 9600bps, the start-up file type and result of the CRC check will be displayed at startup:

```
reading :Object
checkCRC:OK
```

If a download operation is abnormally terminated with power failure, etc. The Master file may be a CRC error, and the device will load the Backup file. Please download again after starting with the Backup file:

```
reading :Object
checkCRC:NG
reading :Backup
checkCRC:OK
```

The table below lists the start-up file types.

Display	Description	Priority
/dev/usb1	The file on a USB flash drive.	High ↑ ↓ Low
/dev/externalcf1	The file on a CF card.	
Object	The Master file.	
Backup	The Backupfile.	

- (3) Confirm the completion of restart

For a restart, Telnet/SSH connection is disconnected. After the device starts up, login again via Telnet/SSH.

(Blank page)

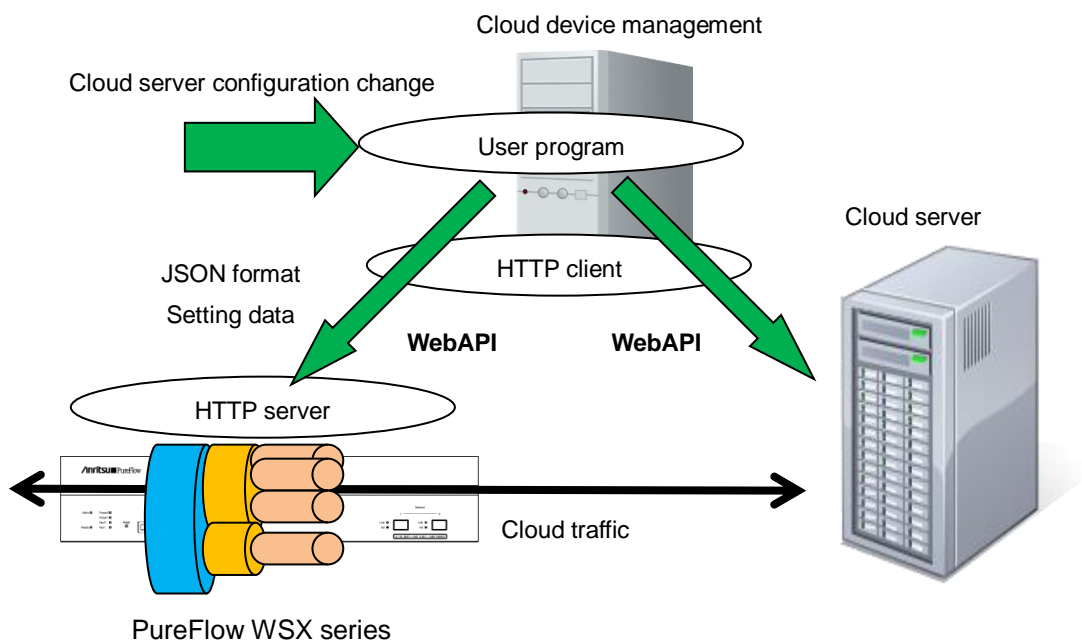
This chapter describes the WebAPI (Web Application Program Interface) feature.

15.1	Overview	15-2
15.2	Communication Protocol	15-3
15.3	HTTP Methods.....	15-3
15.4	JSON Format.....	15-4
15.5	API List	15-5
15.6	Common Error Messages	15-6
15.7	List of Error Messages	15-7

15.1 Overview

The WebAPI feature is used to configure the traffic control feature of the system via HTTP (Hypertext Transfer Protocol: RFC2616). This device functions as an HTTP server, and can be configured using the JSON format (JavaScript Object Notation: RFC4627) via the HTTP client on the external management terminal.

In a cloud environment, it is becoming difficult to manually update the traffic control settings of the network device in accordance with the update of the cloud server configuration. To update the settings of this device automatically, use a programming language supporting the JSON format on the cloud management terminal to create a user program to update the traffic control settings of this device.



You can also use an HTTP connection via SSL encryption (HTTPS: Hypertext Transfer Secure). WebAPI communication is encrypted via HTTPS, which helps prevent eavesdropping and spoofing.

WebAPI is simultaneously available up to 4 clients maximum.

If you run the WebAPI of 5 or more sessions at the same time, you can connect even more than 5 sessions, but an error occurs in any session when sending requests. For example, if you send a request for a 5th session while you are running 1-4 sessions, disconnection or the number of sessions exceeded error occurs in any of 1-5 sessions. Please use WebAPI within four sessions.

HTTP request and the time-out period until the next HTTP request are 15 seconds.

15.2 Communication Protocol

The WebAPI feature use HTTP or HTTPS as the communication protocol. To set up the communication protocol, use the following commands.

set http protocol {normalhttp httpsecure}	Sets the communication protocol of the Web application. normalhttp: Use HTTP. httpsecure: Use HTTPS. It is not possible to simultaneously use of HTTP and HTTPS.
show http	Displays the settings of the Web application.

15.3 HTTP Methods

The WebAPI feature supports the following HTTP methods:

HTTP Method	Usage
HEAD	Used to determine access permissions.
GET	Used to get information. This device uses this for requests to get information.
POST	Used to set information. This device uses this for requests to add, update, and delete information.

If an HTTP client specifies any method other than the above, the HTTP status code 405 (Method Not Allowed) is returned.

15.4 JSON Format

WebAPI uses JSON format data via the GET and POST methods. JSON is a data description language. In the JSON format, a key and value pair delimited by a colon “:” is used as a parameter. Multiple parameters are delimited by commas “,”. These are, as a whole, enclosed by curly brackets “{ “and”}”.

For WebAPI, all keys and values must be specified as strings. Specify the key “command” (API type), and a CLI command parameter (API content). For WebAPI, keys can be specified in random order. They need not be consistent with the CLI command parameter order.

The following JSON example uses API to add a scenario:

```
{
    "command": "add scenario"
    "scenario_name": "/port1/North",
    "action": "aggregate",
    "min_bandwidth": "5M",
    "peak_bandwidth": "8M",
    "class": "2",
    "bufsize": "512k"
}
```

For details of the JSON format, see Appendix E “JSON format”

15.5 API List

WebAPI provides API features to get and set scenario, filter, and rule list information. These features are consistent with relevant CLI commands. The parameters specified for API and the value scope and required/optional settings are also consistent. For details of API features, see Appendix F “Details of WebAPI”.

Target	Action	Relevant CLI command
Scenario	Add	add scenario
	Update	update scenario
	Delete	delete scenario
	Get information	show scenario
Application acceleration	Add	add apl-accel
	Update	update apl-accel
	Delete	delete apl-accel
Filter	Add	add filter
	Delete	delete filter
	Get information	show filter
Rule list	Add a group	add rulelist group
	Delete a group	delete rulelist group
	Add an entry	add rulelist entry
	Delete an entry	delete rulelist entry
	Get information	show rulelist
Channel	Add	add channel
	Delete	delete channel
	Get information	show channel
Channel interface	Set	set ip channel
	Release	unset ip channel
	Get information	show ip channel
OpenFlow controller	Add	add openflow controller
	Delete	delete openflow controller
	Get information	show openflow controller
Channel interface static path	Add	add route
	Delete	delete route
	Get information	show route target
Configuration	Save	save config
	Get information	show save status*

* The API to get configuration information returns the status indicating whether the configuration is being saved. A configuration cannot be saved simultaneously while another configuration is being saved. For the time required for saving, see Chapter 3 “Configuring Settings”.

15.6 Common Error Messages

If the HTTP method and JSON format are correct but the specified content is invalid, an error message is returned in addition to HTTP status code 200 (OK). Common error messages are as follows:

Error Messages	Description
Specified command is invalid.	The API command is invalid. Check whether the specified JSON-format key and value are correct.
Required parameter is not specified.	The required parameter is not specified. Check whether the specified JSON-format key and value are correct.
Specified command is invalid when GET request.	The command (add/update/delete) cannot be specified by the GET method. Check whether the specified JSON-format key and value are correct.
Specified command is invalid when POST request.	The command (get) cannot be specified by the POST method. Check whether the specified JSON-format key and value are correct.
WebAPI session is full.	Maximum WebAPI sessions exceeded. Execute again after a while.
Failed to create pipe.	Cannot create the pipe for internal communication. Execute again after a while.
No response message from LR.	No response from internal software. Execute again after a while.

15.7 List of Error Messages

Specific API error messages are as follows:

API	Error Messages
Add a scenario	Specified Scenario Class is invalid. It must be either of 1,2,3,4,5,6,7,8. • The specified class is invalid.
	Specified scenario fail action class is invalid.It must be either of 1, 2, 3, 4, 5, 6, 7, 8. • The specified Fail Action class is invalid.
	Specified Minimum Bandwidth is invalid. (Valid from 0, 10 k to 10 G) • The specified Minimum Bandwidth is invalid.
	Specified Peak Bandwidth is invalid. (Valid from 10 k to 10 G) • The specified Peak Bandwidth is invalid.
	Specified fail action minimum bandwidth is invalid. (Valid from 0, 10k to 10G) • The specified Fail Action Minimum Bandwidth is invalid.
	Specified fail action peak bandwidth is invalid. (Valid from 10k to 10G) • The specified Fail Action Peak Bandwidth is invalid.
	Peak Bandwidth should be greater than Minimum Bandwidth. • peak_bandwidth must be equal to or greater than min_bandwidth.
	Specified Buff Size is invalid. (Valid from 2 k to 1G) • The specified bufsize is invalid.
	Specified Scenario Name is invalid. •The specified scenario name is invalid.
	Specified Scenario Name is already used. • The specified scenario name has already been used for another scenario.
	Specified Scenario of upper level hierarchy is not found. • The upper level scenario does not exist.
	maximum number of scenario was exceeded. • The number of scenarios exceeds the registration limit.
	Specified Scenario ID is invalid. (Valid from 1 to 40000) • The scenario index is out of range.
	Specified Scenario ID is already used. • The specified scenario index has already been used for another scenario.
	Specified Max Q Num is invalid. (Valid from 1 to 4096) • maxquenum is out of range.
Specified Q Division Field is invalid. Valid fields: default, vlan, cos, inner-vlan, inner-cos, ethertype, sip, dip, tos, proto, sport, dport • The specified quedivision field is invalid.	

API	Error message
Add a scenario (Continued)	failaction is not specified. • fail_min_bw, fail_peak_bw, and fail_class cannot be specified without specifying failaction.
	Specified Failaction is invalid. • fail_min_bw, fail_peak_bw, and fail_class can be specified only when forwardattribute is specified as failaction.
	Invalid IP address • Specified IP address format or value is invalid.
	Peer IP version and second-peer IP version are different. • The IP versions of peer and second-peer shall be matched.
	Peer and second-peer are same IP_address. • The IP addresses for peer and second-peer should be different.
	Specified dport is invalid. (Valid from 10001 to 20000) • The specified dport is invalid.
	Specified Dport is already used. • The specified dport is used in another scenario.
	Specified vid is invalid. (Valid from 1 to 4094) • The specified VLAN ID is invalid.
	Specified inner-vid is invalid. (Valid from 1 to 4094) • The specified Inner-VLAN ID is invalid.
	VID must be specified when inner-VID is specified. • Inner VLAN ID can be specified only when VLAN ID is specified.
	Specified cos is invalid. (Valid from 0 to 7) • The specified CoS value is invalid.
	Specified inner-cos is invalid. (Valid from 0 to 7) • The specified CoS value is invalid.
	VID must be specified when CoS is specified. • The CoS value can be specified only when the VLAN ID is specified.
	Inner-VID must be specified when inner-cos is specified. • The Inner-CoS value can be specified only when the Inner VLAN ID is specified.
	Specified dscp is invalid. (Valid from 0 to 63) • The specified DSCP value is invalid.
	Specified tcp-mem is invalid. (Valid from 64k to 200M) • The specified TCP buffer size is invalid.
	Specified bypass threshold RTT is invalid. (Valid from 0 to 10000) • The specified RTT threshold value of auto bypass of traffic acceleration is invalid.
Specified peak bandwidth is not licensed. • The license of the specified bandwidth does not exist.	

API	Error message
Add a scenario (Continued)	Data block size should be divided by fec block size. • For the data block size, a value that can be divided by the FEC block size shall be set.
	Data block size should be greater than fec block size. • For the data block size, a value that is larger than the FEC block size shall be set.
	Specified fec block size is invalid. (Valid from 2K to 50K) • The specified FEC block size is invalid.
	Specified data block size is invalid. (Valid from 2K to 200K) • The specified data block size is invalid.
	Specified fec session is invalid. (Valid from 0 to 1000) • The specified FEC session count is invalid.
	FEC function is not licensed. • The license of the TCP-FEC function does not exist.
	Specified fec flag is invalid. It must be either of compression enable or fec enable. • Either the compression or the TCP-FEC function can be enabled.
	Maximum number of secondary peer was exceeded. • Exceeds the maximum registered numbers of the scenario that specifies second-peer.
	Maximum number of keep alive scenario was exceeded. • Exceeds the maximum registered numbers of the scenario that enables bypass-keep.
	Specified scenario has packets in buffer. Please wait until the buffer becomes empty, and try again. • The specified scenario is transmitting a packet. Please wait until the completion of transmission, and retry again.

API	Error Messages
Update a scenario	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified Scenario Class is invalid. It must be either of 1,2,3,4,5,6,7,8. • The specified class is invalid.
	Specified scenario fail action class is invalid. It must be either of 1, 2, 3, 4, 5, 6, 7, 8. • The specified Fail Action class is invalid.
	Specified Minimum Bandwidth is invalid. (Valid from 0, 10 k to 10 G) • The specified Minimum Bandwidth is invalid.
	Specified fail action minimum bandwidth is invalid. (Valid from 0, 10k to 10G) • The specified Fail Action Minimum Bandwidth is invalid.
	Specified Peak Bandwidth is invalid. (Valid from 10 k to 10 G) • The specified Peak Bandwidth is invalid.
	Specified fail action peak bandwidth is invalid. (Valid from 10k to 10G) • The specified Fail Action Peak Bandwidth is invalid.
	Peak Bandwidth should be greater than Minimum Bandwidth. • peak_bandwidth must be equal to or greater than min_bandwidth.
	Specified Buff Size is invalid. (Valid from 2 k to 1G) • The specified bufsize is invalid.
	It is necessary to set one or more parameters. • At least one parameter must be set.
	Specified Scenario Mode is invalid. • The specified scenario mode is invalid.
	Specified Max Q Num is invalid. (Valid from 1 to 300000) • maxquenum is out of range.
	Extended number of scenario is not licensed. • Scenario cannot be registered when exceeding the limit number of the scenario expansion licenses. • maxquenum cannot be registered, which exceeds the limit number of the scenario expansion licenses.
	Specified Q Division Field is invalid. Valid fields: default, vlan, cos, inner-vlan, inner-cos, ethertype, sip, dip, tos, proto, sport, dport • The specified quedivision field is invalid.
Fail action forward is incorrect. Specified Failaction is invalid. • fail_min_bw, fail_peak_bw, and fail_class can be specified only when forwardattribute is specified as failaction.	

API	Error Messages
Update a scenario (Continued)	Invalid IP address <ul style="list-style-type: none"> Specified IP address format or value is invalid.
	Specified cos is invalid. (Valid from 0 to 7) <ul style="list-style-type: none"> The specified CoS value is invalid.
	Specified inner-cos is invalid. (Valid from 0 to 7) <ul style="list-style-type: none"> The specified CoS value is invalid.
	Specified dscp is invalid. (Valid from 0 to 63) <ul style="list-style-type: none"> The specified DSCP value is invalid.
	Specified tcp-mem is invalid. (Valid from 64k to 200M) <ul style="list-style-type: none"> The specified TCP buffer size is invalid.
	Specified bypass threshold RTT is invalid. (Valid from 0 to 10000) <ul style="list-style-type: none"> The specified RTT threshold value of auto bypass of traffic acceleration is invalid.
	Specified peak bandwidth is not licensed. <ul style="list-style-type: none"> The license of the specified bandwidth does not exist.
	Data block size should be divided by fec block size. <ul style="list-style-type: none"> For the data block size, a value that can be divided by the FEC block size shall be set.
	Data block size should be greater than fec block size. <ul style="list-style-type: none"> For the data block size, a value that is larger than the FEC block size shall be set.
	Specified fec block size is invalid. (Valid from 2K to 50K) <ul style="list-style-type: none"> The specified FEC block size is invalid.
	Specified data block size is invalid. (Valid from 2K to 200K) <ul style="list-style-type: none"> The specified data block size is invalid.
	Specified fec session is invalid. (Valid from 0 to 1000) <ul style="list-style-type: none"> The specified FEC session count is invalid.
	FEC function is not licensed. <ul style="list-style-type: none"> The license of the TCP-FEC function does not exist.
	Maximum number of keep alive scenario was exceeded. <ul style="list-style-type: none"> Exceeds the maximum registered numbers of the scenario that enables bypass-keep.

API	Error Messages
Delete a scenario	Specified Scenario Name is invalid. <ul style="list-style-type: none"> The specified scenario name is invalid.
	Specified scenario name is not used. <ul style="list-style-type: none"> The specified scenario does not exist.
	Down level hierarchy scenario exists. <ul style="list-style-type: none"> A lower level scenario exists.

API	Error Messages
Get scenario information	Specified Scenario Name is invalid. <ul style="list-style-type: none"> • The specified scenario name is invalid.
	Specified scenario name is not used. <ul style="list-style-type: none"> • The specified scenario does not exist.

API	Error message
Add application acceleration	Specified Scenario Name is invalid. <ul style="list-style-type: none"> • The specified scenario name is invalid.
	Specified scenario name is not used. <ul style="list-style-type: none"> • The specified scenario does not exist.
	Specified scenario name is not wan-accel mode. <ul style="list-style-type: none"> • The specified scenario is not the acceleration mode.
	Specified Scenario Name is already used. <ul style="list-style-type: none"> • The SMB acceleration has already been set in the specified scenario.
	Specified tcp port is invalid.(Valid from 0 to 65535) (Up to 16 ports can be specified with separated comma without space) <ul style="list-style-type: none"> • The specified SMB TCP Port is invalid.
	Specified smb session is invalid. (Valid from 0 to 1000) <ul style="list-style-type: none"> • The specified SMB Session is invalid.
	Specified read cache size is invalid. (Valid from 64k to 60M) <ul style="list-style-type: none"> • The specified Read Cache Size is invalid.

API	Error message
Update application acceleration	Specified channel name is invalid. <ul style="list-style-type: none"> • The specified channel name is invalid.
	Channel name already exists. <ul style="list-style-type: none"> • The specified channel name has already been used in another channel.
	Slot #N is invalid. <ul style="list-style-type: none"> • The specified slot is invalid.
	Specified tcp port is invalid.(Valid from 0 to 65535) (Up to 16 ports can be specified with separated comma without space) <ul style="list-style-type: none"> • The specified SMB TCP Port is invalid.
	Specified smb session is invalid. (Valid from 0 to 1000) <ul style="list-style-type: none"> • The specified SMB Session is invalid.
	Specified read cache size is invalid. (Valid from 64k to 60M) <ul style="list-style-type: none"> • The specified Read Cache Size is invalid.

API	Error message
Delete application acceleration	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified scenario name is not wan-accel mode. • The specified scenario is not the acceleration mode.
	Specified protocol is already disabled. • The specified protocol is already invalid.

API	Error Messages
Add a filter	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified filter Name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid Filter Name length is from 1 to 48.) • The specified filter name is invalid.
	Specified filter Name is already used. • The specified filter name has already been used for another filter.
	Specified Ether type is invalid. (Valid from 0x0000 to 0xFFFF) • The specified Ethernet type is invalid.
	Specified vid is invalid. (Valid from 0 to 4094, Or Start - End) • The specified VLAN ID is invalid.
	Specified cos is invalid. (Valid from 0 to 7, Or Start - End) • The specified CoS value is invalid.
	Specified inner-vid is invalid. (Valid from 0 to 4094, Or Start - End) • The specified VLAN ID is invalid.
	Specified inner-cos is invalid. (Valid from 0 to 7, Or Start - End) • The specified CoS value is invalid.
	The format or value of the specified source IP address is invalid. • The specified source IP address is invalid.
	The format or value of the specified destination IP address is invalid. • The specified destination IP address is invalid.
	The format or value of the specified source IPv6 address is invalid. • The specified Source IPv6 address is invalid.

API	Error Messages
Add a filter (Continued)	The format or value of the specified destination IPv6 address is invalid. • The specified Destination IPv6 address is invalid.
	Specified rulelist name of source IP address is invalid. Specified rulelist name of destination IP address is invalid. Specified rulelist name of source port is invalid. Specified rulelist name of destination port is invalid. • The rule list name is invalid.
	Specified rulelist name of source IP address is not used. Specified rulelist name of destination IP address is not used. Specified rulelist name of source port is not used. Specified rulelist name of destination port is not used. • The specified rule list does not exist.
	IP Filter and rulelist of source IP address is not same type. IP Filter and rulelist of destination IP address is not same type. IP Filter and rulelist of source port is not same type. IP Filter and rulelist of destination port is not same type. • The type is different from that of the target rule list.
	Specified ToS is invalid. (Valid from 0 to 255, Or Start - End) • The specified ToS value or Traffic Class value is invalid.
	Specified protocol number is invalid. (Valid from 0 to 255, Start - End, Or tcp/udp/icmp) • The specified protocol number is invalid.
	Specified Source TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) • The specified sport number is invalid.
	Specified Destination TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) • The specified dport number is invalid.
	Specified Filter Priority is invalid. (Valid from 1 to 40000) • The specified filter priority is invalid.
	maximum number of filter was exceeded. • The number of registered filters exceeds the registration limit.
	It is necessary to set one or more parameters other than Priority. • For the Ethernet filter, specify at least one parameter in addition to Priority.
	Could not Add the Filter. • The filter cannot be registered.

API	Error Messages
Delete a filter	Specified scenario name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified filter name is invalid. (Number only cannot be specified. “all” cannot be specified.) (Valid Filter Name length is from 1 to 48.) • The specified filter name is invalid.
	Specified filter name is not used. • The specified filter does not exist.

API	Error Messages
Get filter information	Specified scenario name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified filter name is invalid. (Number only cannot be specified. “all” cannot be specified.) (Valid Filter Name length is from 1 to 48.) • The specified filter name is invalid.
	Specified filter name is not used. • The specified filter does not exist.

API	Error Messages
Add a rule list group	Specified rulelist name is invalid. (Number only cannot be specified. “all” cannot be specified.) (Valid rulename length is from 1 to 32.) • The rule list name is invalid.
	Specified rulelist name is already in use. • A rule list with the same name already exists.
	Maximum number of rulelist was exceeded. • The number of rule lists exceeds the registration limit.

API	Error Messages
Delete a rule list group	Specified rulelist name is invalid. (Number only cannot be specified. “all” cannot be specified.) (Valid rulename length is from 1 to 32.) • The rule list name is invalid.
	Specified rulelist name is not used. • The specified rule list does not exist.
	Rulelist is used by filter. • The rule list is set in a filter.

API	Error Messages
Add a rule list entry	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid rulename length is from 1 to 32.) <ul style="list-style-type: none"> • The rule list name is invalid.
	Specified rulelist name is not used. <ul style="list-style-type: none"> • The specified rule list does not exist.
	The format or value of the specified IP address is invalid. <ul style="list-style-type: none"> • The specified IP address is invalid.
	Specified TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) <ul style="list-style-type: none"> • The specified TCP/UDP port number is invalid.
	Maximum number of rulelist entry was exceeded. <ul style="list-style-type: none"> • The number of entries for the specified rule list exceeds the limit (512 records).
	Maximum number of total rulelist entry was exceeded. <ul style="list-style-type: none"> • The number of entries of all rule lists exceeds the registration limit (64000 records).
	Specified rulelist entry is already in use. <ul style="list-style-type: none"> • The specified rule list entry had already been registered.
	Rulelist entry and rulelist is not same type. <ul style="list-style-type: none"> • The type is different from that of the target rule list.

API	Error Messages
Delete a rule list entry	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid rulename length is from 1 to 32.) <ul style="list-style-type: none"> • The rule list name is invalid.
	Specified rulelist name is not used. <ul style="list-style-type: none"> • The specified rule list does not exist.
	The format or value of the specified IP address is invalid. <ul style="list-style-type: none"> • The specified IP address is invalid.
	Specified TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) <ul style="list-style-type: none"> • The specified TCP/UDP port number is invalid.
	Rulelist entry and rulelist is not same type. <ul style="list-style-type: none"> • The type is different from that of the target rule list.
	Specified rulelist entry is not used. <ul style="list-style-type: none"> • The specified rule list entry does not exist.

API	Error Messages
Get rule list information	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid rulename length is from 1 to 32.) <ul style="list-style-type: none"> The rule list name is invalid.
	Specified rulelist name is not used. <ul style="list-style-type: none"> The specified rule list does not exist.

API	Error Messages
Add channel	Specified channel name is invalid. <ul style="list-style-type: none"> The specified channel name is invalid.
	Channel name already exists. <ul style="list-style-type: none"> The specified channel name has already been used in another channel.
	Slot #N is invalid. <ul style="list-style-type: none"> The specified slot is invalid.
	Port <slot/port> is invalid. <ul style="list-style-type: none"> The specified port is invalid.
	Specified group name is invalid. <ul style="list-style-type: none"> The specified group name is invalid.
	Specified group name is not used. <ul style="list-style-type: none"> The specified group does not exist.
	Specified vid is invalid. (Valid from 1 to 4094) <ul style="list-style-type: none"> The specified VLAN ID is invalid.
	Specified TPID is invalid. (Valid 0x8100,0x88a8,0x9100,0x9200 or 0x9300.) <ul style="list-style-type: none"> The specified TPID is invalid.
	Specified inner-vid is invalid. (Valid from 1 to 4094) <ul style="list-style-type: none"> The specified Inner-VLAN ID is invalid.
	VID must be specified when inner-VID is specified. <ul style="list-style-type: none"> Inner VLAN ID can be specified only when VLAN ID is specified.
	Specified mtu is invalid. (Valid from 300 to 10200) <ul style="list-style-type: none"> The specified mtu is invalid.
	Specified vid and inner-vid is already used on channel "channel name". <ul style="list-style-type: none"> The specified vid and inner-vid have already been used in the "channel name" channel.
	Specified port is already used on other default-channel. <ul style="list-style-type: none"> The specified port has already been used for another default channel.
	Maximum number of channel was exceeded. <ul style="list-style-type: none"> Exceeds the maximum number of registered channels.

API	Error Messages
Delete channel	Specified channel name is invalid. <ul style="list-style-type: none"> • The specified channel name is invalid.
	Specified channel name is not used. <ul style="list-style-type: none"> • The specified channel does not exist.

API	Error Messages
Get channel information	Specified channel name is invalid. <ul style="list-style-type: none"> • The specified channel name is invalid.
	Specified channel name is not used. <ul style="list-style-type: none"> • The specified channel does not exist.

API	Error Messages
Interface setting	Specified channel name is invalid. <ul style="list-style-type: none"> • The specified channel name is invalid.
	Specified channel name is not used. <ul style="list-style-type: none"> • The specified channel does not exist.
	Invalid IP address <ul style="list-style-type: none"> • Specified IP address format or value is invalid.
	Invalid netmask <ul style="list-style-type: none"> • The format or value of the specified subnet mask is invalid.
	Default-channel cannot be set for this command. <ul style="list-style-type: none"> • The default channel cannot be set.

API	Error Messages
Release interface	Specified channel name is invalid. <ul style="list-style-type: none"> • The specified channel name is invalid.
	Specified channel name is not used. <ul style="list-style-type: none"> • The specified channel does not exist.
	Cannot specified "ipv4" or "ipv6". <ul style="list-style-type: none"> • "IPv4" and "IPv6" cannot be specified for "all".

API	Error Messages
Show interface information	Specified channel name is invalid. <ul style="list-style-type: none"> • The specified channel name is invalid.
	Specified channel name is not used. <ul style="list-style-type: none"> • The specified channel does not exist.
	IP interface is not configured. <ul style="list-style-type: none"> • IP address is not set for the specified channel or the next channel.

API	Error Messages
Add static path	Route entry already exists. • This is an already-existing route entry.
	Invalid IP address • Specified IP address format or value is invalid.
	Invalid netmask • The format or value of the specified subnet mask is invalid. • The value of the specified prefix length is invalid.
	Invalid gateway • The gateway IP address format or value is invalid.
	Default-channel cannot be set for this command. • The default channel cannot be specified.
	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.
	Target IP address and gateway is not same IP version. • The destination IP address and gateway IP address versions do not match.
	Maximum number of route was exceeded. • Exceeds the maximum number of registered static path.

API	Error Messages
Delete static path	Invalid IP address • Specified IP address format or value is invalid.
	Invalid netmask • The format or value of the specified subnet mask is invalid.
	Invalid gateway • The gateway IP address format or value is invalid.
	Route info is not found. • The specified static path does not exist.
	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.
	Target IP address and gateway is not same IP version. • The destination IP address and gateway IP address versions do not match.

API	Error Messages
Display static path information	Specified channel name is invalid. <ul style="list-style-type: none"> • The specified channel name is invalid.
	Specified channel name is not used. <ul style="list-style-type: none"> • The specified channel does not exist.

API	Error Messages
Add OpenFlow controller	Specified IP address already used. <ul style="list-style-type: none"> • The specified IP address is already used.
	The format of value of the specified IP address is invalid. <ul style="list-style-type: none"> • The specified IP address is invalid.
	Specified TCP port number is invalid.(Valid from 1 to 65535) <ul style="list-style-type: none"> • The specified TCP port number is invalid.
	Maximum number of openflow controller was exceeded. <ul style="list-style-type: none"> • The maximum number of OpenFlow controllers registered was exceeded.
	System busy: Another conflicting command is in progress. <ul style="list-style-type: none"> • The OpenFlow command is in execution.
	System busy: Please try again later. <ul style="list-style-type: none"> • The OpenFlow command was timed out.
	OpenFlow function is not licensed. <ul style="list-style-type: none"> • The OpenFlow function is not licensed.

API	Error Messages
Delete OpenFlow controller	Specified IP address is not used. <ul style="list-style-type: none"> • The specified IP address does not exist.
	The format of value of the specified IP address is invalid. <ul style="list-style-type: none"> • The specified IP address is invalid.
	System busy: Another conflicting command is in progress. <ul style="list-style-type: none"> • The OpenFlow command is in execution.
	System busy: Please try again later. <ul style="list-style-type: none"> • The OpenFlow command was timed out.
	OpenFlow function is not licensed. <ul style="list-style-type: none"> • The OpenFlow function is not licensed.

API	Error Messages
Display OpenFlow controller information	No OpenFlow controller is set. <ul style="list-style-type: none">• The OpenFlow controller has not yet been registered.
	System busy: Another conflicting command is in progress. <ul style="list-style-type: none">• The OpenFlow command is in execution.
	OpenFlow function is not licensed. <ul style="list-style-type: none">• The OpenFlow function is not licensed.

API	Error Messages
Save configuration	configuration save is in progress <ul style="list-style-type: none">• The configuration save is in progress
Get configuration information	None

(Blank page)

Chapter 16 *OpenFlow Function*

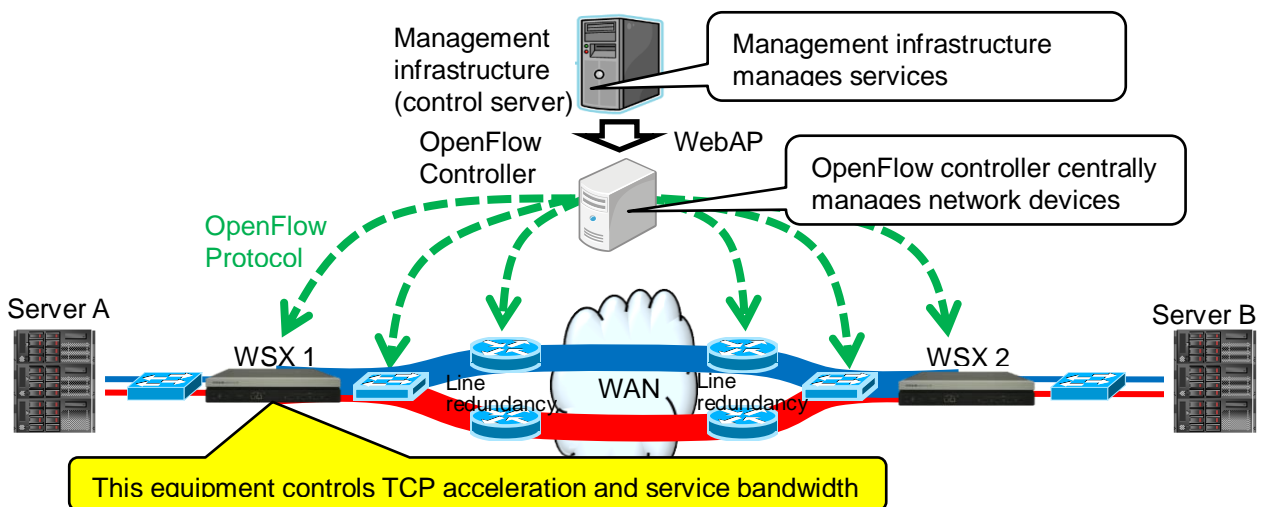
This chapter describes the OpenFlow function.

16.1	Overview	16-2
16.2	OpenFlow Version	16-3
16.3	Supported OpenFlow Messages	16-4
16.4	OpenFlow Messages Supported for CLI Commands	16-6
16.5	JSON Format.....	16-7
16.6	Supported Command List.....	16-8
16.7	Common Error Messages	16-9
16.8	Error Message List.....	16-10

16.1 Overview

The OpenFlow function uses the OpenFlow protocol for setting of the traffic control function of this equipment. This equipment can be set up from an external OpenFlow controller in the JSON (JavaScript Object Notation:RFC4627) format.

In the cloud environment, it is becoming difficult to update the traffic control settings of the Unified Network Controller manually in conjunction with configuration change of the cloud server. Setting update of this equipment can be automated by creating a user program that updates the traffic control settings of this equipment in conjunction with configuration change of the cloud server by using the OpenFlow protocol on the OpenFlow controller on the cloud.



OpenFlow has proactive type and reactive type protocols. This equipment adopts the proactive type protocol.

- Proactive type
Settings such as scenarios and filters are set from the OpenFlow controller for this equipment in advance
- Reactive type
An inquiry is made to the OpenFlow controller by using a Packet-In message when an unknown packet is received

This equipment does not support changes of roles. It always operates as Equal without using the MASTER/SLAVE method. In OpenFlow, up to 2 controllers can be connected simultaneously.

16.2 OpenFlow Version

The OpenFlow version supported by this equipment is compliant with v1.3.4.

If a version other than v1.3 is used, connection is disconnected. Connection is allowed for v1.3, but only v1.3.4 with which this equipment is compliant is supported.

OpenFlow version	Remark
1.0	Not supported
1.1	Not supported
1.2	Not supported
1.3.0	Not supported
1.3.1	Not supported
1.3.2	Not supported
1.4.0	Not supported
1.3.3	Not supported
1.3.4	Supported
1.3.5	To be supported
1.4.1	To be supported
1.5.1	To be supported

16.3 Supported OpenFlow Messages

The OpenFlow messages supported by this equipment are shown below.

Message (type)	Use	Support
OFPT_HELLO (0)	Used to exchange version information, etc.	OK
OFPT_ERROR (1)	Reports an error.	OK
OFPT_ECHO_REQUEST (2)	Echo request	OK
OFPT_ECHO_REPLY (3)	Echo reply	OK
OFPT_EXPERIMENTER (4)	JSON-format input in the data section of the Experimenter message is supported. The setting command of this equipment can be executed.	OK
OFPT_FEATURES_REQUEST (5)	Used to exchange data path information, etc.	OK
OFPT_FEATURES_REPLY (6)	Used to exchange data path information, etc.	OK
OFPT_GET_CONFIG_REQUEST (7)	Requests configuration information of OpenFlow.	OK
OFPT_GET_CONFIG_REPLY (8)	Replies to a request for configuration information of OpenFlow.	OK
OFPT_SET_CONFIG (9)	Sets configuration of OpenFlow.	OK
OFPT_PACKET_IN (10)	—	N/A
OFPT_FLOW_REMOVED (11)	—	N/A
OFPT_PORT_STATUS (12)	—	N/A
OFPT_PACKET_OUT (13)	—	N/A
OFPT_FLOW_MOD (14)	Specify addition/deletion of the flow in the command section of the Flow Mod message and the flow match conditions in the Match section, and JSON-format input in the Experimenter data section of the action field is supported.	OK
OFPT_GROUP_MOD (15)	—	N/A
OFPT_PORT_MOD (16)	—	N/A
OFPT_TABLE_MOD (17)	—	N/A
OFPT_MULTIPART_REQUEST (18)	Requests to acquire various information. Also, when the Multipart type is OFPMP_EXPERIMENTER (0xffff), JSON-format input to the Experimenter data section is supported.	OK
OFPT_MULTIPART_REPLY (19)	Requests to acquire various information.	OK
OFPT_BARRIER_REQUEST (20)	Message execution completion notification request	OK
OFPT_BARRIER_REPLY (21)	Message execution completion notification reply	OK
OFPT_QUEUE_GET_CONFIG_REQUEST (22)	—	N/A
OFPT_QUEUE_GET_CONFIG_REPLY (23)	—	N/A
OFPT_ROLE_REQUEST (24)	OpenFlow controller role notification request	OK
OFPT_ROLE_REPLY (25)	OpenFlow controller role notification reply	OK

Message (type)	Use	Support
OFPT_GET_ASYNC_REQUEST (26)	—	N/A
OFPT_GET_ASYNC_REPLY (27)	—	N/A
OFPT_SET_ASYNC (28)	—	N/A
OFPT_METER_MOD (29)	—	N/A

* When an unsupported message is received, an error message (OFPT_ERROR) is sent.

For details of the messages, refer to "Appendix I Details of OpenFlow Message". For the supported OpenFlow messages that can be used to enter commands to this equipment, refer to "Appendix H Details of OpenFlow Message Supported for CLI Command".

16.4 OpenFlow Messages Supported for CLI Commands

The OpenFlow messages that can be used to enter commands to this equipment using the OpenFlow protocol are as follows.

Message	Use
OFPT_EXPERIMENTER	Enter JSON-format data in the data section of the OFPT_EXPERIMENTER message.
OFPT_MULTIPART_REQUEST	Enter JSON-format data in the data section if the multipart type of the OFPT_MULTIPART_REQUEST message is OFPMP_EXPERIMENTER.
OFPT_FLOW_MOD	Specify addition/deletion of the flow in the command section of the OFPT_FLOW_MOD message and the flow match conditions in the Match section, and enter JSON-format data in the Experimenter data section of the action field.

For details, refer to "Appendix H Details of OpenFlow Message Supported for CLI Command".

16.5 JSON Format

The OpenFlow function supporting the commands of this equipment uses JSON-format data. JSON is a data description language for expressing data. In the JSON description method, a parameter key and value are paired with a colon (":"). Multiple parameters are delimited with commas (","). The entire data is enclosed in curly brackets ("{" and "}").

Describe all keys and values as strings. Specify the key "command" that indicates the command type and the parameters of the CLI command. In OpenFlow, keys are described in no particular order. The order does not have to be the same as the order of parameters of the CLI command.

An example of the JSON description for scenario addition is shown below.

```
{
  "command": "add scenario "
  "scenario_name": "/port1/North",
  "action": "aggregate",
  "min_bandwidth": "5M",
  "peak_bandwidth": "8M",
  "class": "2",
  "bufsize": "512k"
}
```

For details of the JSON description method, refer to "Appendix E JSON Description Method".

16.6 Supported Command List

OpenFlow provides commands for the setting of scenarios, filters, and rule lists as well as for information acquisition. Each function is equivalent to that of the corresponding CLI command. This also applies to the parameters specified, range of value, and whether it can be omitted or not, etc. For details, refer to "Appendix H Details of OpenFlow Message Supported for CLI Command".

Target	Operation	Corresponding CLI command
Scenario	Add	add scenario
	Update	update scenario
	Delete	delete scenario
	Information acquisition	show scenario
	Counter acquisition	show scenario counter
Application acceleration	Add	add apl-accel
	Update	update apl-accel
	Delete	delete apl-accel
Filter	Add	add filter
	Delete	delete filter
	Information acquisition	show filter
Rule list	Group addition	add rulelist group
	Group deletion	delete rulelist group
	Entry addition	add rulelist entry
	Entry deletion	delete rulelist entry
	Information acquisition	show rulelist
Channel	Add	add channel
	Delete	delete channel
	Information acquisition	show channel
Channel interface	Set	set ip channel
	release	unset ip channel
	Information acquisition	show ip channel
Static path of channel interface	Add	add route
	Delete	delete route
	Information acquisition	show route target
Traffic acceleration bypass	Enable	set wan-accel bypass status
	Recovery time setting	set wan-accel bypass recoverytime
	Forced setting	switch wan-accel bypass force

16.7 Common Error Messages

If a setting is invalid even though the JSON format is used correctly, an error message is returned. The common error messages are as follows.

Error message	Description
Specified command is invalid.	The command is invalid. Check whether the specified JSON-format key and value are correct.
Required parameter is not specified.	A required parameter is not specified. Check whether the specified JSON-format key and value are correct.
Failed to create pipe.	An error occurred in PIPE creation for internal communication. Execute again after a while.
No response message from LR.	There is no response from the internal software. Execute again after a while.
System busy.	The system is busy. Execute again after a while.

16.8 Error Message List

The API-specific error messages are as follows.

API	Error message
Scenario addition	Specified Scenario Class is invalid. It must be either of 1,2,3,4,5,6,7,8. • The specified class is invalid.
	Specified scenario fail action class is invalid.It must be either of 1,2,3,4,5,6,7,8. • The specified Fail Action class is invalid.
	Specified Minimum Bandwidth is invalid. (Valid from 0, 10k to 10G) • The specified Minimum Bandwidth is invalid.
	Specified Peak Bandwidth is invalid. (Valid from 10k to 10G) • The specified Peak Bandwidth is invalid.
	Specified fail action minimum bandwidth is invalid. (Valid from 0, 10k to 10G) • The specified Fail Action Minimum Bandwidth is invalid.
	Specified fail action peak bandwidth is invalid. (Valid from 10k to 10G) • The specified Fail Action Peak Bandwidth is invalid.
	Peak Bandwidth should be greater than Minimum Bandwidth. • peak_bandwidth must be set to min_bandwidth or more.
	Specified Buff Size is invalid. (Valid from 2k to 1G) • The specified bufsize is invalid.
	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified Scenario Name is already used. • The specified scenario name has already been used in another scenario.
	Specified Scenario of upper level hierarchy is not found. • The scenario of upper level hierarchy does not exist.
	maximum number of scenario was exceeded. • The maximum number of registered scenarios was exceeded.
	Specified Scenario ID is invalid. (Valid from 1 to 40000) • The scenario index is out of range.
	Specified Scenario ID is already used. • The specified scenario index has already been used in another scenario.
	Specified Max Q Num is invalid. (Valid from 1 to 4096) • maxquenum is out of range.
	Specified Q Division Field is invalid. Valid fields: default, vlan, cos, inner-vlan, inner-cos, ethertype, sip, dip, tos, proto, sport, dport • The specified quedivision field is invalid.

API	Error message
Scenario addition (continued)	failaction is not specified. <ul style="list-style-type: none"> fail_min_bw, fail_peak_bw, fail_class cannot be set without specifying failaction.
	Specified Failaction is invalid. <ul style="list-style-type: none"> fail_min_bw, fail_peak_bw, and fail_class can only be set when forwardattribute is specified as failaction.
	Invalid IP address <ul style="list-style-type: none"> The format or value of the specified IP address is invalid.
	Peer IP version and second-peer IP version are different. <ul style="list-style-type: none"> The IP versions of peer and second-peer must match.
	Peer and second-peer are same IP_address. <ul style="list-style-type: none"> Different IP addresses must be set for peer and second-peer.
	Specified dport is invalid. (Valid from 10001 to 20000) <ul style="list-style-type: none"> The specified dport is invalid.
	Specified Dport is already used. <ul style="list-style-type: none"> The specified dport has already been used in another scenario.
	Specified vid is invalid. (Valid from 1 to 4094) <ul style="list-style-type: none"> The specified VLAN ID is invalid.
	Specified inner-vid is invalid. (Valid from 1 to 4094) <ul style="list-style-type: none"> The specified Inner-VLAN ID is invalid.
	VID must be specified when inner-VID is specified. <ul style="list-style-type: none"> Inner VLAN ID can be specified only when the VLAN ID is specified.
	Specified cos is invalid. (Valid from 0 to 7) <ul style="list-style-type: none"> The specified CoS value is invalid.
	Specified inner-cos is invalid. (Valid from 0 to 7) <ul style="list-style-type: none"> The specified CoS value is invalid.
	VID must be specified when CoS is specified. <ul style="list-style-type: none"> The CoS value can be specified only when the VLAN ID is specified.
	Inner-VID must be specified when inner-cos is specified. <ul style="list-style-type: none"> The Inner-CoS value can be specified only when the Inner VLAN ID is specified.
	Specified dscp is invalid. (Valid from 0 to 63) <ul style="list-style-type: none"> The specified DSCP value is invalid.
	Specified tcp-mem is invalid. (Valid from 64k to 200M) <ul style="list-style-type: none"> The specified TCP buffer size is invalid.
	Specified bypass threshold RTT is invalid. (Valid from 0 to 10000) <ul style="list-style-type: none"> The specified RTT threshold value of auto bypass of traffic acceleration is invalid.
	Specified peak bandwidth is not licensed. <ul style="list-style-type: none"> The specified bandwidth is not licensed.
	Data block size should be divided by fec block size. <ul style="list-style-type: none"> The data block size must be set to a value that is divisible by the FEC block size.
	Data block size should be greater than fec block size. <ul style="list-style-type: none"> The data block size must be set to a value that is greater than the FEC block size.

API	Error message
Scenario addition (continued)	Specified fec block size is invalid. (Valid from 2K to 50K) • The specified FEC block size is invalid.
	Specified data block size is invalid. (Valid from 2K to 200K) • The specified data block size is invalid.
	Specified fec session is invalid. (Valid from 0 to 1000) • The specified FEC session count is invalid.
	FEC function is not licensed. • The TCP-FEC function is not licensed.
	Maximum number of secondary peer was exceeded. • The maximum number of scenarios specified as second-peer was exceeded.
	Maximum number of keep alive scenario was exceeded. • The maximum number of scenarios with bypass-keep enabled was exceeded.
	Specified scenario has packets in buffer. Please wait until the buffer becomes empty, and try again. • The specified scenario is sending out a packet. Wait for a while until it finishes sending, and execute it again.

API	Error message
Scenario update	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified Scenario Class is invalid. It must be either of 1,2,3,4,5,6,7,8. • The specified class is invalid.
	Specified scenario fail action class is invalid. It must be either of 1,2,3,4,5,6,7,8. • The specified Fail Action class is invalid.
	Specified Minimum Bandwidth is invalid. (Valid from 0, 10k to 10G) • The specified Minimum Bandwidth is invalid.
	Specified fail action minimum bandwidth is invalid. (Valid from 0, 10k to 10G) • The specified Fail Action Minimum Bandwidth is invalid.
	Specified Peak Bandwidth is invalid. (Valid from 10k to 10G) • The specified Peak Bandwidth is invalid.
	Specified fail action peak bandwidth is invalid. (Valid from 10k to 10G) • The specified Fail Action Peak Bandwidth is invalid.
	Peak Bandwidth should be greater than Minimum Bandwidth. • peak_bandwidth must be set to min_bandwidth or more.
	Specified Buff Size is invalid. (Valid from 2k to 1G) • The specified bufsize is invalid.
	It is necessary to set one or more parameters. • One or more parameters must be set.
	Specified Scenario Mode is invalid. • The specified scenario mode is invalid.
	Specified Max Q Num is invalid. (Valid from 1 to 300000) • maxquenum is out of range.

API	Error message
Scenario update (continued)	Extended number of scenario is not licensed. <ul style="list-style-type: none"> • Scenarios cannot be registered over the limit of the scenario extension license. • maxquenum over the limit of the scenario extension license cannot be set.
	Specified Q Division Field is invalid. Valid fields: default, vlan, cos, inner-vlan, inner-cos, ethertype, sip, dip, tos, proto, sport, dport <ul style="list-style-type: none"> • The specified quedivision field is invalid.
	Fail action forward is incorrect. Specified Failaction is invalid. <ul style="list-style-type: none"> • fail_min_bw, fail_peak_bw, and fail_class can only be set when forwardattribute is specified as failaction.
	Invalid IP address <ul style="list-style-type: none"> • The format or value of the specified IP address is invalid.
	Specified cos is invalid. (Valid from 0 to 7) <ul style="list-style-type: none"> • The specified CoS value is invalid.
	Specified inner-cos is invalid. (Valid from 0 to 7) <ul style="list-style-type: none"> • The specified CoS value is invalid.
	Specified dscp is invalid. (Valid from 0 to 63) <ul style="list-style-type: none"> • The specified DSCP value is invalid.
	Specified tcp-mem is invalid. (Valid from 64k to 200M) <ul style="list-style-type: none"> • The specified TCP buffer size is invalid.
	Specified bypass threshold RTT is invalid. (Valid from 0 to 10000) <ul style="list-style-type: none"> • The specified RTT threshold value of auto bypass of traffic acceleration is invalid.
	Specified peak bandwidth is not licensed. <ul style="list-style-type: none"> • The specified bandwidth is not licensed.
	Data block size should be divided by fec block size. <ul style="list-style-type: none"> • The data block size must be set to a value that is divisible by the FEC block size.
	Data block size should be greater than fec block size. <ul style="list-style-type: none"> • The data block size must be set to a value that is greater than the FEC block size.
	Specified fec block size is invalid. (Valid from 2K to 50K) <ul style="list-style-type: none"> • The specified FEC block size is invalid.
	Specified data block size is invalid. (Valid from 2K to 200K) <ul style="list-style-type: none"> • The specified data block size is invalid.
	Specified fec session is invalid. (Valid from 0 to 1000) <ul style="list-style-type: none"> • The specified FEC session count is invalid.
	FEC function is not licensed. <ul style="list-style-type: none"> • The TCP-FEC function is not licensed.
	Maximum number of keep alive scenario was exceeded. <ul style="list-style-type: none"> • The maximum number of scenarios with bypass-keep enabled was exceeded.

API	Error message
Scenario deletion	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Down level hierarchy scenario exists. • The scenario of lower level hierarchy exists.

API	Error message
Scenario information acquisition	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.

API	Error message
Scenario counter information acquisition	Specified scenario name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Next scenario is not exist. • The next scenario does not exist.

API	Error message
Add application acceleration	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified scenario name is not wan-accel mode. • The specified scenario is not the acceleration mode.
	Specified Scenario Name is already used. • The SMB acceleration has already been set in the specified scenario.
	Specified tcp port is invalid.(Valid from 0 to 65535) (Up to 16 ports can be specified with separated comma without space) • The specified SMB TCP Port is invalid.
	Specified smb session is invalid. (Valid from 0 to 1000) • The specified SMB Session is invalid.
	Specified read cache size is invalid. (Valid from 64k to 60M) • The specified Read Cache Size is invalid.

API	Error message
Update application acceleration	Specified channel name is invalid. • The specified channel name is invalid.
	Channel name already exists. • The specified channel name has already been used in another channel.
	Slot #N is invalid. • The specified slot is invalid.
	Specified tcp port is invalid.(Valid from 0 to 65535) (Up to 16 ports can be specified with separated comma without space) • The specified SMB TCP Port is invalid.
	Specified smb session is invalid. (Valid from 0 to 1000) • The specified SMB Session is invalid.
	Specified read cache size is invalid. (Valid from 64k to 60M) • The specified Read Cache Size is invalid.

API	Error message
Delete application acceleration	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified scenario name is not wan-accel mode. • The specified scenario is not the acceleration mode.
	Specified protocol is already disabled. • The specified protocol is already invalid.

API	Error message
Filter addition	Specified Scenario Name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified filter Name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid Filter Name length is from 1 to 48.) • The specified filter name is invalid.
	Specified filter Name is already used. • The specified filter name has already been used in another filter.
	Specified Ether type is invalid. (Valid from 0x0000 to 0xFFFF) • The specified Ether type is invalid.
	Specified vid is invalid. (Valid from 0 to 4094, Or Start - End) • The specified VLAN ID is invalid.
	Specified cos is invalid. (Valid from 0 to 7, Or Start - End) • The specified CoS value is invalid.
	Specified inner-vid is invalid. (Valid from 0 to 4094, Or Start - End) • The specified VLAN ID is invalid.
	Specified inner-cos is invalid. (Valid from 0 to 7, Or Start - End) • The specified CoS value is invalid.

API	Error message
Filter addition (continued)	The format or value of the specified source IP address is invalid. • The specified Source IP address is invalid.
	The format or value of the specified destination IP address is invalid. • The specified Destination IP address is invalid.
	The format or value of the specified source IPv6 address is invalid. • The specified Source IPv6 address is invalid.
	The format or value of the specified destination IPv6 address is invalid. • The specified Destination IPv6 address is invalid.
	The format or value of the specified source IPv6 address is invalid. • The specified Source IPv6 address is invalid.
	Specified rulelist name of source IP address is invalid. Specified rulelist name of destination IP address is invalid. Specified rulelist name of source port is invalid. Specified rulelist name of destination port is invalid. • The rule list name is invalid. The format or value of the specified destination IPv6 address is invalid. • The specified Destination IPv6 address is invalid.
	Specified rulelist name of source IP address is not used. Specified rulelist name of destination IP address is not used. Specified rulelist name of source port is not used. Specified rulelist name of destination port is not used. • The specified rule list does not exist. Specified rulelist name of source IP address is invalid. Specified rulelist name of destination IP address is invalid. Specified rulelist name of source port is invalid. Specified rulelist name of destination port is invalid. • The rule list name is invalid.
	IP Filter and rulelist of source IP address is not same type. IP Filter and rulelist of destination IP address is not same type. IP Filter and rulelist of source port is not same type. IP Filter and rulelist of destination port is not same type. • The type is different from the target rule list. Specified rulelist name of source IP address is not used. Specified rulelist name of destination IP address is not used. Specified rulelist name of source port is not used. Specified rulelist name of destination port is not used. • The specified rule list does not exist.
	Specified ToS is invalid. (Valid from 0 to 255, Or Start - End) • The specified ToS value or Traffic Class value is invalid.
	Specified protocol number is invalid. (Valid from 0 to 255, Start - End, Or tcp/udp/icmp) • The specified protocol number is invalid.
	Specified Source TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) • The specified sport number is invalid.
	Specified Destination TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) • The specified dport number is invalid.
	Specified Filter Priority is invalid. (Valid from 1 to 40000) • The specified filter priority is invalid.

API	Error message
Filter addition (continued)	maximum number of filter was exceeded. • The maximum number of registered filters was exceeded.
	It is necessary to set one or more parameters other than Priority. • For the Ethernet filter, at least one parameter other than Priority must be set.
	Could not Add the Filter. • The filter cannot be registered.

API	Error message
Filter deletion	Specified scenario name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified filter name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid Filter Name length is from 1 to 48.) • The specified filter name is invalid.
	Specified filter name is not used. • The specified filter does not exist.

API	Error message
Filter information acquisition	Specified scenario name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Specified filter name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid Filter Name length is from 1 to 48.) • The specified filter name is invalid.
	Specified filter name is not used. • The specified filter does not exist.

API	Error message
Rule list group addition	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid rulename length is from 1 to 32.) • The rule list name is invalid.
	Specified rulelist name is already in use. • A rule list with the same name already exists.
	Maximum number of rulelist was exceeded. • The maximum number of registered rule lists was exceeded.

API	Error message
Rule list group deletion	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid rulename length is from 1 to 32.) • The rule list name is invalid.
	Specified rulelist name is not used. • The specified rule list does not exist.
	Rulelist is used by filter. • A rule list is set to the filter.

API	Error message
Rule list entry addition	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid rulename length is from 1 to 32.) • The rule list name is invalid.
	Specified rulelist name is not used. • The specified rule list does not exist.
	The format or value of the specified IP address is invalid. • The specified IP address is invalid.
	Specified TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) • The specified TCP/UDP port number is invalid.
	Maximum number of rulelist entry was exceeded. • The maximum number of registered rule list entries (512) for the specified rule list was exceeded.
	Maximum number of total rulelist entry was exceeded. • The maximum number of registered rule list entries (64000) for all rule lists was exceeded.
	Specified rulelist entry is already in use. • The specified rule list entry has been registered.
	Rulelist entry and rulelist is not same type. • The type is different from the target rule list.

API	Error message
Rule list entry deletion	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid rulename length is from 1 to 32.) • The rule list name is invalid.
	Specified rulelist name is not used. • The specified rule list does not exist.
	The format or value of the specified IP address is invalid. • The specified IP address is invalid.
	Specified TCP/UDP port number is invalid. (Valid from 0 to 65535. Or Start - End) • The specified TCP/UDP port number is invalid.
	Rulelist entry and rulelist is not same type. • The type is different from the target rule list.
	Specified rulelist entry is not used. • The specified rule list entry does not exist.

API	Error message
Rule list information acquisition	Specified rulelist name is invalid. (Number only cannot be specified. "all" cannot be specified.) (Valid rulename length is from 1 to 32.) • The rule list name is invalid.
	Specified rulelist name is not used. • The specified rule list does not exist.

API	Error message
Channel addition	Specified channel name is invalid. • The specified channel name is invalid.
	Channel name already exists. • The specified channel name has already been used in another channel.
	Slot #N is invalid. • The specified slot is invalid.
	Port <slot/port> is invalid. • The specified port is invalid.
	Specified group name is invalid. • The specified group name is invalid.
	Specified group name is not used. • The specified group does not exist.
	Specified vid is invalid. (Valid from 1 to 4094) • The specified VLAN ID is invalid.
	Specified TPID is invalid. (Valid 0x8100,0x88a8,0x9100,0x9200 or 0x9300.) • The specified TPID is invalid.
	Specified inner-vid is invalid. (Valid from 1 to 4094) • The specified Inner-VLAN ID is invalid.
	VID must be specified when inner-VID is specified. • Inner VLAN ID can be specified only when the VLAN ID is specified.
	Specified mtu is invalid. (Valid from 300 to 10200) • The specified mtu is invalid.
	Specified vid and inner-vid is already used on channel "channel name". • The specified vid and inner-vid have been used in the "channel name" channel.
	Specified port is already used on other default-channel. • The specified port has already been used in another default channel.
	Maximum number of channel was exceeded. • The maximum number of registered channels was exceeded.

API	Error message
Channel deletion	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.

API	Error message
Channel information acquisition	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.

API	Error message
Interface setting	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.
	Invalid IP address • The format or value of the specified IP address is invalid.
	Invalid netmask • The format or value of the specified subnet mask is invalid.
	Default-channel cannot be set for this command. • The default channel cannot be set.

API	Error message
Interface release	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.
	Cannot specified "ipv4" or "ipv6". • "IPv4" or "IPv6" cannot be specified for "all".

API	Error message
Interface information display	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.
	IP interface is not configured. • No IP address is set for the specified channel or next channel.

API	Error message
Static path addition	Route entry already exists. • The rule entry already exists.
	Invalid IP address • The format or value of the specified IP address is invalid.
	Invalid netmask • The format or value of the specified subnet mask is invalid. • The specified prefix length value is invalid.
	Invalid gateway • The format or value of the gateway IP address is invalid.
	Default-channel cannot be set for this command. • The default channel cannot be specified.
	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.
	Target IP address and gateway is not same IP version. • The versions of the destination IP address and gateway IP address do not match.
	Maximum number of route was exceeded. • The maximum number of registered static paths was exceeded.

API	Error message
Static path deletion	Invalid IP address • The format or value of the specified IP address is invalid.
	Invalid netmask • The format or value of the specified subnet mask is invalid.
	Invalid gateway • The format or value of the gateway IP address is invalid.
	Route info is not found. • The specified static path does not exist.
	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.
	Target IP address and gateway is not same IP version. • The versions of the destination IP address and gateway IP address do not match.

API	Error message
Static path information display	Specified channel name is invalid. • The specified channel name is invalid.
	Specified channel name is not used. • The specified channel does not exist.

API	Error message
Traffic acceleration bypass enable	None

API	Error message
Traffic acceleration bypass recovery time settings	Duration is valid from 1 to 600 • The bypass recovery time is out of range.

API	Error message
Traffic acceleration forced bypass enable	Specified scenario name is invalid. • The specified scenario name is invalid.
	Specified scenario name is not used. • The specified scenario does not exist.
	Scenario type is different. Please specify a wan-accel scenario. • The specified scenario is not an acceleration mode scenario.

Chapter 17 Network Bypass Function

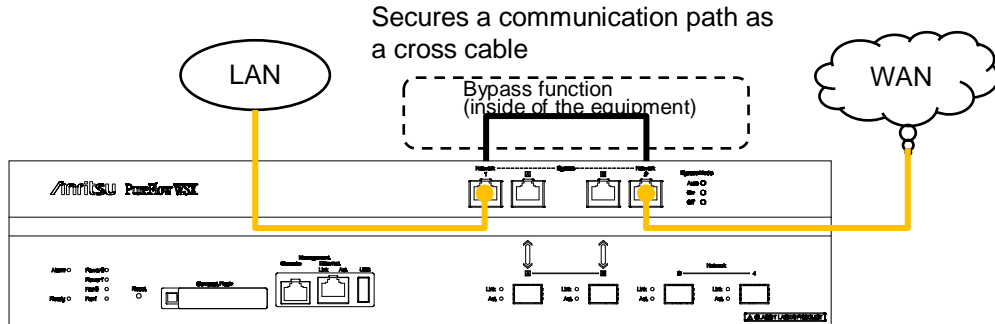
This chapter describes the network bypass function and setting.

17.1	Overview	17-2
17.2	Setting and Checking the Function.....	17-3
17.3	Precautions.....	17-5

17.1 Overview

NF7603A, NF7604A, and NF7605A have the Network port bypass function.

This function can secure a communication path by bypassing the Network port when an equipment error occurs.



If the Network port is in the bypass status, this equipment is disconnected from the network, and the traffic control does not work. Because the equipment operates as a cross cable in the bypass state, it functions as if the opposing devices are directly connected.

The connection ports of the opposing devices temporarily enter the link-down state when it changes to the bypass state. A link is established again between the opposing devices to restart communication.

17.2 Setting and Checking the Function

While this equipment is running, automatic or desired bypass operations can be performed.

The following commands are used for bypass operations.

set bypass {auto on off}	Sets the control mode of network bypass function. If auto is specified, the auto bypass control will be enabled during the detection of an equipment error. If on is specified, the equipment will forcibly be placed in the bypass state. If off is specified, the equipment will forcibly be placed in the non-bypass state. The default value is "auto".
bypass time <time> {on off}	Switches the network bypass temporarily. If on is specified, the equipment is forcibly switched to the bypass state and after a lapse of time seconds, is returned to the previous state automatically. If off is specified, the equipment is forcibly switched to the non-bypass state and after a lapse of time seconds, is returned to the previous state automatically. Executing this command displays the current time and the expiration time of the timer. Note: This command cannot be saved using the save config command.
show bypass	Displays the network bypass function settings and state.

To forcefully switch the Network port to the bypass state, execute the following command.

```
PureFlow(A)> set bypass on
PureFlow(A)>
```

To switch the Network port to the bypass state temporarily for 300 seconds, execute the following command.

```
PureFlow(A)> bypass time 300 on
Current time : Feb 29 17:38:47
Expiring time: Feb 29 17:43:47
PureFlow(A)>
```

To return it to the previous state before execution without waiting for 300 seconds, perform setting again with a shorter time (e.g. 1 second).

To check the settings set with the setting command or the current bypass state of the Network port, use the "show bypass" command.

```
PureFlow(A)> show bypass
Control mode      : auto
Bypass state      : off
Timer remaining   : 12[s]
PureFlow(A)>
```

If auto bypass control is enabled, stopping of the network can be avoided when an equipment error occurs. If "auto" is specified in the "set bypass" command, the Network port enters the bypass state in any of the following timings.

- When the startup of the equipment is complete
It enters the bypass state when an error occurs during the startup of the Forwarding CPU. It enters the non-bypass state when it starts up normally.
- When an equipment error is detected
It enters the bypass state when an error of the Forwarding CPU is detected or a severe error such as a core stop error occurs in the Control CPU.
- When the "reboot system" command is executed
It enters the bypass state before rebooting.
- When the Reset button is pressed
It enters the bypass state when rebooting starts.
- When the power is turned off
It enters the bypass state when the power is shut down.

Note:

Auto bypass control with the "set bypass auto" command operates only when any of the above conditions occurs.

Even if this command is executed, the bypass state does not change unless any of the above conditions occurs.

When operating with auto setting after performing bypass operation with the command, use the "set bypass off" command to return it to the non-bypass state, and then execute the "set bypass auto" command and start operation. It doesn't enter the non-bypass state automatically even when the "set bypass auto" command is executed in the bypass state.

Bypass operations are not recorded to syslog when a severe error occurs in the Control CPU. All other bypass operations are recorded to syslog. Syslog messages to be recorded are as follows.

To find the cause when auto bypass control operates, refer to the message recorded immediately before the following syslog messages.

- Changed to the bypass state
Bypass state was changed to on
- Changed to the non-bypass state
Bypass state was changed to off

17.3 Precautions

When using the network bypass function, note the following.

In the bypass state, this equipment operates as a cross cable. Select the proper type (cross/straight) and length of the cable connecting this equipment and opposing device by referring to "PureFlow WSX Unified Network Controller NF7600 Series Operation Manual" so that communication is available whether it is in the bypass state or non-bypass state.

During network bypass operation, the ports of the opposing devices temporarily enter the link-down state, and a link is established again after several seconds.

The time until a link is established again varies depending on the characteristics of the connected device. It is recommended to check them before actual operation.

In the bypass state, the Network port of this equipment enters the link-down state and the Link LED goes off. As a result, a link change trap and link change syslog of SNMP are sent during network bypass operation. A link change may not be detected in bypass operation when an equipment error is detected.

(Blank page)

Chapter 18 Top Counter

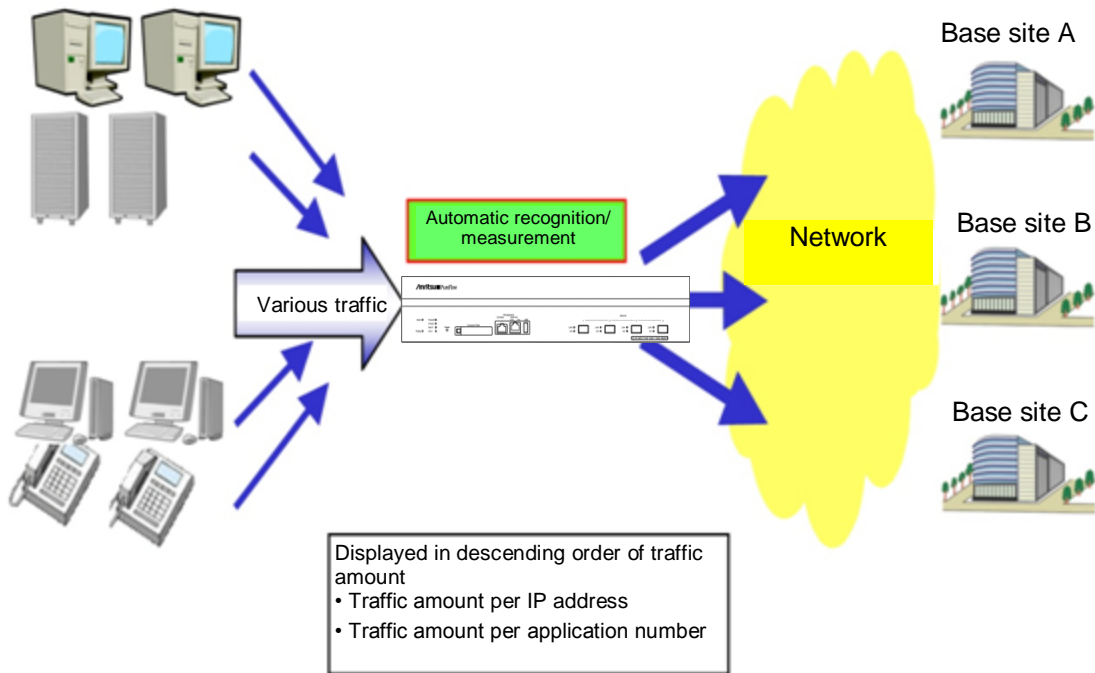
This chapter describes the top counter feature.

18.1	Overview	18-2
18.2	Display Unit of the Top Counter.....	18-2
18.3	Measurement Range of the Top Counter.....	18-3
18.4	Traffic Counter	18-4
18.5	Measuring Traffic at Specific Application Ports	18-5
18.6	Operation Command List	18-5
18.7	Operation Procedure.....	18-6
18.8	Operation Example	18-7
18.9	Cautions	18-9

18.1 Overview

The top counter feature helps you to understand the usage status of traffic. This feature automatically recognizes traffic volume and measures the flow for each IP address or application port number, and displays the top 25 traffic volumes in descending order.

Also, Monitoring Manager 2 allows you to view the usage state in real time on graphs and create a report including past data. For details, see the Monitoring Manager 2 Operation Manual.



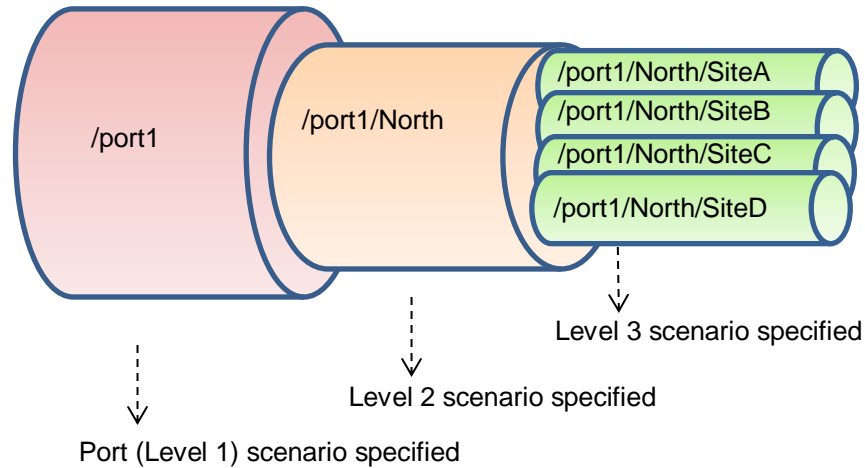
18.2 Display Unit of the Top Counter

The top counter feature measures traffic in the following 4 display units and displays the top 25 traffic volumes for each of the display units.

- Source IP address (SIP)
- Destination IP address (DIP)
- Combination of source IP address and destination IP address (SIP_DIP)
- Application port number (APPLI)

18.3 Measurement Range of the Top Counter

The top counter feature can specify the range of measurement of the top counter from all the traffic passing through this device. Up to 200 scenarios can be specified as the measurement range.



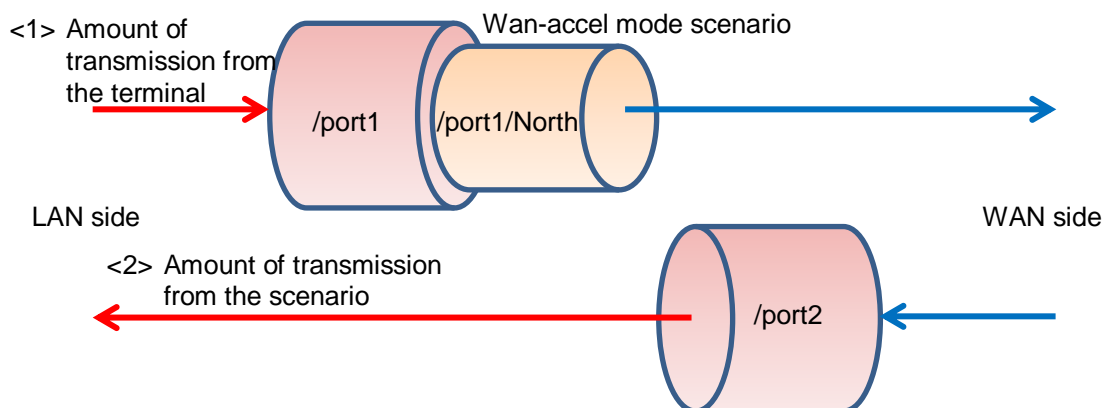
For example, to observe the traffic that consumes the most communication bandwidth in the traffic passing through the level n scenario, specify the level n scenario as the measurement range. This allows you to grasp the traffic with the largest amount of transmission in the traffic input to the scenario.

Pay attention to the following items when using the acceleration mode.

- (1) Traffic that passes through the acceleration mode scenario and is to be accelerated

<1> This allows you to grasp the traffic with the largest amount of transmission from the terminal among the traffic input from the network on the LAN side into the acceleration mode scenario.

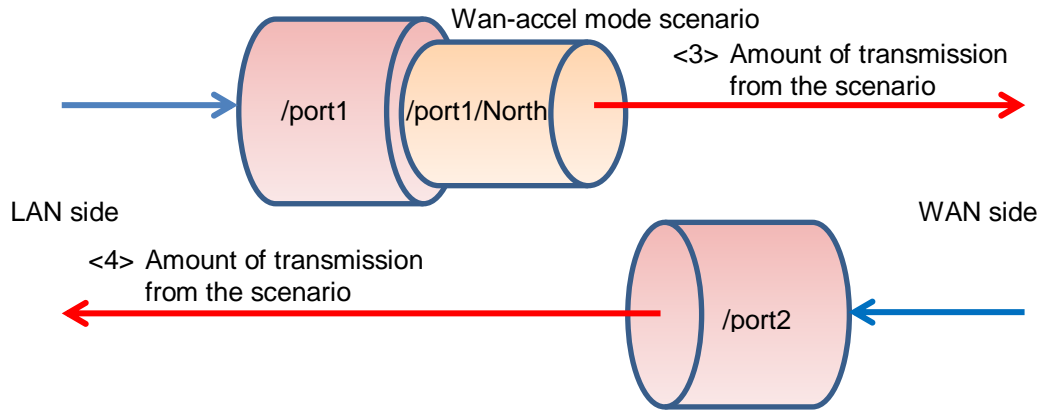
<2> For the traffic to be accelerated and output to the network on the LAN side, the acceleration mode scenario is not grasped but the port scenario or the traffic with the largest amount of transmission in the traffic passing through the level n scenario is grasped.



(2) Traffic that passes through the acceleration mode scenario but is not to be accelerated

<3> This allows you to grasp the traffic with the largest amount of transmission from the scenario among the traffic input from the network on the LAN side into the acceleration mode scenario.

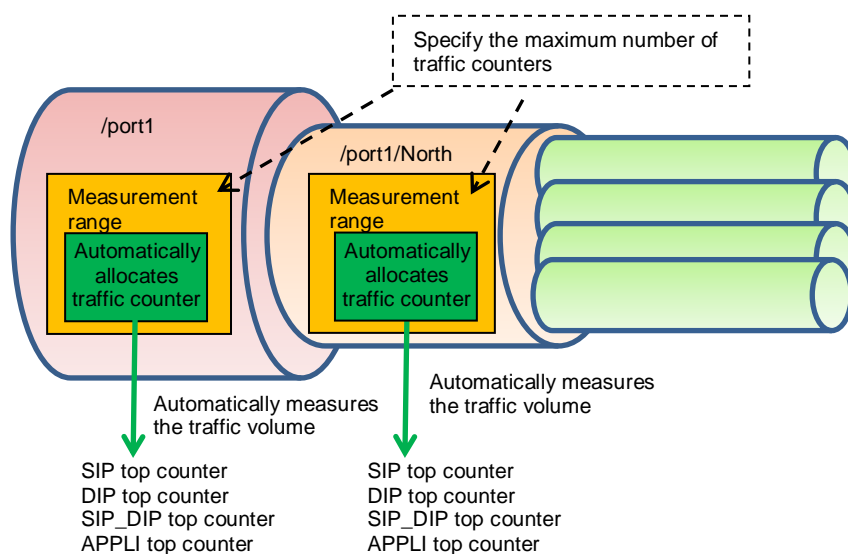
<4> For the traffic to be accelerated and output to the network on the LAN side, the acceleration mode scenario is not grasped but the port scenario or the traffic with the largest amount of transmission in the traffic passing through the level n scenario is grasped.



18.4 Traffic Counter

A traffic counter is automatically allocated to traffic that is automatically recognized such as by IP address or by application port number to measure the transmission traffic volume.

To use the top counter feature, you need to specify the maximum value of available traffic counters for each measurement range in advance. The total number of traffic counters is up to 1000000 for all measurement targets.



18.5 Measuring Traffic at Specific Application Ports

The top counter feature measures traffic volume by allocating traffic counters only to the specified application port number. Well-known applications are registered and measured by default. To check the number of the application port at which measurement will be performed by default, use the “show topcounter config all” command.

You can also measure traffic at an application port specified by you. Add the number of the application port to be measured by using the “add topcounter config appli port” command.

When measuring traffic at specific application ports, an application can be specified to be always monitored. When always monitor is specified, the traffic counter for the relevant application port number is always used. The “show topcounter target” command always shows the traffic from this port in the measurement results even if it is not within the top 25 rankings. An application port to be always monitored can be registered for each measurement range (scenario) by using the “add topcounter config appli port static” command.

18.6 Operation Command List

To operate the top counter feature, use the following commands:

set topcounter	Enables and disables the top counter.
set topcounter config interval time	Sets the collection cycle of the top counter.
add topcounter target	Adds a top counter measurement range.
update topcounter target	Changes the parameters specified for the top counter measurement range.
delete topcounter target	Deletes a top counter measurement range.
show topcounter config	Displays the top counter settings.
show topcounter target	Displays the top counter.
add topcounter config appli port	Adds the number of the application port whose top counter is to be measured.
delete topcounter config appli port	Deletes the number of the application port whose top counter is to be measured.
add topcounter config appli port static	Registers the number of an application port to always be monitored.
delete topcounter config appli port static	Deletes the number of an application port to always be monitored.

18.7 Operation Procedure

The procedure for operating the top counter function is described below.

- (1) Set the measurement range of the top counter.
Use the “add topcounter target” command to specify the traffic whose top counter is to be measured. Any scenario traffic can be specified as the measurement range.
- (2) Set the collection cycle of the top counter as required.
Use the “set topcounter config interval time” command to change the collection cycle of the top counter. If Monitoring Manager 2 is connected, the collection cycle may be changed (see Section 18.9 Cautions (2)). You can check the operating collection cycle with the “show topcounter config” command.
- (3) Add the number of an application port whose top counter is to be measured as required.
To measure an application port other than the default, use the “add topcounter config appli port” command to add a port number. You can check the default port number by using the “show topcounter config all” command.
- (4) Register the number of an application port to always be monitored as required.
Use the “add topcounter config appli port static” command to register the number of an application port to always be monitored. Registration of the number of an application port to be always monitored should be done for each measurement range (scenario).
- (5) Enable collection of the top counter.
Use the “set topcounter enable” command to enable the top counter feature. The top counter is displayed after the top counter feature is enabled and the collection cycle elapses.
- (6) Display the top counter.
Use the “show topcounter target” command to display the top counter. You can display the top counter by source IP address, by destination IP address, by the combination of source IP address and destination IP address, or by application port number.

18.8 Operation Example

An example of the command settings required to use the top counter feature is shown in the table below.

User setting item	Setting	Notes
Measurement range	Network port 1/1 /port1	Set a traffic counter count.
	Level 2 scenario /port1/North	Default setting of traffic counter count
	Level 3 scenario /port1/North/SiteA	Default setting of traffic counter count
Collection cycle	5 minutes	Note that if Monitoring Manager 2 is connected, the collection cycle may be changed (see Section 18.9 Cautions (2)).
Application port number	Add the number of an application port to be measured. 10000 20000 to 20003	In addition to the default application port number, measure application port numbers 10000, 20000, 20001, 20002, and 20003.
	Register the number of an application port to always be monitored. Scenario /port1 Port number 80	Always monitor the HTTP (port number 80) traffic.

The setting commands are as follows:

```
PureFlow(A)> add topcounter target scenario /port1 sip 10000 dip 10000 sip_dip 10000 appli 250
PureFlow(A)> add topcounter target scenario /port1/North
PureFlow(A)> add topcounter target scenario /port1/North/SiteA
PureFlow(A)> set topcounter config interval time 5
PureFlow(A)> add topcounter config appli port 10000
PureFlow(A)> add topcounter config appli port 20000-20003
PureFlow(A)> add topcounter config appli port static /port1 80
PureFlow(A)> set topcounter enable
PureFlow(A)>
```

The top counter is displayed as follows:

```
PureFlow(A)> show topcounter target scenario /port1 group sip
From      : 2013 Jan 02 19:47:55 To      : 2013 Jan 02 19:57:55
Total Octet: 1475806000 Total Packet: 1475806
```

Order	IP Address	Tx Octet	Tx Packet
1	192.168.101.121	8214	111
2	192.168.101.122	5846	79
3	fe80:0000:0000:0000:0290:ccff:fe22:8b4c	5772	78
4	fe80:0000:0000:0000:0290:ccff:fe22:8b4d	5698	77
5	fe80:0000:0000:0000:0290:ccff:fe22:8b4e	3848	52

PureFlow(A)>

```
PureFlow(A)> show topcounter target scenario /port1 group appli
From      : 2013 Jan 02 19:47:55 To      : 2013 Jan 02 19:57:55
Total Octet: 1475806000 Total Packet: 1475806
```

Order	TCP/UDP Port	Type	Tx Octet	Tx Packet
1	10.000		22625	276
2	20.000		1288	46
3	20.001		446	12
4	20.002		446	12
5	20.003		240	20
6	80	static	0	0

PureFlow(A)>

18.9 Cautions

- (1) The correct top counter may not be displayed if there are insufficient traffic counters. If the number of allocated traffic counters is greater than the number of communication nodes actually communicating, there may not be sufficient traffic counters. Communication nodes to which no traffic counter is allocated cannot be displayed in the top counter because the individual flow cannot be measured.
- (2) When Monitoring Manager 2 is used, the top counter may be aggregated in a different cycle than the collection cycle specified by CLI.
When Monitoring Manager 2 is connected to this device, the collection cycle of the top counter may be changed by Monitoring Manager 2. The collection cycle specified by CLI and the collection cycle set by the GUI of Monitoring Manager 2 are compared, and the top counter is collected at the longer cycle. To check the operating collection cycle, use the “show topcounter config” command.
- (3) If both the source port number and destination port number in the received TCP/IP packet are registered as the numbers of the application ports whose top counter is to be measured, the packet will be counted by the traffic counter of the destination port number. It is not counted by the traffic counter of the source port number.
- (4) You can add numbers of application ports whose top counter is to be measured as required, but you cannot delete the application port numbers set by default.
- (5) When the collection cycle of the top counter is changed from CLI or Monitoring Manager 2, the top counter aggregated in a shorter period of time than the specified collection cycle may be displayed only once. This is the result of the top counter from the time when the previous collection cycle was reached to the time when the collection cycle was changed.
- (6) The top counter is updated about 1 minute after the collection cycle of the top counter is reached.
- (7) When the collection cycle of the top counter is set to 1 minute, the total number of traffic counters is limited to 100,000 for all measurement targets.

(Blank page)

Appendix A Default Values

This device provides many setting items for different features. Some items require no setting unless the feature is used, and other items require setting. For the items that require a setting value, a default value is preset. Table 1 lists the setting items and setting values. For details of the commands, See “PureFlow WSX Unified Network Controller NF7600 series Command Reference TCP Acceleration Edition”.

Table 1 Default value list

Setting item	Command	Default value	Setting range
User Name	User Name	root	No setting
Prompt	set prompt	PureFlow	Up to 15 characters
Baud rate	set console baudrate	9600 bps	9600/19200/38400/ 115200 bps
Pager	set pager	enable	enable/disable
Auto logout	set autologout time	10 minutes	1 to 30 minutes
Password	set password	(None)	Up to 16 characters
	set adminpassword	(None)	Up to 16 characters
Network port setting (1000BASE-TSFP only)	set port autonegotiation	enable	enable/disable
	set port speed	1G	1G/100M/10M
	set port duplex	full	full/half
Flow control	set port flow_control	auto	auto Receive Pause on/off Send Pause on/off
Maximum frame length	set port mtu	2048	2048/10240
Ethernet port setting	set port autonegotiation system	enable	enable/disable
	set port speed system	1G	1G/100M/10M
	set port duplex system	full	full/half
SYSLOG	set syslog host	disable	enable/disable
	add syslog host (IP Address)	(None)	IP Address
	add syslog host (UDP port)	514	1 to 65534
	set syslog severity	notice(5)	0 to 6
	set syslog facility ccpu	16(local0)	0 to 23
	set syslog facility fcpu	17(local1)	0 to 23

Setting item	Command	Default value	Setting range
SNMP	set snmp syscontact	Not Yet Set	Up to 200 characters
	set snmp syslocation	Not Yet Set	Up to 200 characters
	set snmp sysname	Not Yet Set	Up to 200 characters
	set snmp traps	Enable for all	Enable/disable per trap
	add snmp view	(None)	view record name OID included/excluded
	add snmp community	(None)	Community name Version View name ReadOnly/ReadWrite
	add snmp group	(None)	Group name Authentication method ReadView WriteView NotifyView
	add snmp user	(None)	User name Group name Authentication method Password
	add snmp host	(None)	IPv4 address Version Authentication method User name/community name Trap/Inform UDP port number Transmission notification
TimeZone	set timezone	UTC +09:00	Offset from UTC
Summer time	set summertime	(None)	Start time Finish time Offset
SNTP	set sntp	disable	enable/disable
	set sntp server	(None)	IP Address
	set sntp interval	3600 seconds	60 to 86400 sec
RADIUS	set radius auth	disable	enable/disable
	set radius auth timeout	5	1 to 30 sec
	set radius auth retransmit	3	0 to 10 times
	set radius auth method	CHAP	CHAP/PAP
RADIUS server	add radius auth server	(None)	IP address Port number Common key Primary

Setting item	Command	Default value	Setting range
System interface	set ip system(IPv4 Address)	192.168.1.1	IPv4 Address
	set ip system(IPv4 netmask)	255.255.255.0	IPv4 Address
	set ip system(IPv4 up/down)	up	up/down
	set ip system(IPv6 Address)	::192.168.1.1	IPv6 Address
	set ip system(IPv6 prefixlen)	64	0 to 128
	set ip system(IPv6 up/down)	up	up/down
	set ip system gateway(IPv4)	(None)	IPv4 Address
	set ip system gateway(IPv6)	(None)	IPv6 Address
System interface filter	add ip system filter	(None)	Filter Index sip, dip, tos, proto, sport, dport permit/deny
Auto reboot	set autoreboot	enable	enable/disable
Flow aging time	set agingtime	300 seconds	1 to 1800 sec
Communication gap mode setting	set bandwidth mode	gap	gap/no_gap
Peak burst size	set shaper peak burst size	9216Byte	1536~9216Byte
Port group	add port group	(None)	Group name Port number
Scenario tree mode	set scenario tree mode	inbound	inbound/outbound
Traffic acceleration bypass	set wan-accel bypass status	enable	enable/disable
	set wan-accel bypass recoverytime	60 seconds	1 to 600 seconds
Link-down transfer feature	set lpt	disable	enable/disable
	add lpt pair port	(None)	Port number
Telnet connection setting	set telnet	enable	enable/disable
SSH connection setting	set ssh	enable	enable/disable
HTTP protocol	set http protocol	normalhttp	normalhttp /httpsecure
Network bypass setting	set bypass	Auto	auto/on/off
Top Counter	set topcounter	disable	enable/disable
	set topcounter config interval time	5 minutes	1 / 5 / 60 / 180 / 1440 minutes

(Blank page)

Appendix B syslog Messages

Table 2 lists the syslog messages. The items in this table are sorted by severity (the value in parentheses indicates the severity).

For reference:

Some syslog messages have a hexadecimal number in brackets ([] or < >) added. The hexadecimal number in the brackets indicates the location in the source code or the variable value, which Anritsu Networks will use for troubleshooting.

Table 2 syslog list

Severity	Syslog message	Occurs when	Action
Emergency (0)	Temperature #N of the system is critical : xx.xx	System temperature is in the dangerous range. (#N is 1 to 5) (xx.xx is temperature (°C))	Continued use may damage the hardware. Turn off the power immediately.
Alert (1)	Temperature #N of the system is OK : xx.xx	System temperature range returned to a normal value. (#N is 1 to 5) (xx.xx is temperature (°C))	No recovery measure is required.
	Temperature #N of the system is abnormal : xx.xx	System temperature is abnormal. (#N is 1 to 5) (xx.xx is temperature (°C))	Check that the temperature in the installation environment is in the range of 0 to 40°C. If it is within this range, replace the device. Otherwise, change the installation location.
	Power #N inserted	Power supply is inserted. (#N is 0 or 1)	No recovery measure is required.
	Power #N removed	Power supply is removed. (#N is 0 or 1)	No recovery measure is required.
	Power #N failed	Power supply failure is detected. (#N is 0 or 1)	Check the following. • If the power cable is connected • If the supply voltage is within the valid range (AC 100 V to AC 127 V / AC 200 V to AC 240 V) • If the power supply fan is working
	Power #N OK	Power supply failure is recovered. (#N is 0 or 1)	No recovery measure is required.
	Fan #N inserted	Fan unit is inserted. (#N is 0 or 1)	No recovery measure is required.

Severity	Syslog message	Occurs when	Action
Alert (1) (Continued)	Fan #N removed	Fan unit is removed. (#N is 0 or 1)	No recovery measure is required.
	Fan #N failed	Fan unit failure is detected. (#N is 0 or 1)	Check the following. • If the fan is working
	Fan #N OK	Fan unit failure is recovered. (#N is 0 or 1)	No recovery measure is required.
	No response from Slot #N	No response from the module (#N is 1)	Contact Anritsu Networks support.
	Slot #N response is OK	Response from the module recovered (#N is 1)	No recovery measure is required.
	System Buffer %s almost full	The usage of System Buffer %s exceeded 90%.	Check the traffic state and various settings.
	System Buffer %s recovered	The usage of System Buffer %s exceeded 90% and then dropped below 50%.	No recovery measure is required.
	TCP WARP Engine Buffer #N almost full	The usage of the TCP WARP Engine Buffer exceeded 90%. (#N is 1 to 100.)	Check the traffic state and various settings.
	TCP WARP Engine Buffer #N recovered	The usage of the TCP WARP Engine Buffer exceeded 90%, and then dropped below 50%. (#N is 1 to 100.)	No recovery measure is required.
	Critical error on FCPU Core[#N], Code[#M] Data1[0xxxxxxxx] Data2[0xxxxxxxx]	FCPU core failed and stopped.	Contact Anritsu Networks support.
	Queue blocktime exceeded. [S:#M Q:#Q]	Stop of packet transmission of Queue Q generated in Scenario M was detected.	Contact Anritsu Networks support.
Detected FCPU IIC error on port[#N/#M]	FCPU IIC interface failed. (#N is 1) (#M is 1 to 4)	Contact Anritsu Networks support.	
Error (3)	CLI Command %s, failed during restoration %msg	Command %s failed during the configuration restore at start. The error message is %msg.	Contact Anritsu Networks support.

Severity	Syslog message	Occurs when	Action
Notice (5)	The buffer of queue exceeded the limit. [S:#M,Q:#Q]	The packet buffer usage of Queue Q generated in Scenario M exceeded the limit value.	Packets were discarded because the queue buffer was full. Check the input burst length setting.
	The buffer of queue is less than 50% of the limit. [S:#M,Q:#Q]	The packet buffer usage of Queue Q generated in Scenario M exceeded the limit value, and then dropped below 50% of the limit value.	No recovery measure is required.
	Flow registration failure for the system.	The number of flows in the device exceeded the maximum value.	Check the traffic state and various settings.
	Flow registration available for the system.	The number of flows in the device exceeded the maximum value, and then dropped below 50% of the maximum value.	No recovery measure is required.
	Queue allocation failure for the system.	The number of individual queues in the device exceeded the maximum value.	The action when the maximum number of individual queues is exceeded is applied. Check the traffic status.
	Queue allocation available for the system.	The number of individual queues in the device exceeded the maximum value, and then dropped below 90% of the maximum value.	No recovery measure is required.
	Queue allocation failure for the scenario. [S:#M]	The number of individual queues in Scenario M exceeded the limit.	The action when the maximum number of individual queues is exceeded is applied. Check the traffic status.
	Queue allocation available for the scenario. [S:#M]	The number of individual queues in Scenario M exceeded the limit value, and then dropped below 50% of the limit value.	No recovery measure is required.
	Flow learn queue overflow	The traffic that exceeds the TCP session learning performance was input.	The TCP acceleration for the session that failed to learn is not performed. Check the traffic state.
	Detected MCU-C failure[xx]	An MCU-C error is detected.	Contact Anritsu Networks support.
	Detected MCU-C recovery	The MCU-C error is recovered.	No recovery measure is required.
	Detected MCU-S failure[xx]	An MCU-S error is detected.	Contact Anritsu Networks support.
	Detected MCU-S recovery	The MCU-S error is recovered.	No recovery measure is required.

Severity	Syslog message	Occurs when	Action
Notice (5) (Continued)	Wan-accel scenario switched to secondary-peer. [S:#M]	The opposite device is switched from Primary-peer to Secondary-peer for the WAN-accel scenario M.	Check the line state and opposing device of Primary-peer.
	Wan-accel scenario switched back to primary-peer. [S:#M]	Fallback of the opposite device is performed from Secondary-peer to Primary-peer for the WAN-accel scenario M.	No recovery measure is required.
	Wan-accel scenario switched to Bypass status.(TCP connection error) [S:#M]	WAN-accel scenario M switched the communication status to Bypass. (TCP connection error)	No recovery measure is required.
	Wan-accel scenario switched to Bypass status.(RTT threshold) [S:#M]	WAN-accel scenario M switched the communication status to Bypass. (Falls below the RTT threshold.)	No recovery measure is required.
	Wan-accel scenario switched to Bypass status.(ping timeout) [S:#M]	WAN-accel scenario M switched the communication status to Bypass. (ping did not pass through.)	No recovery measure is required.
	Wan-accel scenario switched to Bypass status.(Peer scenario error) [S:#M]	WAN-accel scenario M switched the communication status to Bypass. (Scenario could not be found.)	No recovery measure is required.
	Wan-accel scenario switched to Acceleration status. [S:#M]	WAN-accel scenario M switched the communication status to Acceleration.	No recovery measure is required.
	Wan-accel scenario switched to Force Bypass status. [S:#M]	WAN-accel scenario M switched the communication status to Force Bypass.	No recovery measure is required.
	Appli-Accel Sessions exceeded the limit. [P:#P]	The upper limit of the Appli-Accel session available in the entire device is exceeded, and Appli-Accel for the excess sessions is disabled. Protocol is #P.	No recovery measure is required.

Severity	Syslog message	Occurs when	Action															
Notice (5) (Continued)	Appli-Accel Sessions exceeded the limit. [P:#P, S:#M]	The upper limit of the Appli-Accel session specified for each scenario is exceeded, and Appli-Accel for the excess sessions is disabled. Protocol is #P. Scenario is #M.	No recovery measure is required.															
	Appli-Accel Sessions is less than 50% of the limit. [P:#P]	The Appli-Accel session usage rate is recovered to 50% or less of the upper limit available in the entire device. Protocol is #P.	No recovery measure is required.															
	Appli-Accel Sessions is less than 50% of the limit. [P:#P, S:#M]	The Appli-Accel session usage rate is recovered to 50% or less of the upper limit specified for each scenario. Protocol is #P. Scenario is #M.	No recovery measure is required.															
	Appli-Accel Buffer almost full. [P:#P, ID:#I]	Usage ratio of the buffer being used in Appli-Accel exceeded 90%. Protocol is #P. Buffer specific ID is #I.	Check the WAN-accel scenario setting that uses Appli-Accel of the protocol #P.															
	Appli-Accel Buffer recoverd. [P:#P, ID:#I]	Usage ratio of the buffer being used in Appli-Accel is recovered to 50% or less. Protocol is #P. Buffer specific ID is #I.	No recovery measure is required.															
	Session limits between monitoring manager occurred.	The limit number of Monitoring Manager 2 connections is exceeded.	Monitoring Manager 2 may not be able to get information when the following limits are exceeded: Ensure the limits are not exceeded.															
	Session limits between monitoring manager is released.	The number of Monitoring Manager 2 connections exceeded the limit, and then dropped below it.	<table border="1"> <thead> <tr> <th>Cycle</th> <th>Scenarios</th> <th>Monitoring Manager 2 connections</th> </tr> </thead> <tbody> <tr> <td>10 sec</td> <td>2000</td> <td>2</td> </tr> <tr> <td>10 sec</td> <td>4000</td> <td>1</td> </tr> <tr> <td>30 sec</td> <td>No limit</td> <td>4</td> </tr> <tr> <td>60 sec</td> <td>No limit</td> <td>4</td> </tr> </tbody> </table>	Cycle	Scenarios	Monitoring Manager 2 connections	10 sec	2000	2	10 sec	4000	1	30 sec	No limit	4	60 sec	No limit	4
	Cycle	Scenarios	Monitoring Manager 2 connections															
10 sec	2000	2																
10 sec	4000	1																
30 sec	No limit	4																
60 sec	No limit	4																
Monitoring manager session connected. (xxx.xxx.xxx.xxx)	Session connected to the monitoring manager 2. (xxx.xxx.xxx.xxx)	No recovery measure is required.																

Severity	Syslog message	Occurs when	Action
Notice (5) (Continued)	Monitoring manager session disconnected [State:#N]. (xxx.xxx.xxx.xxx)	Session disconnected with the monitoring manager 2. (xxx.xxx.xxx.xxx) (State: #N is communication state)	Check the node registration and connection status of the monitoring manager 2.
	Bypass state was changed to on.	The Network port is set to the bypass ON state.	The syslog describing the cause of the bypass connection is recorded immediately before this syslog. Specify the reason for the bypass connection state, and take necessary measures.
	Bypass state was changed to off.	The Network port is set to the bypass OFF state.	No recovery measure is required.
	Detected MCU-B failure[xx]	An error of MCU-B is detected.	Contact our support.
	Detected MCU-B recovery	An error of MCU-B is recovered.	No recovery measure is required.
Informational (6)	Port #N/#M changed Up from Down.	Port link-up occurred (#N is 1) (#M is 1 to 4)	No recovery measure is required.
	Port #N/#M changed Down from Up.	Port link-down occurred (#N is 1) (#M is 1 to 4)	Check the following. <ul style="list-style-type: none"> • If any cable disconnection occurred • If the right cable (multi mode/single mode, straight/cross) is used • If the Speed/Duplex and Pause settings of the Network port match the connected device
	Port #N/#M changed PowerDown with Link Pass Through.	The link down transfer feature operated. (#N is 1) (#M is 1 to 4)	Check the following. <ul style="list-style-type: none"> • If any cable disconnection occurred • If the right cable (multi mode/single mode, straight/cross) is used • If the Speed/Duplex and Pause settings of the Network port match the connected device
	Warning. Port #N/#M Oper duplex is Half.	Port link-up in a half-duplex occurred (#N is 1) (#M is 1 to 4)	Check the following. <ul style="list-style-type: none"> • If the Speed/Duplex settings of the Network port match the connected device
	Management Ethernet Port changed Up from Down.	Management Ethernet port link-up occurred.	No recovery measure is required.
	Management Ethernet Port changed Down from Up.	Management Ethernet port link-down occurred.	Check the following. <ul style="list-style-type: none"> • If any cable disconnection occurred • If the right cable is used

Severity	Syslog message	Occurs when	Action
Informational (6) (Continued)	AnritsuPureFlow Software Version x.x.x.x	Device startup	No recovery measure is required.
	User %s authentication from RADIUS server was Accept	RADIUS authentication of user name %s was accepted.	No recovery measure is required.
	User %s authentication from RADIUS server was Reject	RADIUS authentication of user name %s was rejected.	No recovery measure is required.
	User %s authentication from RADIUS server was Timeout	RADIUS authentication of user name %s timed out.	No recovery measure is required.
	User root logged in by SSH(xxx.xxx.xxx.xxx)	A user of the SSH host logged into this device.	No recovery measure is required.
	User root logged in by TELNET	A user of the Telnet host logged into this device.	No recovery measure is required.
	OpenFlow session connected. (xxx.xxx.xxx.xxx)	Connection with the OpenFlow controller (xxx.xxx.xxx.xxx) is made.	No recovery measure is required.
	OpenFlow session disconnected. (xxx.xxx.xxx.xxx)	Connection with the OpenFlow controller (xxx.xxx.xxx.xxx) is disconnected.	No recovery measure is required.
Software License : %s	The software license %s is valid. (NONE if the software license is invalid.	No recovery measure is required.	

(Blank page)

Appendix C List of SNMP Traps

Table 3 lists the SNMP traps.

Only those Traps that are enabled are sent out. To enable or disable a Trap, use the set snmp traps command. For details of the commands, see “PureFlow WSX Unified Network Controller NF7600 series Command Reference TCP Acceleration Edition”.

Table 3 SNMP Trap List

MIB object name	Name of command setting	Occurs when	Action
coldStart(1.3.6.1.6.3.1.1.5.1)	coldstart	Device startup is complete.	Check the following. <ul style="list-style-type: none"> • If any power disconnection occurred • If the reset button was pressed • If the restart command is executed • If the automatic boot feature is working
warmStart(1.3.6.1.6.3.1.1.5.2)	warmstart	Not output	
linkDown(1.3.6.1.6.3.1.1.5.3)	linkdown	Port link-down	Check the following. <ul style="list-style-type: none"> • If any cable disconnection occurred • If the right cable (single mode/multi mode, straight/cross) is used • If the Speed/Duplex and Pause settings of the Network port match the connected device
linkUp(1.3.6.1.6.3.1.1.5.4)	linkup	Link-up	No recovery measure is required.
authenticationFailure(1.3.6.1.6.3.1.1.5.5)	authentication	SNMP invalid access detected	Check if the access permission community name, IP address, and level (get/set) set to this device match the SNMP manager side.
pfGsPowerInsertEvent(1.3.6.1.4.1.1151.2.1.7.20.0.3)	powerinsert	Power supply is inserted	No recovery measure is required.
pfGsPowerExtractEvent(1.3.6.1.4.1.1151.2.1.7.20.0.4)	powerextract	Power supply is removed.	No recovery measure is required.

MIB object name	Name of command setting	Occurs when	Action
pfGsPowerFailureEvent(1.3.6.1.4.1.1151.2.1.7.20.0.5)	powerfailure	Power supply failure is detected.	Check the following. <ul style="list-style-type: none"> • If the power cable is connected • If the supply voltage is within the valid range (AC 100 V to 240 VAC/127 V / AC 200 V to AC 240 V) • If the power supply fan is working
pfGsPowerRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.6)	powerrecovery	Power supply failure is recovered.	No recovery measure is required.
pfGsModuleFailureAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.7)	modulefailurealarm	Module error detected	Contact Anritsu support.
pfGsModuleFailureRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.8)	modulefailurerecovery	Module error recovered	No recovery measure is required.
pfGsFanInsertEvent(1.3.6.1.4.1.1151.2.1.7.20.0.11)	faninsert	Fan unit is inserted.	No recovery measure is required.
pfGsFanExtractEvent(1.3.6.1.4.1.1151.2.1.7.20.0.12)	fanextract	Fan unit is removed.	No recovery measure is required.
pfGsFanFailureEvent(1.3.6.1.4.1.1151.2.1.7.20.0.13)	fanfailure	Fan unit failure is detected.	Check the following. <ul style="list-style-type: none"> • If the fan is working
pfGsFanRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.14)	fanrecovery	Fan unit failure is recovered.	No recovery measure is required.
pfGsQueueBuffAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.15)	queuebuffalarm	The packet buffer usage of in the scenario exceeded the limit value.	Packets were discarded because the queue buffer was full. Check the input burst length setting.
pfGsQueueBuffRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.16)	queuebuffrecovery	The packet buffer usage in the scenario exceeded the limit value, and then dropped below 50% of the limit value.	No recovery measure is required.
pfGsSystemBuffAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.17)	systembuffalarm	The usage of the system buffer exceeded 90%.	Check the traffic state and various settings.

MIB object name	Name of command setting	Occurs when	Action
pfGsSystemBuffRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.18)	systembuffrecovery	The usage of the system buffer exceeded 90% and then dropped below 50%.	No recovery measure is required.
pfGsSystemHeatAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.19)	systemheatalarm	The system temperature exceeded 50°C or dropped below -5°C.	Modify the air conditioning or device layout so that the environment temperature becomes 0°C to 40°C or lower.
pfGsSystemHeatRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.20)	systemheatrecovery	The system temperature exceeded 50°C and then dropped below 45°C. Or it dropped below -5°C, and then exceeded 0°C.	No recovery measure is required.
pfGsIndividualQueueAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.21)	queueallocalarm	The number of individual queues in the device exceeded the maximum value.	The action in case the maximum number of individual queues is exceeded is applied. Check the traffic status.
pfGsIndividualQueueRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.22)	queueallocalrecovery	The number of individual queues in the device exceeded the maximum value, and then dropped below 90% of the maximum value.	No recovery measure is required.
pfGsMaxQnumAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.23)	maxqnumalarm	The number of individual queues in the scenario exceeded the limit.	The action in case the maximum number of individual queues is exceeded is applied. Check the traffic status.
pfGsMaxQnumRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.24)	maxqnumrecovery	The number of individual queues in the scenario exceeded the limit value, and then dropped below 50% of the limit value.	No recovery measure is required.
pfGsQueueBuffByScIdAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.25)	queuebuffalarm	The packet buffer usage amount of the applicable scenario exceeds the limit value.	Packet being discarded occurs due to the queue buffer full. Please check the input burst length setting.
pfGsQueueBuffByScIdRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.26)	queuebuffrecovery	The packet buffer usage amount of the applicable scenario exceeds the limit value, and falls below 50% of the limit value.	No recovery measure is required.

MIB object name	Name of command setting	Occurs when	Action
pfGsMaxQnumByScIdAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.27)	maxqnumalarm	The number of individual queues of the applicable scenario exceeds the limit value.	The action taken when exceeding the maximum number is applied because the individual queues reached the limit number of scenarios. Check the traffic state.
pfGsMaxQnumByScIdRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.28)	maxqnumrecovery	The individual queues of the applicable scenario exceed the limit value, and fall below 50% of the limit value.	No recovery measure is required.
pfGsTcpAccelBypassByScIdAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.29)	tcpbypassalarm	The applicable WAN-accel scenario switched the communication status to Bypass.	No recovery measure is required.
pfGsTcpAccelBypassByScIdRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.30)	tcpbypassrecovery	The applicable WAN-accel scenario switched the communication status from Bypass.	No recovery measure is required.
pfGsTcpAccelPeerByScIdAlarmEvent(1.3.6.1.4.1.1151.2.1.7.20.0.31)	peeralarm	The applicable WAN-accel scenario switched the opposing device from Primary-peer to Secondary-peer.	Check the Primary-peer line status and opposing device status.
pfGsTcpAccelPeerByScIdRecoveryEvent(1.3.6.1.4.1.1151.2.1.7.20.0.32)	peerrecovery	The applicable WAN-accel scenario switched the opposing device from Secondary-peer to Primary-peer.	No recovery measure is required.
pfGsBypassOnEvent(1.3.6.1.4.1.1151.2.1.7.20.0.33)	bypasson	The network bypass function disconnected the communication path from the Normal side and connected to the Bypass side.	Specify the reason for the bypass state, and take necessary measures.
pfGsBypassOffEvent(1.3.6.1.4.1.1151.2.1.7.20.0.34)	bypassoff	The network bypass function disconnected the communication path from the Bypass side and connected to the Normal side.	No recovery measure is required.

Appendix D Enterprise MIB List



Table 4 shows a list of the Enterprise MIB objects of this device.

Table 4 List of Enterprise MIB Objects of PureFlow WSX series

MIB group	MIB object name	Description
pureFlowGsMib		<p>PureFlow GS Enterprise MIB tree. The object ID is 1.3.6.1.4.1.1151.2.1.7.</p> <p>Objects in the tree and their IDs (in parentheses) are as follows:</p> <p>PureFlow GS Enterprise MIB tree is a common MIB tree for the PureFlow GS series. This manual describes PureFlow WSX series MIB objects.</p>
pfGsSystem(1.3.6.1.4.1.1151.2.1.7.1)	pfGsSystemType(1.3.6.1.4.1.1151.2.1.7.1.1)	Shows the software model name. nf7600s001a(6) :NF7600-S001A
	pfGsSystemSlotNumber(1.3.6.1.4.1.1151.2.1.7.1.2)	Shows the number of slots for installing modules.
	pfGsSystemSoftwareRev(1.3.6.1.4.1.1151.2.1.7.1.3)	Shows the version of the system software.
	pfGsSystemOperationTime(1.3.6.1.4.1.1151.2.1.7.1.5)	Shows the elapsed time from system startup. The unit is 10 ms. This MIB object is updated every hour. Therefore, all digits other than time are always 0.
	pfGsSystemCcpu5sec(1.3.6.1.4.1.1151.2.1.7.1.6)	Shows the average value of control-system CPU use rate in the last 5 seconds.
	pfGsSystemCcpu1min(1.3.6.1.4.1.1151.2.1.7.1.7)	Shows the average value of control-system CPU use rate in the last 1 minute.
	pfGsSystemCcpu5min(1.3.6.1.4.1.1151.2.1.7.1.8)	Shows the average value of control-system CPU use rate in the last 5 minutes.
	pfGsSystemCcpuMemory5sec(1.3.6.1.4.1.1151.2.1.7.1.9)	Shows the average value of memory use rate of the control-system CPU in the last 5 seconds.
	pfGsSystemCcpuMemory1min(1.3.6.1.4.1.1151.2.1.7.1.10)	Shows the average value of memory use rate of the control-system CPU in the last 1 minute.
	pfGsSystemCcpuMemory5min(1.3.6.1.4.1.1151.2.1.7.1.11)	Shows the average value of memory use rate of the control-system CPU in the last 5 minutes.
	pfGsSystemFcpuTable(1.3.6.1.4.1.1151.2.1.7.1.12)	<p>The table for the CPU and memory use rates of the forwarding-system CPU.</p> <p>This table contains the following objects:</p>
pfGsSystemFcpuEntry(1.3.6.1.4.1.1151.2.1.7.1.12.1)	<p>The entry table for the CPU and memory use rates of the forwarding-system CPU. The table index is pfSystemFcpuIndex.</p> <p>This table contains the following objects:</p>	

MIB group	MIB object name	Description
pfGsSystem(1.3.6.1.4.1.1151.2.1.7.1) (Continued)	pfGsSystemFcpuIndex(1.3.6.1.4.1.1151.2.1.7.1.12.1.1)	Shows the forwarding-system CPU number. Front view <div style="border: 1px solid black; width: 100px; height: 20px; margin: 10px auto; text-align: center;">1</div>
	pfGsSystemFcpu5sec(1.3.6.1.4.1.1151.2.1.7.1.12.1.2)	Shows the average value of forwarding-system CPU use rate in the last 5 seconds.
	pfGsSystemFcpu1min(1.3.6.1.4.1.1151.2.1.7.1.12.1.3)	Shows the average value of forwarding-system CPU use rate in the last 1 minute.
	pfGsSystemFcpu5min(1.3.6.1.4.1.1151.2.1.7.1.12.1.4)	Shows the average value of forwarding-system CPU use rate in the last 5 minutes.
	pfGsSystemFcpuMemory5sec(1.3.6.1.4.1.1151.2.1.7.1.12.1.5)	Shows the average value of forwarding-system memory use rate in the last 5 seconds.
	pfGsSystemFcpuMemory1min(1.3.6.1.4.1.1151.2.1.7.1.12.1.6)	Shows the average value of forwarding-system memory use rate in the last 1 minute.
	pfGsSystemFcpuMemory5min(1.3.6.1.4.1.1151.2.1.7.1.12.1.7)	Shows the average value of forwarding-system memory use rate in the last 5 minutes.
	pfGsSystemBuffTable(1.3.6.1.4.1.1151.2.1.7.1.13)	The table for system buffer. This table contains the following objects:
	pfGsSystemBuffEntry(1.3.6.1.4.1.1151.2.1.7.1.13.1)	The entry table is for system buffer. The table index is pfGsSystemBuffIndex. This table contains the following objects:
	pfGsSystemBuffIndex(1.3.6.1.4.1.1151.2.1.7.1.13.1.1)	Shows the system buffer number. 1: Packet buffer 2: The message block for the bandwidth control engine 3: The packet output command area 4: The packet buffer for In-band transmitted packets 5: Not Used 6: Not Used 7: Not Used 8: Not Used 9: A temporary area for packets in progress
	pfGsSystemBuffMax(1.3.6.1.4.1.1151.2.1.7.1.13.1.2)	Shows the maximum capacity of the system buffer.
	pfGsSystemBuffRemaining(1.3.6.1.4.1.1151.2.1.7.1.13.1.3)	Shows the remaining capacity of the system buffer.

MIB group	MIB object name	Description
pfGsSystem(1.3.6.1.4.1.1151.2.1.7.1) (Continued)	pfGsSystemTempTable(1.3.6.1.4.1.1151.2.1.7.1.14)	The table for system temperature. This table contains the following objects:
	pfGsSystemTempEntry(1.3.6.1.4.1.1151.2.1.7.1.14.1)	The entry table for system temperature. The table index is pfGsSystemTempIndex. This table contains the following objects:
	pfGsSystemTempIndex(1.3.6.1.4.1.1151.2.1.7.1.14.1.1)	Shows the system temperature number. 1: Intake 2: Not Used 3: Not Used 4: Not Used 5: Not Used 6: Not Used 7: Not Used 8: Not Used 9: Not Used
	pfGsSystemTempValue(1.3.6.1.4.1.1151.2.1.7.1.14.1.2)	Shows the system temperature value. The unit is Centigrade.
	pfGsSystemBypassMode(1.3.6.1.4.1.1151.2.1.7.1.15)	Displays the control mode of the network bypass function. notAvailable(0): The network bypass function is not available in this system. auto(1): Auto control on (2): Forced bypass off (3): Forced non-bypass
	pfGsSystemBypassState(1.3.6.1.4.1.1151.2.1.7.1.16)	Displays the network bypass state. notAvailable(0): The network bypass function is not available in this system. on(1): Bypass state off (2): Non-bypass state
pfGsSystemBypassTimeRemaining(1.3.6.1.4.1.1151.2.1.7.1.17)	The remaining time of temporary bypass switching is shown in seconds. If temporary bypass switching is not being executed, 0 second is displayed.	

MIB group	MIB object name	Description
pfGsModule(1.3.6.1.4.1.1151.2.1.7.2)	pfGsModuleTable(1.3.6.1.4.1.1151.2.1.7.2.1)	The table for module information. This table contains the following objects:
	pfGsModuleEntry(1.3.6.1.4.1.1151.2.1.7.2.1.1)	The entry table for module information. The table index is pfGsModuleIndex. This table contains the following objects:
	pfGsModuleIndex(1.3.6.1.4.1.1151.2.1.7.2.1.1.1)	Shows the module number. Front view 
	pfGsModuleLocation(1.3.6.1.4.1.1151.2.1.7.2.1.1.2)	Shows the implemented slot number of the module. (It is the same as the module number.) Front view 
	pfGsModuleType(1.3.6.1.4.1.1151.2.1.7.2.1.1.3)	Shows the module type. unknown(1): Other than the following: empty(2): Not implemented ge2gt(3): GbE/2T fe2ft(4): FE/2T xge2sfp(5): 10GbE/2SFP+ xge4sfp(6): 10GbE/4SFP+ ge4sfp(7): GbE/4SFP
	pfGsModuleDescr(1.3.6.1.4.1.1151.2.1.7.2.1.1.4)	Shows the module name.
	pfGsModulePortNumber(1.3.6.1.4.1.1151.2.1.7.2.1.1.5)	Shows the number of ports implemented on the module.
	pfGsModuleOperStatus(1.3.6.1.4.1.1151.2.1.7.2.1.1.6)	Shows the module status. other(1): other than the following: operational(2): Normal malfunctioning(3): Error other than 6 notpresent(4): Not implemented standby(5): Not used notResponding(6): No response
	pfGsModuleRevision(1.3.6.1.4.1.1151.2.1.7.2.1.1.7)	Shows the hardware revision of the module.
pfGsModuleSerialNumber(1.3.6.1.4.1.1151.2.1.7.2.1.1.8)	Shows the serial number of the module.	

MIB group	MIB object name	Description								
pfGsPower(1.3.6.1.4.1.1151.2.1.7.3)	pfGsPowerTable(1.3.6.1.4.1.1151.2.1.7.3.1)	The table for the power supply unit information. This table contains the following objects:								
	pfGsPowerEntry(1.3.6.1.4.1.1151.2.1.7.3.1.1)	The entry table for the power supply unit information. The table index is pfGsPowerIndex. This table contains the following objects:								
	pfGsPowerIndex(1.3.6.1.4.1.1151.2.1.7.3.1.1.1)	Shows the power supply unit number. Back view <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Fan</td> <td>Fan</td> <td>Power</td> <td>Power</td> </tr> <tr> <td>2</td> <td>1</td> <td>2</td> <td>1</td> </tr> </table>	Fan	Fan	Power	Power	2	1	2	1
	Fan	Fan	Power	Power						
	2	1	2	1						
	pfGsPowerOperStatus(1.3.6.1.4.1.1151.2.1.7.3.1.1.2)	Shows the power supply unit status. other(1): other than the following: operational(2): Normal malfunctioning(3): Error (input error or fan stop) notpresent(4): Not implemented outputerror(5): (Not used) inputerror(6): (Not used) fanfailure(7): (Not used)								
pfGsPowerUpTime(1.3.6.1.4.1.1151.2.1.7.3.1.1.3)	Shows the elapsed time after the power supply unit is inserted. The unit is 10 ms.									
pfGsPowerFanSpeed(1.3.6.1.4.1.1151.2.1.7.3.1.1.4)	Shows the fan revolutions of the power supply unit. The unit is RPM.									
pfGsxFan(1.3.6.1.4.1.1151.2.1.7.4)	pfGsxFanTable(1.3.6.1.4.1.1151.2.1.7.4.1)	The table for fan unit information. This table contains the following objects:								
	pfGsxFanEntry(1.3.6.1.4.1.1151.2.1.7.4.1.1)	The entry table for fan unit information. The table index is pfGsxFanIndex. This table contains the following objects:								
	pfGsxFanIndex(1.3.6.1.4.1.1151.2.1.7.4.1.1.1)	Shows the fan unit number. Back view <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Fan</td> <td>Fan</td> <td>Power</td> <td>Power</td> </tr> <tr> <td>2</td> <td>1</td> <td>2</td> <td>1</td> </tr> </table>	Fan	Fan	Power	Power	2	1	2	1
	Fan	Fan	Power	Power						
	2	1	2	1						
	pfGsxFanOperStatus(1.3.6.1.4.1.1151.2.1.7.4.1.1.2)	Shows the fan unit status. other(1): other than the following: operational(2): Normal malfunctioning(3): Error (fan stop) notpresent(4): Not implemented								
pfGsxFanUpTime(1.3.6.1.4.1.1151.2.1.7.4.1.1.3)	Shows the elapsed time after the fan unit is inserted. The unit is 10 ms.									
pfGsxFanSpeed(1.3.6.1.4.1.1151.2.1.7.4.1.1.4)	Shows the fan revolutions of the fan unit. The unit is RPM.									

MIB group	MIB object name	Description
pfGsFlowInformation(1.3.6.1.4.1.1151.2.1.7.8)	pfGsFlowInformationResourceTotal(1.3.6.1.4.1.1151.2.1.7.8.1)	Shows the total number of flows the device can use.
	pfGsFlowInformationResourceUsed(1.3.6.1.4.1.1151.2.1.7.8.2)	Shows the number of flows being used by the device.
	pfGsFlowInformationResourceAvailable(1.3.6.1.4.1.1151.2.1.7.8.3)	Shows the number of flows to be used by the device.
pfGsxScenarioStatistics(1.3.6.1.4.1.1151.2.1.7.9)	pfGsxScenarioStatisticsTable(1.3.6.1.4.1.1151.2.1.7.9.1)	The table for scenario counter. This table contains the following objects:
	pfGsxScenarioStatisticsEntry(1.3.6.1.4.1.1151.2.1.7.9.1.1)	The entry table for scenario counter. The table index is pfGsxScenarioStatisticsScenarioSortIndex. This table contains the following objects: Reference: The next table shows how to get an object OID in this table.
	pfGsxScenarioStatisticsScenarioSortIndex(1.3.6.1.4.1.1151.2.1.7.9.1.1.1)	Shows the sort number of the scenario. A sort number is added automatically when a scenario is registered or deleted. Sort numbers correspond to the scenario order.
	pfGsxScenarioStatisticsScenarioName(1.3.6.1.4.1.1151.2.1.7.9.1.1.2)	Shows the scenario name.
	pfGsxScenarioStatisticsScenarioType(1.3.6.1.4.1.1151.2.1.7.9.1.1.3)	Shows the type of the scenario. discard(0): Discard scenario individual(1): Individual queue scenario aggregate(2): Aggregate queue scenario application(3): (Not used) wanaccel(4): Traffic Acceleration Scenario
	pfGsxScenarioStatisticsRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.4)	Shows the number of received octets of the scenario.
	pfGsxScenarioStatisticsRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.5)	Shows the number of received packets of the scenario.
	pfGsxScenarioStatisticsTxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.6)	Shows the number of transmitted octets of the scenario.
	pfGsxScenarioStatisticsTxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.7)	Shows the number of transmitted packets of the scenario.
	pfGsxScenarioStatisticsDiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.8)	Shows the number of discarded octets of the scenario.
pfGsxScenarioStatisticsDiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.9)	Shows the number of discarded packets of the scenario.	

MIB group	MIB object name	Description
pfGsxScenarioStatistics(1.3.6.1.4.1.1151.2.1.7.9) (Continued)	pfGsxScenarioStatisticsHC RxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.10)	Shows the number of received octets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsHC RxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.11)	Shows the number of received packets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsHC TxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.12)	Shows the number of transmitted octets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsHC TxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.13)	Shows the number of transmitted packets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsHC DiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.14)	Shows the number of discarded octets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsHC DiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.15)	Shows the number of discarded packets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsDefaultQueueRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.16)	Shows the number of received octets of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueueRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.17)	Shows the number of received packets of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueueTxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.18)	Shows the number of transmitted octets of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueueTxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.19)	Shows the number of transmitted packets of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueueDiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.20)	Shows the number of discarded octets of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueueDiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.21)	Shows the number of discarded packets of the scenario default queue.

MIB group	MIB object name	Description
pfGsxScenarioStatistics(1.3.6.1.4.1.1151.2.1.7.9) (Continued)	pfGsxScenarioStatisticsDefaultQueHCRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.22)	Shows the number of received octets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsDefaultQueHCRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.23)	Shows the number of received packets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsDefaultQueHCTxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.24)	Shows the number of transmitted octets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsDefaultQueHCTxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.25)	Shows the number of transmitted packets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsDefaultQueHCDiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.26)	Shows the number of discarded octets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatisticsDefaultQueHCDiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.27)	Shows the number of discarded packets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
pfGsxScenarioInformation(1.3.6.1.4.1.1151.2.1.7.10)	pfGsxScenarioInformationTable(1.3.6.1.4.1.1151.2.1.7.10.1)	The table for scenario information. This table contains the following objects:
	pfGsxScenarioInformationEntry(1.3.6.1.4.1.1151.2.1.7.10.1.1)	The entry table for scenario information. The table index is pfGsxScenarioInformationScenarioSortIndex. This table contains the following objects: Reference: The next table shows how to get an object OID in this table.
	pfGsxScenarioInformationScenarioSortIndex(1.3.6.1.4.1.1151.2.1.7.10.1.1.1)	Shows the sort number of the scenario. A sort number is added automatically when a scenario is registered or deleted. Sort numbers correspond to the scenario order.
	pfGsxScenarioInformationScenarioName(1.3.6.1.4.1.1151.2.1.7.10.1.1.2)	Shows the scenario name.

MIB group	MIB object name	Description
pfGsxScenarioInformation(1.3.6.1.4.1.1151.2.1.7.10) (Continued)	pfGsxScenarioInformationScenarioType(1.3.6.1.4.1.1151.2.1.7.10.1.1.3)	Shows the type of the scenario. discard(0): Discard scenario individual(1): Individual queue scenario aggregate(2): Aggregate queue scenario application(3): (Not used) wanaccel(4): Traffic Acceleration Scenario
	pfGsxScenarioInformationDefFlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.4)	Shows the number of default flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInformationClass1FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.5)	Shows the number of Class 1 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInformationClass2FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.6)	Shows the number of Class 2 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInformationClass3FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.7)	Shows the number of Class 3 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInformationClass4FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.8)	Shows the number of Class 4 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInformationClass5FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.9)	Shows the number of Class 5 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInformationClass6FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.10)	Shows the number of Class 6 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInformationClass7FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.11)	Shows the number of Class 7 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInformationClass8FlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.12)	Shows the number of Class 8 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.

MIB group	MIB object name	Description
pfGsxScenarioInformation(1.3.6.1.4.1.1151.2.1.7.10) (Continued)	pfGsxScenarioInformationTotalFlowNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.13)	Shows the total number of flows generated in connection with the scenario. Note: This object is not supported. The value is same as the number of default flows.
	pfGsxScenarioInformationDefBuffRatio(1.3.6.1.4.1.1151.2.1.7.10.1.1.25)	Shows the current buffer use rate of the scenario default queue. The unit is %.
	pfGsxScenarioInformationDefBuff(1.3.6.1.4.1.1151.2.1.7.10.1.1.26)	Shows the current buffer usage of the scenario default queue. The unit is bytes.
	pfGsxScenarioInformationDefPeakBuffRatio(1.3.6.1.4.1.1151.2.1.7.10.1.1.27)	Shows the current buffer peak use rate of the scenario default queue. The unit is %.
	pfGsxScenarioInformationDefPeakBuff(1.3.6.1.4.1.1151.2.1.7.10.1.1.28)	Shows the current buffer peak usage of the scenario default queue. The unit is bytes.
	pfGsxScenarioInformationTxPeakRateBps(1.3.6.1.4.1.1151.2.1.7.10.1.1.29)	Shows the peak transmission rate of the scenario in the last 1 minute. The unit is bits/s. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioInformationTxAveRateBps(1.3.6.1.4.1.1151.2.1.7.10.1.1.31)	Shows the average transmission rate of the scenario in the last 1 minute. The unit is bits/s. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioInformationIndQueNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.33)	Shows the number of current individual queues in the individual queue mode scenario. For scenarios other than the individual queue mode, the value is fixed to 0.
	pfGsxScenarioInformationAccelSessNum(1.3.6.1.4.1.1151.2.1.7.10.1.1.34)	Shows the number of TCP sessions that apply current WAN acceleration of the acceleration mode scenario. For scenarios other than the acceleration mode, the value is fixed to 0.
	pfGsxScenarioInformationAccelBypassStatus(1.3.6.1.4.1.1151.2.1.7.10.1.1.35)	Shows the current bypass status of the acceleration mode scenario.
pfGsxScenarioInformationAccelActivePeer(1.3.6.1.4.1.1151.2.1.7.10.1.1.36)	Shows information on the current active Peer of the acceleration mode scenario.	

MIB group	MIB object name	Description
pfGsxScenarioStatByScId(1.3.6.1.4.1.1151.2.1.7.11)	pfGsxScenarioStatByScIdTable(1.3.6.1.4.1.1151.2.1.7.11.1)	The table for scenario counter. This table contains the following objects:
	pfGsxScenarioStatByScIdEntry(1.3.6.1.4.1.1151.2.1.7.11.1.1)	The entry table for scenario counter. The table index is pfGsxScenarioStatByScIdScenarioId. This table contains the following objects: Reference: The next table shows how to get an object OID in this table.
	pfGsxScenarioStatByScIdScenarioId(1.3.6.1.4.1.1151.2.1.7.11.1.1.1)	Shows the ID of the scenario. The scenario ID can be registered when registering the scenario. If no ID is specified for the scenario at registration, an ID is automatically assigned to the scenario.
	pfGsxScenarioStatByScIdScenarioName(1.3.6.1.4.1.1151.2.1.7.11.1.1.2)	Shows the scenario name.
	pfGsxScenarioStatByScIdScenarioType(1.3.6.1.4.1.1151.2.1.7.11.1.1.3)	Shows the type of the scenario. discard(0): Discard scenario individual(1): Individual queue scenario aggregate(2): Aggregate queue scenario application(3): (Not used) wanaccel(4): Traffic Acceleration Scenario
	pfGsxScenarioStatByScIdRxBxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.4)	Shows the number of received octets of the scenario.
	pfGsxScenarioStatByScIdRxBxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.5)	Shows the number of received packets of the scenario.
	pfGsxScenarioStatByScIdTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.6)	Shows the number of transmitted octets of the scenario.
	pfGsxScenarioStatByScIdTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.7)	Shows the number of transmitted packets of the scenario.
	pfGsxScenarioStatByScIdDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.8)	Shows the number of discarded octets of the scenario.
	pfGsxScenarioStatByScIdDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.9)	Shows the number of discarded packets of the scenario.
	pfGsxScenarioStatByScIdHCRxBxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.10)	Shows the number of received octets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
pfGsxScenarioStatByScIdHCRxBxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.11)	Shows the number of received packets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.	

MIB group	MIB object name	Description
pfGsxScenarioStatByScId(1.3.6.1.4.1.1151.2.1.7.11) (Continued)	pfGsxScenarioStatByScIdHCTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.12)	Shows the number of transmitted octets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdHCTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.13)	Shows the number of transmitted packets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdHCDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.14)	Shows the number of discarded octets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdHCDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.15)	Shows the number of discarded packets of the scenario in 64 bits. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdDefaultQueueRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.16)	Shows the number of received octets of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueueRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.17)	Shows the number of received packets of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueueTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.18)	Shows the number of transmitted octets of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueueTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.19)	Shows the number of transmitted packets of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueueDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.20)	Shows the number of discarded octets of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueueDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.21)	Shows the number of discarded packets of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueueHCRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.22)	Shows the number of received octets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdDefaultQueueHCRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.23)	Shows the number of received packets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.

MIB group	MIB object name	Description
pfGsxScenarioStatByScId(1.3.6.1.4.1.1151.2.1.7.11) (Continued)	pfGsxScenarioStatByScIdDefaultQueHCTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.24)	Shows the number of transmitted octets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdDefaultQueHCTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.25)	Shows the number of transmitted packets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdDefaultQueHCDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.26)	Shows the number of discarded octets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdDefaultQueHCDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.27)	Shows the number of discarded packets of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
pfGsxScenarioInfoByScId(1.3.6.1.4.1.1151.2.1.7.12)	pfGsxScenarioInfoByScIdTable(1.3.6.1.4.1.1151.2.1.7.12.1)	The table for scenario information. This table contains the following objects:
	pfGsxScenarioInfoByScIdEntry(1.3.6.1.4.1.1151.2.1.7.12.1.1)	The entry table for scenario information. The table index is pfGsxScenarioInfoByScIdScenarioId. This table contains the following objects: Reference: The next table shows how to get an object OID in this table.
	pfGsxScenarioInfoByScIdScenarioId(1.3.6.1.4.1.1151.2.1.7.12.1.1.1)	Shows the ID of the scenario. The scenario ID can be registered when registering the scenario. If no ID is specified for the scenario at registration, an ID is automatically assigned to the scenario.
	pfGsxScenarioInfoByScIdScenarioName(1.3.6.1.4.1.1151.2.1.7.12.1.1.2)	Shows the scenario name.
	pfGsxScenarioInfoByScIdScenarioType(1.3.6.1.4.1.1151.2.1.7.12.1.1.3)	Shows the type of the scenario. discard(0): Discard scenario individual(1): Individual queue scenario aggregate(2): Aggregate queue scenario application(3): (Not used) wanaccel(4): Traffic Acceleration Scenario
	pfGsxScenarioInfoByScIdDefaultFlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.4)	Shows the number of default flows generated in connection with the scenario.
	pfGsxScenarioInfoByScIdClass1FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.5)	Shows the number of Class 1 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.

MIB group	MIB object name	Description
pfGsxScenarioInfoByScId(1.3.6.1.4.1.1151.2.1.7.12) (Continued)	pfGsxScenarioInfoByScIdClass2FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.6)	Shows the number of Class 2 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInfoByScIdClass3FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.7)	Shows the number of Class 3 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInfoByScIdClass4FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.8)	Shows the number of Class 4 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInfoByScIdClass5FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.9)	Shows the number of Class 5 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInfoByScIdClass6FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.10)	Shows the number of Class 6 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInfoByScIdClass7FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.11)	Shows the number of Class 7 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInfoByScIdClass8FlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.12)	Shows the number of Class 8 flows generated in connection with the scenario. Note: This object is not supported. The value is fixed to zero.
	pfGsxScenarioInfoByScIdTotalFlowNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.13)	Shows the total number of flows generated in connection with the scenario. Note: This object is not supported. The value is same as the number of default flows.
	pfGsxScenarioInfoByScIdDefaultBuffRatio(1.3.6.1.4.1.1151.2.1.7.12.1.1.25)	Shows the current buffer use rate of the scenario default queue. The unit is %.
	pfGsxScenarioInfoByScIdDefaultBuff(1.3.6.1.4.1.1151.2.1.7.12.1.1.26)	Shows the current buffer usage of the scenario default queue. The unit is bytes.
pfGsxScenarioInfoByScIdDefaultPeakBuffRatio(1.3.6.1.4.1.1151.2.1.7.12.1.1.27)	Shows the current buffer peak use rate of the scenario default queue. The unit is %.	

MIB group	MIB object name	Description
pfGsxScenarioInfoByScId(1.3.6.1.4.1.1151.2.1.7.12) (Continued)	pfGsxScenarioInfoByScIdDefPeakBuff(1.3.6.1.4.1.1151.2.1.7.12.1.1.28)	Shows the current buffer peak usage of the scenario default queue. The unit is bytes.
	pfGsxScenarioInfoByScIdTxPeakRateBps(1.3.6.1.4.1.1151.2.1.7.12.1.1.29)	Shows the peak transmission rate of the scenario in the last 1 minute. The unit is bits/s. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioInfoByScIdTxAveRateBps(1.3.6.1.4.1.1151.2.1.7.12.1.1.31)	Shows the average transmission rate of the scenario in the last 1 minute. The unit is bits/s. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioInfoByScIdIndQueNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.33)	Shows the number of current individual queues in the individual queue mode scenario. For scenarios other than the individual queue mode, the value is fixed to 0.
	pfGsxScenarioInfoByScIdAccelSessNum(1.3.6.1.4.1.1151.2.1.7.12.1.1.34)	Shows the number of TCP sessions that apply current WAN acceleration of the acceleration mode scenario. For scenarios other than the acceleration mode, the value is fixed to 0.
	pfGsxScenarioInfoByScIdAccelBypassStatus(1.3.6.1.4.1.1151.2.1.7.12.1.1.35)	Shows the current bypass status of the acceleration mode scenario.
	pfGsxScenarioInfoByScIdAccelActivePeer(1.3.6.1.4.1.1151.2.1.7.12.1.1.36)	Shows information on the current active Peer of the acceleration mode scenario.

For reference:

How to get an OID in the scenario counter and scenario information tables

To get an object OID in the table, refer to the following:

For pfGsxScenarioStatisticsTable,

the OID of pfGsxScenarioStatisticsEntry is as follows:

1.3.6.1.4.1.1151.2.1.7.9.1.1.EntryOID.ScenarioSortIndex

Fixed value

Entry ID: Entry number in the table. Entries appear in the order defined in Table 4. The length is 1.

pfGsxScenarioStatisticsScenarioSortIndex	1
pfGsxScenarioStatisticsScenarioName	2
pfGsxScenarioStatisticsScenarioType	3
pfGsxScenarioStatisticsRxOctets	4
pfGsxScenarioStatisticsRxPackets	5
pfGsxScenarioStatisticsTxOctets	6
pfGsxScenarioStatisticsTxPackets	7
pfGsxScenarioStatisticsDiscardOctets	8
pfGsxScenarioStatisticsDiscardPackets	9
pfGsxScenarioStatisticsHCRxOctets	10
pfGsxScenarioStatisticsHCRxPackets	11
pfGsxScenarioStatisticsHCTxOctets	12
pfGsxScenarioStatisticsHCTxPackets	13
pfGsxScenarioStatisticsHCDiscardOctets	14
pfGsxScenarioStatisticsHCDiscardPackets	15
pfGsxScenarioStatisticsDefaultQueRxOctets	16
pfGsxScenarioStatisticsDefaultQueRxPackets	17
pfGsxScenarioStatisticsDefaultQueTxOctets	18
pfGsxScenarioStatisticsDefaultQueTxPackets	19
pfGsxScenarioStatisticsDefaultQueDiscardOctets	20
pfGsxScenarioStatisticsDefaultQueDiscardPackets	21
pfGsxScenarioStatisticsDefaultQueHCRxOctets	22
pfGsxScenarioStatisticsDefaultQueHCRxPackets	23
pfGsxScenarioStatisticsDefaultQueHCTxOctets	24
pfGsxScenarioStatisticsDefaultQueHCTxPackets	25
pfGsxScenarioStatisticsDefaultQueHCDiscardOctets	26
pfGsxScenarioStatisticsDefaultQueHCDiscardPackets	27

ScenarioSortIndex: The sort number of the scenario. The length is 16. The sort number is consistent with the scenario tree display order, and automatically assigned when registering or deleting a scenario. The sort number changes when the scenario configuration changes since sort numbers are assigned when registering or deleting a scenario. To get the sort number of a specific scenario, use "get next" to get the entire pfGsxScenarioStatisticsTable with the scenario configuration determined, and use the scenario name as the key to find a relevant entry.

For pfGsxScenarioInformationTable,
the OID of pfGsxScenarioInformationEntry is as follows:
1.3.6.1.4.1.1151.2.1.7.10.1.1.EntryOID.ScenarioSortIndex

Fixed value

Entry ID:	Entry number in the table. Note that numbers are not sequential. The length is 1.
	pfGsxScenarioInformationScenarioSortIndex 1
	pfGsxScenarioInformationScenarioName 2
	pfGsxScenarioInformationScenarioType 3
	pfGsxScenarioInformationDefFlowNum 4
	pfGsxScenarioInformationDefBuffRatio 25
	pfGsxScenarioInformationDefBuff 26
	pfGsxScenarioInformationDefPeakBuffRatio 27
	pfGsxScenarioInformationDefPeakBuff 28
	pfGsxScenarioInformationTxPeakRateBps 29
	pfGsxScenarioInformationTxAveRateBps 31
	pfGsxScenarioInformationIndQueNum 33
	pfGsxScenarioInformationAccelSessNum 34
	pfGsxScenarioInformationAccelBypassStatus 35
	pfGsxScenarioInformationAccelActivePeer 36

ScenarioSortIndex: The sort number of the scenario. The length is 16. Use the same way as sort number acquisition in pfGsxScenarioStatisticsTable.

For pfGsxScenarioStatByScIdTable,
the OID of pfGsxScenarioStatByScIdEntry is as follows:
1.3.6.1.4.1.1151.2.1.7.11.1.1.EntryOID.ScenarioId

Fixed value

Entry OID: Entry number in the table. Entries appear in the order defined in Table 4. The length is 1.

pfGsxScenarioStatByScIdScenarioId	1
pfGsxScenarioStatByScIdScenarioName	2
pfGsxScenarioStatByScIdScenarioType	3
pfGsxScenarioStatByScIdRxOctets	4
pfGsxScenarioStatByScIdRxPackets	5
pfGsxScenarioStatByScIdTxOctets	6
pfGsxScenarioStatByScIdTxPackets	7
pfGsxScenarioStatByScIdDiscardOctets	8
pfGsxScenarioStatByScIdDiscardPackets	9
pfGsxScenarioStatByScIdHCRxOctets	10
pfGsxScenarioStatByScIdHCRxPackets	11
pfGsxScenarioStatByScIdHCTxOctets	12
pfGsxScenarioStatByScIdHCTxPackets	13
pfGsxScenarioStatByScIdHCDiscardOctets	14
pfGsxScenarioStatByScIdHCDiscardPackets	15
pfGsxScenarioStatByScIdDefaultQueRxOctets	16
pfGsxScenarioStatByScIdDefaultQueRxPackets	17
pfGsxScenarioStatByScIdDefaultQueTxOctets	18
pfGsxScenarioStatByScIdDefaultQueTxPackets	19
pfGsxScenarioStatByScIdDefaultQueDiscardOctets	20
pfGsxScenarioStatByScIdDefaultQueDiscardPackets	21
pfGsxScenarioStatByScIdDefaultQueHCRxOctets	22
pfGsxScenarioStatByScIdDefaultQueHCRxPackets	23
pfGsxScenarioStatByScIdDefaultQueHCTxOctets	24
pfGsxScenarioStatByScIdDefaultQueHCTxPackets	25
pfGsxScenarioStatByScIdDefaultQueHCDiscardOctets	26
pfGsxScenarioStatByScIdDefaultQueHCDiscardPackets	27

ScenarioId: Scenario ID of the scenario The length is 1. The scenario ID is specified when the scenario is registered.

If no ID is specified for the scenario at registration, an ID is automatically assigned to the scenario. In this case, run the "show scenario name" command to confirm the assigned scenario ID.

For pfGsxScenarioInfoByScIdTable,
the OID of pfGsxScenarioInfoByScIdEntry is as follows:
1.3.6.1.4.1.1151.2.1.7.12.1.1.EntryOID.ScenarioId

Fixed value

Entry ID: Entry number in the table. Note that numbers are not sequential. The length is 1.

pfGsxScenarioInfoByScIdScenarioId	1
pfGsxScenarioInfoByScIdScenarioName	2
pfGsxScenarioInfoByScIdScenarioType	3
pfGsxScenarioInfoByScIdDefFlowNum	4
pfGsxScenarioInfoByScIdDefBuffRatio	25
pfGsxScenarioInfoByScIdDefBuff	26
pfGsxScenarioInfoByScIdDefPeakBuffRatio	27
pfGsxScenarioInfoByScIdDefPeakBuff	28
pfGsxScenarioInfoByScIdTxPeakRateBps	29
pfGsxScenarioInfoByScIdTxAveRateBps	31
pfGsxScenarioInfoByScIdIndQueNum	33
pfGsxScenarioInfoByScIdAccelSessNum	34
pfGsxScenarioInfoByScIdAccelBypassStatus	35
pfGsxScenarioInfoByScIdAccelActivePeer	36

ScenarioId: Scenario ID of the scenario. The length is 1. Use the same way as scenario ID acquisition in pfGsxScenarioStatByScIdTable.

(Blank page)

Appendix E JSON Format

This appendix describes the JSON (JavaScript Object Notation:RFC4627) description format.

JSON is a simple, text-based data description language defined by RFC4627.

JSON has 4 primitives and 2 structured objects. The WebAPI of this device uses a string primitive and an object structure only.

	Type	Example	Description
Primitives	string	"PureFlow"	Character string
	number	123	Numerical value
	boolean	true	Indicates true or false.
	null	null	Indicates no value.
Structures	object	{name:value}	An array of pairs of a name and a value (or no pair).
	array	[value, value]	An array of values (or no value)

The following description is based on the API for adding scenarios described in Appendix F "Details of WebAPI".

API	Key	Value	Relevant CLI command and parameter
Add a scenario (Discard)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"discard"	action discard
	"scenario_id" (Optional)	Scenario ID	[scenario <scenario_id>]

Make a key and value pair delimited by a colon.

```

"command": "add scenario"
"scenario_name": "/port1/North"
"action": "discard"
"scenario_id": "1"
```

The scenario ID can be omitted if not required .

```
“command”：“add scenario”  
“scenario_name”：“/port1/North”  
“action”：“discard”
```

Connect these three parameters with commas (.). Do not add a comma to the last parameter.

```
“command”：“add scenario”, “scenario_name”：“/port1/North”, “action”：“discard”
```

Finally, enclose them in curly brackets ({}) to make an object structure.

```
{“command”：“add scenario”, “scenario_name”：“/port1/North”, “action”：“discard”}
```

For better syntax reading, you can add a half-width space, tab, or line break before and after curly brackets, colons, and commas.

```
{  
    “command” : “add scenario”,  
    “scenario_name” : “/port1/North”,  
    “action” : “discard”  
}
```

Parameters for the WebAPI of this device can be in random order. They need not be consistent with the order described in Appendix F “Details of WebAPI”.

```
{  
    “action” : “discard”,  
    “scenario_name” : “/port1/North”,  
    “command” : “add scenario”  
}
```

Appendix F Details of WebAPI

This appendix describes details of the WebAPI of this device.

For the WebAPI, provide JSON data for the following URL:

http://IP address of the system interface/shapermng/json

To use HTTPS (Hypertext Transfer Secure), specify "https" at the beginning of the URL.

https://<System interface IP address>/shapermng/json

Keys and values should be in lowercase. Optional parameters can be omitted if they need not be specified. If a key is wrongly spelled, the parameter is ignored. Required parameters can cause errors if they are spelled wrongly, but wrongly spelled optional parameters and undefined parameters do not cause an error.

For details of the values to specify, see “Command Reference (PureFlow WSX Unified Network Controller: NF7600 series)”.

API	Key	Value	Relevant CLI command and parameter
Add a scenario (Discard)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"discard"	action discard
	"scenario_id" (Optional)	Scenario ID	[scenario <scenario_id>]
Add a scenario (Aggregate)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"aggregate"	action aggregate
	"cos" (Optional)	Cos value	[cos <user_priority>]
	"inner-cos" (Optional)	Inner-Cos value	[inner-cos <user_priority>]
	"dscp" (Optional)	dscp	[dscp <dscp>]
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]

API	Key	Value	Relevant CLI command and parameter
Add a scenario (Aggregate) (Continued)	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
	"scenario_id" (Optional)	Scenario ID	[scenario <scenario_id>]
Add a scenario (Individual)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"individual"	action individual
	"cos" (Optional)	Cos value	[cos <user_priority>]
	"inner-cos" (Optional)	Inner-Cos value	[inner-cos <user_priority>]
	"dscp" (Optional)	dscp	[dscp <dscp>]
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
	"scenario_id" (Optional)	Scenario ID	[scenario <scenario_id>]
	"maxqnum" (Optional)	Maximum number of individual queues	[maxquenum <quenum>]
	"quedivision" (Optional)	Individual queue division target	[quedivision <field>]
	"failaction" (Optional)	Action in case the maximum number of individual queues is exceeded	[failaction <discard forwardbesteffort forwardattribute>]
	"fail_min_bw" (Optional)	Minimum bandwidth in case the maximum number of individual queues is exceeded	[fail_min_bw <min_bandwidth>]
	"fail_peak_bw" (Optional)	Peak bandwidth in case the maximum number of individual queues is exceeded	[fail_peak_bw <peak_bandwidth>]
"fail_class" (Optional)	Class in case the maximum number of queues is exceeded.	[fail_class <class>]	

API	Key	Value	Relevant CLI command and parameter
Add a scenario (Wan-accel)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"wan-accel"	action wan-accel
	"peer" (Required)	IP address	peer <IP_address>
	"second_peer" (Optional)	IP address	[second-peer <IP_address >]
	"dport" (Optional)	Destination port number	[dport <port>]
	"vid" (Optional)	VLAN ID	[vid <VID>]
	"inner vid" (Optional)	Inner-VLAN ID	[inner-vid <VID>]
	"cos" (Optional)	Cos value	[cos <user_priority>]
	"inner-cos" (Optional)	Inner-Cos value	[inner-cos <user_priority>]
	"dscp" (Optional)	dscp	[dscp <dscp>]
	"compression" (Optional)	Compress “enable”: Enables compression. “disable”: Disables compression. When omitted, “disable” is applied.	[compression {enable disable}]
"tcp_mem" (Optional)	TCP buffer size	[tcp-mem {auto <size>}]	

API	Key	Value	Relevant CLI command and parameter
Add a scenario (Wan-accel) (Continued)	"cc_mode" (Optional)	Congestion control mode "normal": Changes the congestion control mode to the normal mode. "semi-fast": Changes the congestion control mode to the high-speed mode. "fast": Changes the congestion control mode to the high-speed mode. When omitted, "normal" is applied.	[cc-mode {normal semi-fast fast}]
	"bypass_thresh" (Optional)	RTT	[bypass-thresh <rtt>]
	"bypass_keepalive" (Optional)	keepalive of auto bypass function "enable": Enables keepalive. "disable": Disables keepalive. When omitted, "disable" is applied.	[bypass-keepalive {enable disable}]
	"fec" (Optional)	FEC "enable": Enables the FEC function. "disable": Disables the FEC function. When omitted, "disable" is applied.	[fec {enable disable}]
	"block_size" (Optional)	FEC block size	[block-size <size>]
	"data_block_size" (Optional)	FEC data block size	[data-block-size <size>]
	"fec_session" (Optional)	FEC session count	[fec-session <session>]
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Maximum bandwidth	[peak_bw <peak_bandwidth>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]

API	Key	Value	Relevant CLI command and parameter
Add a scenario (Aggregate)	"command" (Required)	"update scenario"	update scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"aggregate"	action aggregate
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
Update a scenario (Individual)	"command" (Required)	"update scenario"	update scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"individual"	action individual
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
	"maxqnum" (Optional)	Maximum number of individual queues	[maxquenum <quenum>]
	"quedivision" (Optional)	Individual queue division target	[quedivision <field>]
	"failaction" (Optional)	Action in case the maximum number of individual queues is exceeded	[failaction <discard forwardbesteffort forwardattribute>]
	"fail_min_bw" (Optional)	Minimum bandwidth in case the maximum number of individual queues is exceeded	[fail_min_bw <min_bandwidth>]
	"fail_peak_bw" (Optional)	Peak bandwidth in case the maximum number of individual queues is exceeded	[fail_peak_bw <peak_bandwidth>]
	"fail_class" (Optional)	Class in case the maximum number of queues is exceeded.	[fail_class <class>]

API	Key	Value	Relevant CLI command and parameter
Update a scenario (Wan-accel)	"command" (Required)	"update scenario"	update scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"wan-accel"	action wan-accel
	"compression" (Optional)	Compress "enable": Enables compression. "disable": Disables compression. When omitted, "disable" is applied.	[compression {enable disable}]
	"tcp_mem" (Optional)	TCP buffer size	[tcp-mem {auto <size>}]
	"cc_mode" (Optional)	Congestion control mode "normal": Changes the congestion control mode to the normal mode. "semi-fast": Changes the congestion control mode to the high-speed mode. "fast": Changes the congestion control mode to the high-speed mode. When omitted, "normal" is applied.	[cc-mode {normal semi-fast fast}]
	"bypass_thresh" (Optional)	RTT	[bypass-thresh <rtt>]
"bypass_keepalive" (Optional)	keepalive of auto bypass function "enable": Enables keepalive. "disable": Disables keepalive. When omitted, "disable" is applied.	[bypass-keepalive {enable disable}]	

API	Key	Value	Relevant CLI command and parameter
Update a scenario (Wan-accel) (Continued)	"fec" (Optional)	FEC "enable": Enables the FEC function. "disable": Disables the FEC function. When omitted, "disable" is applied.	[fec {enable disable}]
	"block_size" (Optional)	FEC block size	[block-size <size>]
	"data_block_size" (Optional)	FEC data block size	[data-block-size <size>]
	"fec_session" (Optional)	FEC session count	[fec-session <session>]
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Maximum bandwidth	[peak_bw <peak_bandwidth>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
Delete a scenario (all specified)	"command" (Required)	"delete scenario"	delete scenario
	"scenario_name" (Required)	"all"	all
Delete a scenario (specified scenario)	"command" (Required)	"delete scenario"	delete scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"recursive" (Optional)	"recursive"	[recursive]
Get scenario information	"command" (Required)	"show scenario"	show scenario
	"scenario_name" (Required)	Scenario name	name <scenario_name>
	"search_type" (Optional)	How to get "exact": Gets information of the specified scenario. "next": Gets information of the scenario next to the specified scenario. When omitted or the value is incorrectly spelled, "exact" is applied.	None

API for getting scenario information

The API for getting scenario information provides the "search_type" parameter. Specify "exact" or "next" for "search_type".

"exact": Gets information of the scenario specified by "scenario_name".

"next": Gets information of the scenario next to the scenario specified by "scenario_name". Information to be retrieved is in the scenario tree order in the same way the CLI command "show scenario".

When "search_type" is omitted, "exact" is applied.

To get information of a specific scenario, specify a scenario name and "exact".

To get information of all scenarios in the same way as the CLI command "show scenario all", specify "next" and follow the procedure below.

For the first scenario, specify nothing for "scenario_name".

```
"scenario_name" : "" (empty string)
```

```
"search_type" : "next"
```

This gets information of the scenario "/port1" heading the scenario tree.

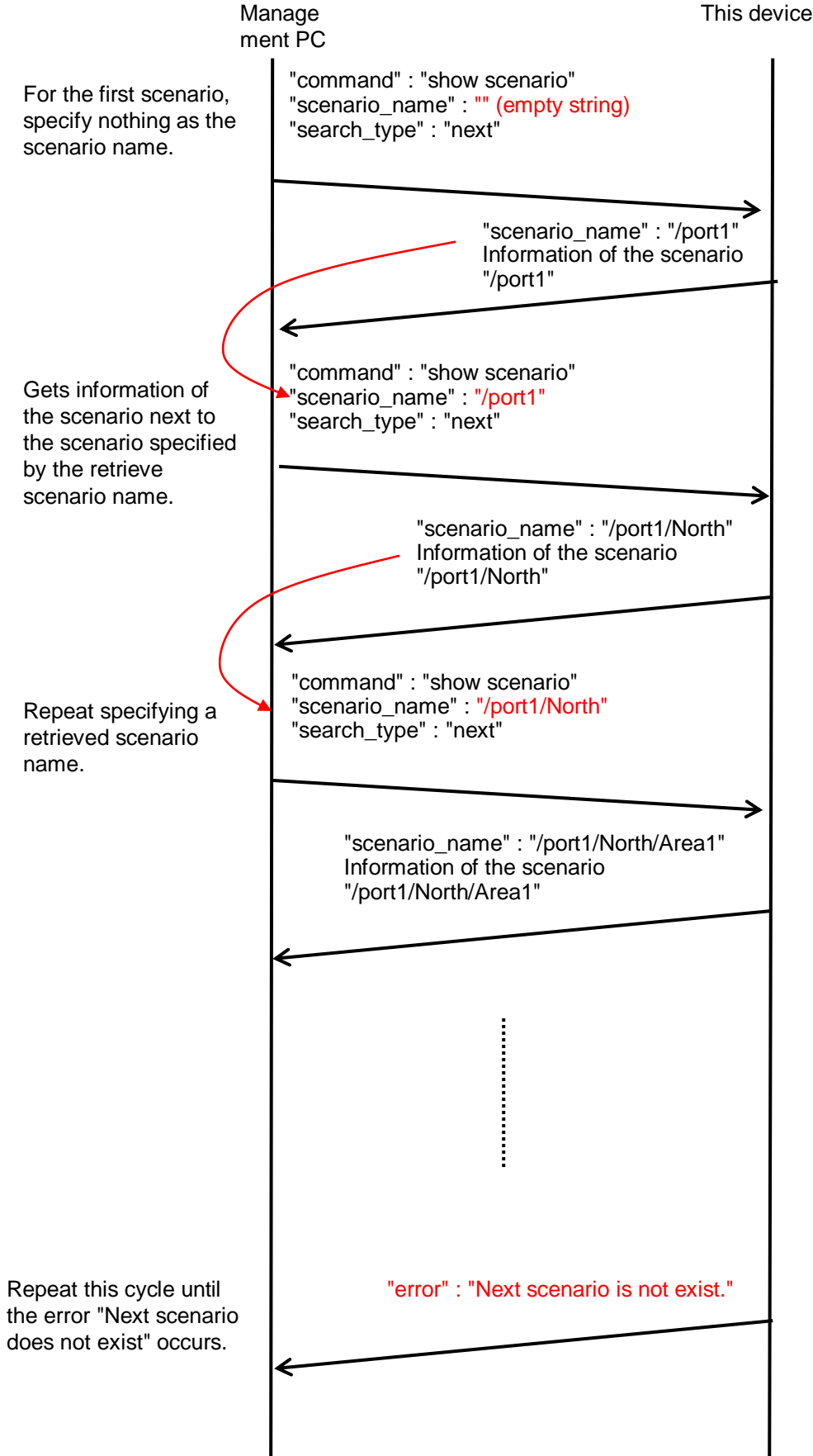
Then, specify the name of the retrieved scenario for "scenario_name".

```
"scenario_name" : "/port1"
```

```
"search_type" : "next"
```

This gets information of the scenario next to "/port1" in the scenario tree.

Repeat this cycle (specify the retrieved scenario name and "next") to get further information. If you specify the name of the last scenario in the scenario tree and specify "next", the error message "Next scenario does not exist" will appear.



API	Key	Value	Relevant CLI command and parameter
Add a filter (Bridge-ctrl)	"command" (Required)	"add filter"	add filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"type" (Required)	"bridge-ctrl"	bridge-ctrl
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
Add a filter (Ethernet)	"command" (Required)	"add filter"	add filter
	"scenario_name" (Required)	Filter name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"type" (Required)	"ethernet"	Ethernet
	"vid" (Optional)	VLAN ID	[vid {<VID> none}]
	"cos" (Optional)	CoS	[cos <user_priority>]
	"inner-vid" (Optional)	Inner-VLAN ID	[inner-vid {<VID> none}]
	"inner-cos" (Optional)	Inner-CoS	[inner-cos <user_priority>]
	"ethertype" (Optional)	Ethernet Type/Length	[ethertype <type>]
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
Add a filter (IPv4)	"command" (Required)	"add filter"	add filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"type" (Required)	"ipv4"	ipv4
	"vid" (Optional)	VLAN ID	[vid {<VID> none}]
	"cos" (Optional)	CoS	[cos <user_priority>]

API	Key	Value	Relevant CLI command and parameter
Add a filter (IPv4) (Continued)	"inner-vid" (Optional)	Inner-VLAN ID	[inner-vid {<VID> none}]
	"inner-cos" (Optional)	Inner-CoS	[inner-cos <user_priority>]
	"sip" or "sip list" (Optional)	Source IPv4 address or Rule list name If "sip" and "sip list" are used at the same time, "sip list" is prioritized.	[sip [list] {<src_IP_address> <list_name>}]
	"dip" or "dip list" (Optional)	Destination IPv4 address or Rule list name If "dip" and "dip list" are used at the same time, "dip list" is prioritized.	[dip [list] {<dst_IP_address> <list_name>}]
	"tos" (Optional)	ToS	[tos <type_of_service>]
	"proto" (Optional)	Protocol number	[proto <protocol>]
	"sport" or "sport list" (Optional)	Source port number or Rule list name If "sport" and "sport list" are used at the same time, "sport list" is prioritized.	[sport [list] {<sport> <list_name>}]
	"dport" or "dport list" (Optional)	Destination port number or Rule list name If "dport" and "dport list" are used at the same time, "dport list" is prioritized.	[dport [list] {<dport> <list_name>}]
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
Add a filter (IPv6)	"command" (Required)	"add filter"	add filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"type" (Required)	"ipv6"	ipv6
	"vid" (Optional)	VLAN ID	[vid {<VID> none}]

API	Key	Value	Relevant CLI command and parameter
Add a filter (IPv6) (Continued)	"cos" (Optional)	CoS	[cos <user_priority>]
	"inner-vid" (Optional)	Inner-VLAN ID	[inner-vid {<VID> none}]
	"inner-cos" (Optional)	Inner-CoS	[inner-cos <user_priority>]
	"sip" or "sip list" (Optional)	Source IPv6 address or Rule list name If "sip" and "sip list" are used at the same time, "sip list" is prioritized.	[sip [list] {<src_IP_address> <list_name>}]
	"dip" or "dip list" (Optional)	Destination IPv6 address or Rule list name If "dip" and "dip list" are used at the same time, "dip list" is prioritized.	[dip [list] {<dst_IP_address> <list_name>}]
	"tos" (Optional)	ToS	[tos <type_of_service>]
	"proto" (Optional)	Protocol number	[proto <protocol>]
	"sport" or "sport list" (Optional)	Source port number or Rule list name If "sport" and "sport list" are used at the same time, "sport list" is prioritized.	[sport [list] {<sport> <list_name>}]
	"dport" or "dport list" (Optional)	Destination port number or Rule list name If "dport" and "dport list" are used at the same time, "dport list" is prioritized.	[dport [list] {<dport> <list_name>}]
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
Delete a filter (all specified)	"command" (Required)	"delete filter"	delete filter
	"scenario_name" (Required)	"all"	All
Delete a filter (scenario specified)	"command" (Required)	"delete filter"	delete filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>

API	Key	Value	Relevant CLI command and parameter
Delete a filter (filter specified)	"command" (Required)	"delete filter"	delete filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
Get scenario information	"command" (Required)	"show filter"	show filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"search_type" (Optional)	Acquisition method "exact": Acquires the information of the specified filter. "next": Acquires the information of the filter next to the specified filter. When omitted or the value is incorrectly spelled, "exact" is applied.	None

API for getting filter information

The API for getting filter information provides the "search_type" parameter. Specify "exact" or "next" for "search_type".

"exact": Acquires the information on the filter specified in "scenario_name" and "filter_name".

"next": Acquires the information on the filter next to the filter specified in "scenario_name" and "filter_name".

Acquire the scenarios in the alphabetical order of filter names, the same as the "show filter" CLI command. When the bottom filter of the scenario is specified, get the information on the filter at the head of the next scenario.

To acquire the specified filter information, specify the scenario name and filter name, and apply "exact" to acquire the scenario.

To acquire all the filter information on all the scenarios, the same as the "show filter all" in the CLI command, use "next". The acquisition procedure using "next" is the same as that of the scenario acquisition API.

API	Key	Value	Relevant CLI command and parameter
Add application acceleration	"command" (Required)	"add apl-accel"	add apl-accel
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"protocol " (Required)	Protocol name	protocol smb
	"tcp_port" (Optional)	TCP port number	[tcp <port>]
	"smb-session" (Optional)	Session count	[smb-session <session>]
	"read-attr" (Optional)	Substitute response for the SMB2 QUERY_INFO command in the reading operation "enable": Enables the substitute response for the SMB2 QUERY_INFO command "disable": Disables the substitute response for the SMB2 QUERY_INFO command When omitted, "enable" is applied.	[read-attr {enable disable}]
	"read-operation" (Optional)	Substitute response for the SMB2 READ command in the reading operation "enable": Enables the substitute response for the SMB2 READ command "disable": Disables the substitute response for the SMB2 READ command When omitted, "enable" is applied.	[read-operation {enable disable}]
"read-cache-size" (Optional)	Cache size of the substitute response in the reading operation	[read-cache-size <size>]	

API	Key	Value	Relevant CLI command and parameter
Add application acceleration (Continued)	"write-attr" (Optional)	Substitute response for the SMB2 QUERY_INFO command in the writing operation "enable": Enables the substitute response for the SMB2 QUERY_INFO command "disable": Disables the attribution substitute response for the SMB2 QUERY_INFO command When omitted, "enable" is applied.	[write-attr {enable disable}]
	"write-attr-1st" (Optional)	Substitute response for the SMB2 SET_INFO command before the writing operation "enable": Enables the substitute response for the SMB2 SET_INFO command "disable": Disables the substitute response for the SMB2 SET_INFO command When omitted, "disable" is applied.	[write-attr-1st {enable disable}]

API	Key	Value	Relevant CLI command and parameter
Add application acceleration (Continued)	"write-attr-2nd" (Optional)	Substitute response for the SMB2 SET_INFO command after the writing operation "enable": Enables the substitute response for the SMB2 SET_INFO command "disable": Disables the substitute response for the SMB2 SET_INFO command When omitted, "disable" is applied.	[write-attr-2nd {enable disable}]
	"write-operation" (Optional)	Substitute response for the SMB2 WRITE command in the writing operation "enable": Enables the substitute response for the SMB2 WRITE command "disable": Disables the substitute response for the SMB2 WRITE command When omitted, "enable" is applied.	[write-operation {enable disable}]
Update application acceleration	"command" (Required)	"update apl-accel"	update apl-accel
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"protocol " (Required)	Protocol name	protocol smb
	"tcp_port" (Optional)	TCP port number	[tcp <port>]
	"smb-session" (Optional)	Session count	[smb-session <session>]

API	Key	Value	Relevant CLI command and parameter
Update application acceleration (Continued)	"read-attr" (Optional)	Substitute response for the SMB2 QUERY_INFO command in the reading operation "enable": Enables the substitute response for the SMB2 QUERY_INFO command "disable": Disables the substitute response for the SMB2 QUERY_INFO command When omitted, "enable" is applied.	[read-attr {enable disable}]
	"read-operation" (Optional)	Substitute response for the SMB2 READ command in the reading operation "enable": Enables the substitute response for the SMB2 READ command "disable": Disables the substitute response for the SMB2 READ command When omitted, "enable" is applied.	[read-operation {enable disable}]
	"read-cache-size" (Optional)	Cache size of the substitute response in the reading operation	[read-cache-size <size>]

API	Key	Value	Relevant CLI command and parameter
Update application acceleration (Continued)	"write-attr" (Optional)	Substitute response for the SMB2 QUERY_INFO command in the writing operation "enable": Enables the substitute response for the SMB2 QUERY_INFO command "disable": Disables the attribution substitute response for the SMB2 QUERY_INFO command When omitted, "enable" is applied.	[write-attr {enable disable}]
	"write-attr-1st" (Optional)	Substitute response for the SMB2 SET_INFO command before the writing operation "enable": Enables the substitute response for the SMB2 SET_INFO command "disable": Disables the substitute response for the SMB2 SET_INFO command When omitted, "disable" is applied.	[write-attr-1st {enable disable}]

API	Key	Value	Relevant CLI command and parameter
Update application acceleration (Continued)	"write-attr-2nd" (Optional)	Substitute response for the SMB2 SET_INFO command after the writing operation "enable": Enables the substitute response for the SMB2 SET_INFO command "disable": Disables the substitute response for the SMB2 SET_INFO command When omitted, "disable" is applied.	[write-attr-2nd {enable disable}]
	"write-operation" (Optional)	Substitute response for the SMB2 WRITE command in the writing operation "enable": Enables the substitute response for the SMB2 WRITE command "disable": Disables the substitute response for the SMB2 WRITE command When omitted, "enable" is applied.	[write-operation {enable disable}]
Delete application acceleration	"command" (Required)	"delete apl-accel"	delete apl-accel
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"protocol " (Required)	Protocol name	protocol smb

API	Key	Value	Relevant CLI command and parameter
Add a rule list group	"command" (Required)	"add rulelist group"	add rulelist group
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	Rule list type	{ipv4 ipv6 l4port}
Delete a rule list group (all specified)	"command" (Required)	"delete rulelist group"	delete rulelist group
	"list_name" (Required)	"all"	all
Delete a rule list group (group specified)	"command" (Required)	"delete rulelist group"	delete rulelist group
	"list_name" (Required)	Rule list name	<list_name>
Add a rule list entry (IPv4)	"command" (Required)	"add rulelist entry"	add rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv4"	ipv4
	"IP_address" (Required)	IPv4 address	<IP_address>
Add a rule list entry (IPv6)	"command" (Required)	"add rulelist entry"	add rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv6"	ipv6
	"IP_address" (Required)	IPv6 address	<IP_address>
Add a rule list entry (L4Port)	"command" (Required)	"add rulelist entry"	add rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"l4port"	l4port
	"port" (Required)	L4 port number	<port>
Delete a rule list entry (all specified)	"command" (Required)	"delete rulelist entry"	delete rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"all"	all

API	Key	Value	Relevant CLI command and parameter
Delete a rule list entry (IPv4)	"command" (Required)	"delete rulelist entry"	delete rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv4"	ipv4
	"IP_address" (Required)	IPv4 address	<IP_address>
Delete a rule list entry (IPv6)	"command" (Required)	"delete rulelist entry"	delete rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv6"	ipv6
	"IP_address" (Required)	IPv6 address	<IP_address>
Delete a rule list entry (L4Port)	"command" (Required)	"delete rulelist entry"	delete rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"l4port"	l4port
	"port" (Required)	L4 port number	<port>
Get rule list information	"command" (Required)	"show rulelist"	show rulelist
	"list_name" (Required)	Rule list name	[<list_name>]
	"rules" (Required)	Delete a rule list	None
	"search_type" (Optional)	How to get "exact": Gets the specified rule list entry. "next": Gets the rule list entry next to the specified one. When omitted or the value is incorrectly spelled, "exact" is applied.	None

API for getting rule list information

The API for getting rule list information provides the "rules" parameter which is not available for CLI command "show rulelist".

Specify a rule list entry (IP address or L4 port number) as a value for "rules". Even for a single value, use a hyphen to specify a range value.

IPv4 address 192.168.1.1-192.168.1.1

IPv6 address FE80::0001-FE80::0001

L4 port number 1000-1000

For rule lists for which no rule list entry set, "none" is retrieved.

Specify "exact" or "next" for "search_type" to specify the retrieving method.

"exact": Gets the rule list entry specified by "list_name" and "rules".

"next": Gets the rule list entry next to the one specified by "list_name" and "rules".

Information to be retrieved is in the same order as the CLI command "show rulelist".

If the last rule list entry of the rule list is specified, information of the first rule list entry of the next rule list is retrieved.

To get information of a specific rule list entry, specify a rule list name and a rule list entry, and then specify "exact".

To get information of all rule list entries of all rule lists in the same way as the CLI command "show rulelist all", use "next". The procedure for using "next" is the same as for the API for getting scenario information.

API	Key	Value	Relevant CLI command and parameter
Add channel (Normal channel)	"command" (Required)	"add channel"	add channel
	"channel_name" (Required)	Channel name	<channel_name>
	"lan" (Required)	Lan-side port number or Port group	lan {<slot/port> <group_name>}
	"wan" (Required)	Wan-side port number or Port group	wan {<slot/port> <group_name>}
	"channel_type" (Required)	Channel type "normal": Adds a normal channel. "default": Add the default channel.	None
	"vid" (Required)	VLAN ID	vid {<VID> none}
	"inner_vid" (Optional)	Inner-VLAN ID	[inner-vid {<VID> none}]
	"tpid" (Optional)	tpid	[tpid <tpid>]
	"inner_tpid" (Optional)	inner-tpid	[inner-tpid <tpid>]
	"mtu" (Optional)	mtu	[mtu <mtu>]
Add channel (Default channel)	"command" (Required)	"add channel"	add channel
	"channel_name" (Required)	Channel name	<channel_name>
	"lan" (Required)	Lan-side port number or Port group	lan {<slot/port> <group_name>}
	"wan" (Required)	Wan-side port number or Port group	wan {<slot/port> <group_name>}
	"channel_type" (Required)	Channel type "normal": Adds a normal channel. "default": Add the default channel.	None

API	Key	Value	Relevant CLI command and parameter
Delete channel (all specified)	"command" (Required)	"delete channel"	delete channel
	"channel_name" (Required)	"all"	all
Delete channel (channel name specified)	"command" (Required)	"delete channel"	delete channel
	"channel_name" (Required)	Channel name	<channel_name>
Show channel information	"command" (Required)	"show channel"	show channel
	"channel_name" (Required)	Channel name	name <channel_name>
	"search_type" (Optional)	Acquisition method "exact": Acquires the specified channel. "next": Acquires the channel next to the specified channel. When omitted or the value is incorrectly spelled, "exact" is applied.	None

API for getting channel information

The API for getting channel information provides the "search_type" parameter that specifies the acquisition method. Specify "exact" or "next" as a value for "search_type".

"exact": Gets channel information specified by "channel_name".

"next": Gets channel information next to the one specified by "channel_name". The acquisition order is the alphabetical order of the channel name the same as the "show channel" CLI command.

To get information of a specific channel, specify the channel name, and then specify "exact". To get information of all channels in the same way as the CLI command "show channel all", specify "next". The acquisition procedure using "next" is the same as that of the scenario acquisition API.

API	Key	Value	Relevant CLI command and parameter
Interface setting	"command" (Required)	"set ip channel"	set ip channel
	"channel_name" (Required)	Channel name	<channel_name>
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"netmask" (Required)	IPv4 net mask or IPv6 prefix	netmask <netmask>
Release interface (all specified)	"command" (Required)	"unset ip channel"	unset ip channel
	"channel_name" (Required)	"all"	all
Release interface (channel name specified)	"command" (Required)	"unset ip channel"	unset ip channel
	"channel_name" (Required)	Channel name	<channel_name>
	"type" (Optional)	Target of release setting "ipv4": Releases the IPv4 channel interface setting. "ipv6": Releases the IPv6 channel interface setting. When omitted, Releases both IPv4 and IPv6 channel interface settings.	{ipv4 ipv6}
Show interface information	"command" (Required)	"show ip channel"	show ip channel
	"channel_name" (Required)	Channel name	name <channel_name>
	"search_type" (Optional)	Acquisition method "exact": Acquires the specified channel. "next": Acquires the channel next to the specified channel. When omitted or the value is incorrectly spelled, "exact" is applied.	None

API for getting interface information

The API for getting interface information provides the "search_type" parameter that specifies the acquisition method. Specify "exact" or "next" as a value for "search_type".

"exact": Gets interface information specified by "channel_name".

"next": Gets interface information next to the one specified by "channel_name". The acquisition order is the alphabetical order of the channel name the same as the "show ip interface" CLI command.

To get information of a specific interface, specify the interface name, and then specify "exact".

To acquire information of all interfaces in the same manner as the CLI command "show ip interface all", specify "next". The acquisition procedure using "next" is the same as that of the scenario acquisition API.

API	Key	Value	Relevant CLI command and parameter
Add static path (Specifying destination)	"command" (Required)	"add route"	add route
	"route_type" (Required)	target	target
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"netmask" (Required)	IPv4 net mask or IPv6 prefix	netmask <netmask>
	"gateway" (Required)	IPv4 address or IPv6 address	gateway <gateway>
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The transmission Network port is located on the LAN side. "wan":The transmission Network port is located on the WAN side.	{lan wan}
Add static path (default gateway)	"command" (Required)	"add route"	add route
	"route_type" (Required)	default	default
	"gateway" (Required)	IPv4 address or IPv6 address	gateway <gateway>
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The transmission Network port is located on the LAN side. "wan":The transmission Network port is located on the WAN side.	{lan wan}
Delete static path (all specified)	"command" (Required)	"delete route"	delete route
	"route_type" (Required)	all	all

API	Key	Value	Relevant CLI command and parameter
Delete static path (Specifying destination))	"command" (Required)	"delete route"	delete route
	"route_type" (Required)	target	target
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"netmask" (Required)	IPv4 net mask or IPv6 prefix	netmask <netmask>
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The transmission Network port is located on the LAN side. "wan":The transmission Network port is located on the WAN side.	{lan wan}
Delete static path (default gateway)	"command" (Required)	"delete route"	delete route
	"route_type" (Required)	default	default
	"gateway" (Required)	IPv4 address or IPv6 address	gateway <gateway>
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The transmission Network port is located on the LAN side. "wan":The transmission Network port is located on the WAN side.	{lan wan}
	"type" (Optional)	Target of release setting "ipv4":Deletes the IPv4 static path information setting. "ipv6":Deletes the IPv6 static path information setting. When omitted, Deletes both IPv4 and IPv6 static path information settings.	{ipv4 ipv6}

API	Key	Value	Relevant CLI command and parameter
Show static path information (Specifying destination))	"command" (Required)	"show route target"	show route target
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"netmask" (Required)	IPv4 net mask or IPv6 prefix	netmask <netmask>
	"gateway" (Required)	IPv4 address or IPv6 address	gateway <gateway>
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The transmission Network port is located on the LAN side. "wan": The transmission Network port is located on the WAN side.	{lan wan}
	"search_type" (Optional)	Acquisition method "exact": Acquires the specified channel. "next": Acquires the channel next to the specified channel. When omitted or the value is incorrectly spelled, "exact" is applied.	None

API for getting path information

The API for getting path information provides the "search_type" parameter that specifies the acquisition method. Specify "exact" or "next" as a value for "search_type".

"exact": Gets path information that matches all of the input parameters.

"next": Gets path information next to the one that matches all of the input parameters. The acquisition order is the entry order of the channel name the same as the "show route" CLI command.

To get information of a specific path, specify information on all of the paths, and then specify "exact".

To get information of all paths in the same way as the CLI command "show route all", specify "next". The acquisition procedure using "next" is the same as that of the scenario acquisition API.

API	Key	Value	Relevant CLI command and parameter
Add a OpenFlow controller	"command" (Required)	"add openflow controller"	add openflow controller
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"tcp" (Required)	TCP port number	[tcp <port>]
Delete a OpenFlow controller	"command" (Required)	"delete openflow controller"	delete openflow controller
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
Get OpenFlow controller information	"command" (Required)	"show openflow controller"	show openflow controller

API	Key	Value	Relevant CLI command and parameter
Saves configuration	"command" (Required)	"save config"	save config
Gets the execution status of the save configuration command	"command" (Required)	"show save status"	None

API for saving configuration

The API for saving configuration terminates without waiting for the completion of the save. The configuration save is executed in the background. It returns the error message, "configuration save is in progress" when it is further instructed to save a configuration by using this API when a save command execution is in progress. For details about the time required for saving a configuration, see Chapter 3 "Configuring Settings".

API for getting the execution status of the save configuration command

There is no relevant CLI command. This API gets the execution status of the save configuration command.

"configuration save is in progress": The save configuration command execution is in progress.

"configuration save is not in progress": The save configuration command execution is completed.

Appendix G WebAPI Sample Programs

Python is a widely used programming language for Web API. Python provides HTTP and JSON libraries, and suits the WebAPI of this device.

This appendix shows sample programs for WebAPI features described in Appendix F using Python version 2.7.2.

Setting information

Use an “add” type API to add a setting, “update” to update a setting, and “delete” to delete a setting.

The “add”, “update”, “set”, and “delete” type APIs send commands and parameters, and receive responses. This section describes the “add scenario” command as common behavior in the API.

- 1 A sample program for single setting.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
params = {
    'command': 'add scenario',
    'scenario_name': '/port1/North',
    'action': 'aggregate',
    'min_bandwidth': '100M',
    'peak_bandwidth': '1G',
    'bufsize': '1M'
}
json_data = json.dumps(params)

# POST request
response = urllib2.urlopen(url, json_data)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '-----'
print data
print
```

The urlopen of Python returns after the HTTP request has transmitted, and process the session termination with this device in back ground. Therefore, when several urlopen called, the previous session may not been terminated at this device on the next urlopen. If this operation is repeated, resource of session will be run out at this device, and WebAPI will be unavailable temporarily.

To run several APIs continuously, please program to keep HTTP connection for several APIs using the HTTP persistent connection. The following describes a sample program using the HTTP persistent connection.

2 A sample program keeping the connection for several settings.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL IP address and file name of WebAPI
ip    = '192.168.1.1'
file  = 'shapermng/json'

# Open connection HTTP
conn = httplib.HTTPConnection(ip)
# Open connection HTTPS
#conn = httplib.HTTPSConnection(ip)

# Define parameters.
params = {
    'command': 'add scenario',
    'scenario_name': '/port1/North',
    'action': 'aggregate',
    'min_bandwidth': '100M',
    'peak_bandwidth': '1G',
    'bufsize': '1M'
}
json_data = json.dumps(params)

# POST request
conn.request("POST", '/' + file, json_data)
response = conn.getresponse()

# Display the response.
print 'RESPONSE:', response

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA      : '
print '-----'
print data
print

# Close connection.
conn.close()
```

Saving the configuration

When modifying the configuration is completed, use the API for saving the configuration to save the configuration changes.

The API for saving the configuration sends a command and receives a response to confirm the result.

The API for saving the configuration responds before completing the saving operation, which is running in the background. When this API tries to save a configuration while another configuration is being saved, the error message “configuration save is in progress” is returned. In this case, wait for a while, and retry saving. For the time required to save a configuration, see Chapter 3 “Configuring Settings”.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
params = {
    'command': 'save config'
}
json_data = json.dumps(params)

# POST request
response = urllib2.urlopen(url, json_data)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '-----'
print data
print
```

Getting the running status of configuration saving

This API gets whether the configuration is being saved.

This API returns the following in the response:

“configuration save is in progress”: The configuration is being saved.

“configuration save is not in progress”: The configuration has been saved.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
params = {
    'command': 'show save status'
}

# Encode the URL.
params_url = urllib.urlencode(params)

# GET request
response = urllib2.urlopen(url+'?' + params_url)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '-----'
print data
print
```

Displaying information

Use the “show” type API to view the set contents.

The “show” type API sends commands and parameters, and receives responses and shows data.

You need a different programming method to get a single entry only or all entries. A sample source code for each API is shown below.

(1) Getting channel information (specified channel)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# Specify “exact” for “search_type”.
params = {
    'command': 'show channel',
    'channel_name': 'dc_tokyo',
    'search_type': 'exact'
}

# Encode the URL.
params_url = urllib.urlencode(params)

# GET request
response = urllib2.urlopen(url+'?' + params_url)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '-----'
print data
print
```

(2) Getting channel information (all channels)

<1> A sample program keeping the connection to get all channel information.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL IP address and file name of WebAPI
ip    = '192.168.1.1'
file  = 'shapermng/json'

# Open connection. HTTP
conn = httplib.HTTPConnection(ip)
# Open connection. HTTPS
#conn = httplib.HTTPSConnection(ip)

# Define parameters.
# To display all channels, specify an empty string for the first channel name.
# Specify "next" for "search_type".
params = {
    'command': 'show channel',
    'channel_name': '',
    'search_type': 'next'
}

while 1:
    # Encode the URL.
    params_url    = urllib.urlencode(params)

    # GET request
    conn.request("GET", '/' + file + '?' + params_url)
    response = conn.getresponse()

    # Display the response.
    print 'RESPONSE:', response

    data = response.read()
    print 'LENGTH  :', len(data)
    print 'DATA      :'
    print '-----'
    print data
    print
```

(continued)

```
# From the data part of the response (JSON format string)
# get Python dictionary data.
json_data = json.loads(data)

# Exit if no channel name exists as JSON key.
if json_data.has_key("channel_name")==False:
    break

# Get a channel name.
channel_name = json_data['channel_name']

# Update the channel name to the retrieved one, and continue.
params['channel_name'] = channel_name

# Close connection.
conn.close()
```

<2> A sample program open and close the connection every time when acquiring a channel

When the following sample program is used, resource of session may be run out at this device and `urlopen` may fail depending on the performance of the terminal. If `urlopen` fails, use the sample program which gets all scenarios while keeping the connection of <1>.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# To display all channels, specify an empty string for the first channel name.
# Specify "next" for "search_type".
params = {
    'command': 'show channel',
    'channel_name': '',
    'search_type': 'next'
}

while 1:
    # Encode the URL.
    params_url = urllib.urlencode(params)

    # GET request
    response = urllib2.urlopen(url+'?' +params_url)

    # Display the response.
    print 'RESPONSE:', response
    print 'URL      :', response.geturl()

    data = response.read()
    print 'LENGTH :', len(data)
    print 'DATA   :'
    print '-----'
    print data
    print

    # From the data part of the response (JSON format string)
    # get Python dictionary data.
    json_data = json.loads(data)

    # Exit if no channel name exists as JSON key.
    if json_data.has_key("channel_name")==False:
        break
    # Get a channel name.
    channel_name = json_data['channel_name']

    # Update the channel name to the retrieved one, and continue.
    params['channel_name'] = channel_name
```


(3) Getting IP interface information (specified channel)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# Specify "exact" for "search_type".
params = {
    'command': 'show ip interface',
    'channel_name': 'dc_tokyo',
    'search_type': 'exact'
}

# Encode the URL.
params_url = urllib.urlencode(params)

# GET request
response = urllib2.urlopen(url+'?' + params_url)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    : '
print '-----'
print data
print
```

(4) Getting IP interface information (all channels)

<1> A sample program to get information on the IP interface of all of the channels while keeping the connection

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL IP address and file name of WebAPI
ip    = '192.168.1.1'
file  = 'shapermng/json'

# Open connection. HTTP
conn = httplib.HTTPConnection(ip)
# Open connection. HTTPS
#conn = httplib.HTTPSConnection(ip)

# Define parameters.
# To display all IP interfaces, specify an empty string for the first channel name.
# Specify "next" for "search_type".
params = {
    'command': 'show ip interface',
    'channel_name': '',
    'search_type': 'next'
}

while 1:
    # Encode the URL.
    params_url    = urllib.urlencode(params)

    # GET request
    conn.request("GET", '/' + file + '?' + params_url)
    response = conn.getresponse()

    # Display the response.
    print 'RESPONSE:', response
    data = response.read()
    print 'LENGTH   :', len(data)
    print 'DATA       :'
    print '-----'
    print data
    print
```

(continued)

```
# From the data part of the response (JSON format string)
# get Python dictionary data.
json_data = json.loads(data)

# Exit if no channel name exists as JSON key.
if json_data.has_key("channel_name")==False:
    break

# Get a channel name.
channel_name = json_data['channel_name']

# Update the channel name to the retrieved one, and continue.
params['channel_name'] = channel_name

# Close connection.
conn.close()
```

<2> A sample program open and close the connection every time when acquiring information on the IP interface of a channel

When the following sample program is used, resource of session may be run out at this device and urlopen may fail depending on the performance of the terminal. If urlopen fails, use the sample program which gets all scenarios while keeping the connection of <1>.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# To display all channels, specify an empty string for the first channel name.
# Specify "next" for "search_type".
params = {
    'command': 'show ip interface',
    'channel_name': '',
    'search_type': 'next'
}

while 1:
    # Encode the URL.
    params_url = urllib.urlencode(params)

    # GET request
    response = urllib2.urlopen(url+'?' + params_url)

    # Display the response.
    print 'RESPONSE:', response
    print 'URL      :', response.geturl()

    data = response.read()
    print 'LENGTH  :', len(data)
    print 'DATA    :'
    print '-----'
    print data
    print

    # From the data part of the response (JSON format string)
    # get Python dictionary data.
    json_data = json.loads(data)

    # Exit if no channel name exists as JSON key.
    if json_data.has_key("channel_name")==False:
        break
    # Get a channel name.
    channel_name = json_data['channel_name']

    # Update the channel name to the retrieved one, and continue.
    params['channel_name'] = channel_name
```

(5) Getting static path information (specified destination)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# Specify "exact" for "search_type".
params = {
    'command': 'show route target',
    'IP_address': '192.168.100.0',
    'netmask': '255.255.255.0',
    'gateway': '192.168.100.1',
    'channel_name': 'ch1',
    'output_if': 'lan',
    'search_type': 'exact'
}

# Encode the URL.
params_url = urllib.urlencode(params)

# GET request
response = urllib2.urlopen(url+'?'+params_url)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    : '
print '-----'
print data
print
```

(6) Getting static path information (all destinations)

<1> A sample program to get all of the static path information while keeping the connection.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL IP address and file name of WebAPI
ip    = '192.168.1.1'
file  = 'shapermng/json'

# Open connection. HTTP
conn = httplib.HTTPConnection(ip)
# Open connection. HTTPS
#conn = httplib.HTTPSConnection(ip)

# Define parameters.
# To display information on all static paths, specify an empty string for all parameters.
# Specify "next" for "search_type".
params = {
    'command': 'show route target',
    'IP_address': '',
    'netmask': '',
    'gateway': '',
    'channel_name': '',
    'output_if': '',
    'search_type': 'next'
}

while 1:
    # Encode the URL.
    params_url    = urllib.urlencode(params)

    # GET request
    conn.request("GET", '/' + file + '?' + params_url)
    response = conn.getresponse()

    # Display the response.
    print 'RESPONSE:', response
    data = response.read()
    print 'LENGTH  :', len(data)
```

(continued)

```
print 'DATA      :'  
print '-----'  
print data  
print  
  
# From the data part of the response (JSON format string)  
# get Python dictionary data.  
json_data1 = json.loads(data)  
json_data2 = json.loads(data)  
json_data3 = json.loads(data)  
json_data4 = json.loads(data)  
json_data5 = json.loads(data)  
  
# Exit if no target IP exists as JSON key.  
if json_data1.has_key("target")==False:  
    break  
# Exit if no netmask exists as JSON key.  
if json_data2.has_key("netmask")==False:  
    break  
# Exit if no gateway exists as JSON key.  
if json_data3.has_key("gateway")==False:  
    break  
# Exit if no channel_name exists as JSON key.  
if json_data4.has_key("channel_name")==False:  
    break  
# Exit if no output_if exists as JSON key.  
if json_data5.has_key("output_if")==False:  
    break  
# Get target IP.  
IP_address = json_data1['target']  
# Get netmask.  
netmask = json_data2['netmask']  
# Get gateway.  
gateway = json_data3['gateway']  
# Get channel_name.  
channel_name = json_data4['channel_name']  
# Get output_if.  
output_if = json_data5['output_if']  
  
# Update target IP to the retrieved one, and continue.  
params['IP_address'] = IP_address  
# Update netmask to the retrieved one, and continue.  
params['netmask'] = netmask
```

(continued)

```
# Update gateway to the retrieved one, and continue.
params['gateway'] = gateway
# Update channel_name to the retrieved one, and continue.
params['channel_name'] = channel_name
# Update output_if to the retrieved one, and continue.
params['output_if'] = output_if
# Close connection.
conn.close()
```


<2> A sample program open and close the connection every time when acquiring the static path information of a channel.

When the following sample program is used, resource of session will be run out at this device and urlopen may fail depending on the performance of the terminal. If urlopen fails, use the sample program which gets all scenarios while keeping the connection of <1>.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url    = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url   = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# To display information on all static paths, specify an empty string for all parameters.
# Specify "next" for "search_type".
params = {
    'command': 'show route target',
    'IP_address': "",
    'netmask': "",
    'gateway': "",
    'channel_name': "",
    'output_if': "",
    'search_type': 'next'
}

while 1:
    # Encode the URL.
    params_url    = urllib.urlencode(params)

    # GET request
    conn.request("GET", '/' + file + '?' + params_url)
    response = conn.getresponse()

    # Display the response.
    print 'RESPONSE:', response

    data = response.read()
    print 'LENGTH  :', len(data)
    print 'DATA      :',
    print '-----'
    print data
    print
```

```
# From the data part of the response (JSON format string)
# get Python dictionary data.
json_data1 = json.loads(data)
json_data2 = json.loads(data)
json_data3 = json.loads(data)
json_data4 = json.loads(data)
json_data5 = json.loads(data)

# Exit if no target IP exists as JSON key.
if json_data1.has_key("target")==False:
    break
# Exit if no netmask exists as JSON key.
if json_data2.has_key("netmask")==False:
    break
# Exit if no gateway exists as JSON key.
if json_data3.has_key("gateway")==False:
    break
# Exit if no channel_name exists as JSON key.
if json_data4.has_key("channel_name")==False:
    break
# Exit if no output_if exists as JSON key.
if json_data5.has_key("output_if")==False:
    break
# Get target IP.
IP_address = json_data1['target']
# Get netmask.
netmask = json_data2['netmask']
# Get gateway.
gateway = json_data3['gateway']
# Get channel_name.
channel_name = json_data4['channel_name']
# Get output_if.
output_if = json_data5['output_if']

# Update target IP to the retrieved one, and continue.
params['IP_address'] = IP_address
# Update netmask to the retrieved one, and continue.
params['netmask'] = netmask

# Update gateway to the retrieved one, and continue.
params['gateway'] = gateway
# Update channel_name to the retrieved one, and continue.
params['channel_name'] = channel_name
```

(continued)

```
# Update output_if to the retrieved one, and continue.  
params['output_if'] = output_if
```

(6) Getting scenario information (specified scenario)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# Specify "exact" for "search_type".
params = {
    'command': 'show scenario',
    'scenario_name': '/port1/North',
    'search_type': 'exact'
}

# Encode the URL.
params_url = urllib.urlencode(params)

# GET request
response = urllib2.urlopen(url+'?' +params_url)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '-----'
print data
print
```

(7) Getting scenario information (all scenarios)

<1> A sample program to get all scenarios while keeping the connection.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL IP address and file name of WebAPI
ip    = '192.168.1.1'
file  = 'shapermng/json'

# Open connection. HTTP
conn = httplib.HTTPConnection(ip)
# Open connection. HTTPS
#conn = httplib.HTTPSConnection(ip)

# Define parameters.
# To display all scenarios, specify an empty string for the first scenario.
# Specify "next" for "search_type".
params = {
    'command': 'show scenario',
    'scenario_name' : "",
    'search_type' : 'next'
}

while 1:
    # Encode URL.
    params_url    = urllib.urlencode(params)

    # GET request.
    conn.request("GET", '/' + file + '?' + params_url)
    response = conn.getresponse()

    # Display the response.
    print 'RESPONSE:', response

    data = response.read()
    print 'LENGTH  :', len(data)
    print 'DATA      : '
    print '-----'
    print data
    print

    # From the data part of the response (JSON format string)
    # get Python dictionary data.
    json_data = json.loads(data)

    # Exit if no scenario name exists as JSON key.
    if json_data.has_key("scenario_name")==False:
        break

    # Get a scenario name.
    scenario_name = json_data['scenario_name']

    # Update the scenario name to the retrieved one, and continue.
    params['scenario_name'] = scenario_name

# Close connection.
conn.close()
```

<2> A sample program open and close the connection for each scenario.

When the following sample program is used, resource of session may be run out at this device and urlopen may fail depending on the performance of the terminal. If urlopen fails, use the sample program which gets all scenarios while keeping the connection of <1>.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# To display all scenarios, specify an empty string for the first scenario.
# Specify "next" for "search_type".
params = {
    'command': 'show scenario',
    'scenario_name': '',
    'search_type': 'next'
}

while 1:
    # Encode the URL.
    params_url = urllib.urlencode(params)

    # GET request
    response = urllib2.urlopen(url+'?' +params_url)

    # Display the response.
    print 'RESPONSE:', response
    print 'URL      :', response.geturl()

    data = response.read()
    print 'LENGTH :', len(data)
    print 'DATA   :'
    print '-----'
    print data
    print

    # From the data part of the response (JSON format string)
    # get Python dictionary data.
    json_data = json.loads(data)

    # Exit if no scenario name exists as JSON key.
    if json_data.has_key("scenario_name")==False:
        break

    # Get a scenario name.
    scenario_name = json_data['scenario_name']

    # Update the scenario name to the retrieved one, and continue.
    params['scenario_name'] = scenario_name
```

(8) Getting filter information (specified filter)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# Specify "exact" for "search_type".
params = {
    'command': 'show filter',
    'scenario_name' : '/port1/North',
    'filter_name' : 'filter1',
    'search_type' : 'exact'
}

# Encode the URL.
params_url = urllib.urlencode(params)

# GET request
response = urllib2.urlopen(url+'?' +params_url)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '-----'
print data
print
```

(9) Getting filter information (all filters)

<1> A sample program keeping the connection to get all filters.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL IP address and file name of WebAPI
ip    = '192.168.1.1'
file  = 'shapermng/json'

# Open connection. HTTP
conn = httplib.HTTPConnection(ip)
# Open connection. HTTPS
#conn = httplib.HTTPSConnection(ip)

# Define parameters.
# To display all filters,
# specify an empty string for the first scenario name and filter name
# Specify "next" for "search_type".
params = {
    'command': 'show filter',
    'scenario_name': '',
    'filter_name': '',
    'search_type': 'next'
}

while 1:
    # Encode URL.
    params_url    = urllib.urlencode(params)

    # GET request.
    conn.request("GET", 'http://'+ip+'/' +file+'?' +params_url)
    response = conn.getresponse()

    # Display the response.
    print 'RESPONSE:', response

    data = response.read()
    print 'LENGTH  :', len(data)
    print 'DATA      : '
    print '-----'
    print data
    print

    # From the data part of the response (JSON format string)
    # get Python dictionary data.
    json_data = json.loads(data)

    # Exit if no scenario name exists as JSON key.
    if json_data.has_key("scenario_name")==False:
        break

    # Exit if no filter name exists as JSON key.
    if json_data.has_key("filter_name")==False:
        break
```


(Continued)

```
# Get a scenario name.
scenario_name = json_data['scenario_name']

# Get a filter name.
filter_name = json_data['filter_name']

# Update the scenario name and filter name to the retrieved one,
# and continue.
params['scenario_name'] = scenario_name
params['filter_name'] = filter_name

# Close connection.
conn.close()
```

<2> A sample program open and close the connection for each filter.

When the following sample program is used, resource of session may be run out at this device and urlopen may fail depending on the performance of the terminal. If urlopen fails, use the sample program which gets all scenarios while keeping the connection of <1>.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# To display all filters,
# specify an empty string for the first scenario name and filter name.
# Specify "next" for "search_type".
params = {
    'command': 'show filter',
    'scenario_name' : '',
    'filter_name' : '',
    'search_type' : 'next'
}

while 1:
    # Encode the URL.
    params_url = urllib.urlencode(params)

    # GET request
    response = urllib2.urlopen(url+'?' +params_url)

    # Display the response.
    print 'RESPONSE:', response
    print 'URL      :', response.geturl()

    data = response.read()
    print 'LENGTH  :', len(data)
    print 'DATA    :'
    print '-----'
    print data
    print

    # From the data part of the response (JSON format string)
    # get Python dictionary data.
    json_data = json.loads(data)

    # Exit if no scenario name exists as JSON key.
    if json_data.has_key("scenario_name")==False:
        break

    # Exit if no filter name exists as JSON key.
    if json_data.has_key("filter_name")==False:
        break
```

(Continued)

```
# Get a scenario name.  
scenario_name = json_data['scenario_name']  
  
# Get a filter name.  
filter_name = json_data['filter_name']  
  
# Update the scenario and filter names to the retrieved ones, and continue.  
params['scenario_name'] = scenario_name  
params['filter_name'] = filter_name
```

(10) Getting rule list information (specified rule list entry)

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# Specify "exact" for "search_type".
params = {
    'command': 'show rulelist',
    'list_name': 'v4servers',
    'rules': '192.168.10.1-192.168.10.1',
    'search_type': 'exact'
}

# Encode the URL.
params_url = urllib.urlencode(params)

# GET request
response = urllib2.urlopen(url+'?'+params_url)

# Display the response.
print 'RESPONSE:', response
print 'URL      :', response.geturl()

data = response.read()
print 'LENGTH  :', len(data)
print 'DATA    :'
print '-----'
print data
print
```

(11) Get rule list information (all rule lists)

<1> A sample program to get all rule lists while keeping the connection.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL IP address and file name of WebAPI
ip    = '192.168.1.1'
file  = 'shapermng/json'

# Open connection. HTTP
conn = httplib.HTTPConnection(ip)
# Open connection. HTTPS
#conn = httplib.HTTPSConnection(ip)

# Define parameters.
# To display all rule lists,
# specify an empty string for the first rule list name and rule list entry.
# Specify "next" for "search_type".
params = {
    'command': 'show rulelist',
    'list_name': '',
    'rules': '',
    'search_type': 'next'
}

while 1:
    # Encode URL.
    params_url    = urllib.urlencode(params)

    # GET request.
    conn.request("GET", 'http://'+ip+'/' +file+'?' +params_url)
    response = conn.getresponse()

    # Display the response.
    print 'RESPONSE:', response

    data = response.read()
    print 'LENGTH  :', len(data)
    print 'DATA      : '
    print '-----'
    print data
    print

    # From the data part of the response (JSON format string)
    # get Python dictionary data.
    json_data = json.loads(data)

    # Exit if no rule list name exists as JSON key.
    if json_data.has_key("list_name")==False:
        break

    # Exit if no rule list entry exists as JSON key.
    if json_data.has_key("rules")==False:
        break
```

```
# Get a rule list name.
scenario_name = json_data['list_name']

# Get a rule list entry.
rules = json_data['rules']

# Update the rule list name and rule list entry to the retrieved one,
# and continue.
# "none" indicates that no more rule list entry in this rule list.
# Specify an empty string to get the next rule list.
params['list_name'] = list_name
if rules == 'none':
    params['rules'] = ""
else:
    params['rules'] = rules

# Close connection.
conn.close()
```

<2> A sample program open and close the connection for each rule list.

When the following sample program is used, resource of session may be run out at this device and urlopen may fail depending on the performance of the terminal. If urlopen fails, use the sample program which gets all scenarios while keeping the connection of <1>.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json

# Define URL URL of WebAPI. HTTP
url = 'http://192.168.1.1/shapermng/json'
# Define URL URL of WebAPI. HTTPS
#url = 'https://192.168.1.1/shapermng/json'

# Define parameters.
# To display all rule lists,
# specify an empty string for the first rule list name and rule list entry.
# Specify "next" for "search_type".
params = {
    'command': 'show rulelist',
    'list_name': '',
    'rules': '',
    'search_type': 'next'
}

while 1:
    # Encode the URL.
    params_url = urllib.urlencode(params)

    # GET request
    response = urllib2.urlopen(url+'?' + params_url)

    # Display the response.
    print 'RESPONSE:', response
    print 'URL      :', response.geturl()

    data = response.read()
    print 'LENGTH  :', len(data)
    print 'DATA    :'
    print '-----'
    print data
    print

    # From the data part of the response (JSON format string)
    # get Python dictionary data.
    json_data = json.loads(data)

    # Exit if no rule list name exists as JSON key.
    if json_data.has_key("list_name")==False:
        break

    # Exit if no rule list entry exists as JSON key.
    if json_data.has_key("rules")==False:
        break
```

(Continued)

```
# Get a rule list name.
list_name = json_data['list_name']

# Get a rule list entry.
rules = json_data['rules']

# Update the rule list name and rule list entry to the retrieved ones,
# and continue.
# "none" indicates that no more rule list entry in this rule list.
# Specify an empty string to get the next rule list.
params['list_name'] = list_name
if rules == 'none':
    params['rules'] = ""
else:
    params['rules'] = rules
```


Appendix H Details of OpenFlow Message Supported for CLI Command

Detailed OpenFlow for this equipment is indicated.

For OpenFlow, provides the JSON data for the data section when the message type is OFPT_EXPERIMENTER (4) and for the data section when the message type is OFPT_MULTIPART_REQUEST (18) and the multipart type is the packet of OFPMP_EXPERIMENTER (0xffff).

In addition, when the message type is OFPT_FLOW_MOD (14), the filter command can be set by using the match field, instruction field, and action field.

All the keys and values are specified with the character strings. If it is not required to specify the keys that can be omitted, no description is required. If a key is incorrectly spelled, this key and value are ignored. An error occurs if a key that must be specified is incorrectly spelled. However, please note that no error occurs even if a key that can be omitted or an undefined key is incorrectly spelled.

For details of the specified value, see “PureFlow WSX Unified Network Controller NF7600 Series Command Reference”.

Next, indicates the JSON data for the data section when the message type is OFPT_EXPERIMENTER (4) and for the data section when the message type is OFPT_MULTIPART_REQUEST (18) and the multipart type is the packet of OFPMP_EXPERIMENTER (0xffff).

At this time, set EXPERIMENTER ID to 0x00000091.

Set the value corresponding to the CLI command type in the exp_type field.

Corresponding CLI command	exp_type
add scenario	1
update scenario	2
delete scenario	3
show scenario	4
show scenario counter	5
add apl-accel	6
update apl-accel	7
delete apl-accel	8
add filter	9
delete filter	10
show filter	11
add rulelist group	12
delete rulelist group	13
add rulelist entry	14
delete rulelist entry	15
show rulelist	16

Corresponding CLI command	exp_type
add channel	17
delete channel	18
show channel	19
set ip channel	20
unset ip channel	21
show ip channel	22
add route	23
delete route	24
show route target	25
set wan-accel bypass status	26
set wan-accel bypass recoverytime	27
switch wan-accel bypass force	28

Next, indicates the JSON data corresponding to the data section.

CLI command type	Key	Value	Corresponding CLI command and parameter
Add a scenario (Discard)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"discard"	action discard
	"scenario_id" (Optional)	Scenario ID	[scenario <scenario_id>]
Add a scenario (Aggregate)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"aggregate"	action aggregate
	"cos" (Optional)	Cos value	[cos <user_priority>]
	"inner-cos" (Optional)	Inner-Cos value	[inner-cos <user_priority>]
	"dscp" (Optional)	dscp	[dscp <dscp>]
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
"scenario_id" (Optional)	Scenario ID	[scenario <scenario_id>]	

CLI command type	Key	Value	Corresponding CLI command and parameter
Add a scenario (Individual)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"individual"	action individual
	"cos" (Optional)	Cos value	[cos <user_priority>]
	"inner-cos" (Optional)	Inner-Cos value	[inner-cos <user_priority>]
	"dscp" (Optional)	dscp	[dscp <dscp>]
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
	"scenario_id" (Optional)	Scenario ID	[scenario <scenario_id>]
	"maxqnum" (Optional)	Maximum number of individual queues	[maxquenum <quenum>]
	"quedivision" (Optional)	Individual que division target	[quedivision <field>]
	"failaction" (Optional)	Operation if the number of individual queues is exceeded	[failaction <discard forwardbesteffort forwardattribute>]
	"fail_min_bw" (Optional)	Minimum bandwidth if the number of individual queues is exceeded	[fail_min_bw <min_bandwidth>]
	"fail_peak_bw" (Optional)	Peak bandwidth if the number of individual queues is exceeded	[fail_peak_bw <peak_bandwidth>]
"fail_class" (Optional)	Class if the number of individual queues is exceeded	[fail_class <class>]	
Add a scenario (Wan-accel)	"command" (Required)	"add scenario"	add scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"wan-accel"	action wan-accel
	"peer" (Required)	IP address	peer <IP_address>
	"second_peer" (Optional)	IP address	[second-peer <IP_address >]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"dport" (Optional)	Destination port number	[dport <port>]
	"vid" (Optional)	VLAN ID	[vid <VID>]
	"inner-vid" (Optional)	Inner-VLAN ID	[inner-vid <VID>]
	"cos" (Optional)	Cos value	[cos <user_priority>]
	"inner-cos" (Optional)	Inner-Cos value	[inner-cos <user_priority>]
	"dscp" (Optional)	dscp	[dscp <dscp>]
	"compression" (Optional)	Compression "enable": Enables compression. "disable": Disables compression When omitted, "disable" is applied.	[compression {enable disable}]
	"tcp_mem" (Optional)	TCP buffer size	[tcp-mem {auto <size>}]
	"cc_mode" (Optional)	Congestion control mode "normal": Sets the congestion control mode to normal mode. "semi-fast": Sets the congestion control mode to high speed mode. "fast": Sets the congestion control mode to high speed mode. When omitted, "normal" is applied.	[cc-mode {normal semi-fast fast}]
	"bypass_thresh" (Optional)	RTT	[bypass-thresh <rtt>]
	"bypass_keepalive" (Optional)	Auto bypass keepalive "enable": Enables keepalive. "disable": Disables keepalive. When omitted, "disable" is applied.	[bypass-keepalive {enable disable}]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"fec" (Optional)	FEC "enable": Enables the FEC function. "disable": Disables the FEC function. When omitted, "disable" is applied.	[fec {enable disable}]
	"block_size" (Optional)	FEC block size	[block-size <size>]
	"data_block_size" (Optional)	FEC data block size	[data-block-size <size>]
	"fec_session" (Optional)	FEC session count	[fec-session <session>]
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
Update a scenario (Aggregate)	"command" (Required)	"update scenario"	update scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"aggregate"	action aggregate
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
Update a scenario (Individual)	"command" (Required)	"update scenario"	update scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"individual"	action individual
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
	"maxqnum" (Optional)	Maximum number of individual queues	[maxqnum <qnum>]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"quedivision" (Optional)	Individual queue division target	[quedivision <field>]
	"failaction" (Optional)	Operation if the number of individual queues is exceeded	[failaction <discard forwardbesteffort forwardattribute>]
	"fail_min_bw" (Optional)	Minimum bandwidth if the number of individual queues is exceeded	[fail_min_bw <min_bandwidth>]
	"fail_peak_bw" (Optional)	Peak bandwidth if the number of individual queues is exceeded	[fail_peak_bw <peak_bandwidth>]
	"fail_class" (Optional)	Class if the number of individual queues is exceeded	[fail_class <class>]
Update a scenario (Wan-accel)	"command" (Required)	"update scenario"	update scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"action" (Required)	"wan-accel"	action wan-accel
	"compression" (Optional)	Compression "enable": Enables compression. "disable": Disables compression When omitted, "disable" is applied.	[compression {enable disable}]
	"tcp_mem" (Optional)	TCP buffer size	[tcp-mem {auto <size>}]
	"cc_mode" (Optional)	Congestion control mode "normal": Sets the congestion control mode to normal mode. "semi-fast": Sets the congestion control mode to high speed mode. "fast": Sets the congestion control mode to high speed mode. When omitted, "normal" is applied.	[cc-mode {normal semi-fast fast}]
	"bypass_thresh" (Optional)	RTT	[bypass-thresh <rtt>]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"bypass_keepalive" (Optional)	Auto bypass keepalive "enable": Enables keepalive. "disable": Disables keepalive. When omitted, "disable" is applied.	[bypass-keepalive {enable disable}]
	"fec" (Optional)	FEC "enable": Enables the FEC function. "disable": Disables the FEC function. When omitted, "disable" is applied.	[fec {enable disable}]
	"block_size" (Optional)	FEC block size	[block-size <size>]
	"data_block_size" (Optional)	FEC data block size	[data-block-size <size>]
	"fec_session" (Optional)	FEC session count	[fec-session <session>]
	"min_bandwidth" (Optional)	Minimum bandwidth	[min_bw <min_bandwidth>]
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
Delete a scenario (all specified)	"command" (Required)	"delete scenario"	delete scenario
	"scenario_name" (Required)	"all"	all
Delete a scenario (scenario specified)	"command" (Required)	"delete scenario"	delete scenario
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"recursive" (Optional)	"recursive"	[recursive]
Get scenario information	"command" (Required)	"show scenario"	show scenario
	"scenario_name" (Required)	Scenario name	name <scenario_name>

CLI command type	Key	Value	Corresponding CLI command and parameter
	"search_type" (Optional)	Acquisition method "exact": Acquires the information on the specified scenario. "next": Acquires the information on the scenario next to the specified scenario. When omitted or the value is incorrectly spelled, "exact" is applied.	None
Get scenario counter information	"command" (Required)	"show scenario counter"	show scenario counter
	"scenario_name" (Required)	Scenario name	name <scenario_name>
	"search_type" (Optional)	Acquisition method "exact": Acquires the information on the specified scenario. "next": Acquires the information on the scenario next to the specified scenario. When omitted or the value is incorrectly spelled, "exact" is applied.	None
	"default_queue" (Optional)	"default_queue"	[default_queue]

About scenario information acquisition

To acquire the scenario information, there is the "search_type" parameter to specify the acquisition method. Specify "exact" or "next" in "search_type" as the value.

"exact": Acquires the information on the scenario specified in "scenario_name".

"next": Acquires the information on the scenario next to the scenario specified in "scenario_name".

Acquire the scenarios in the order of the scenario tree, the same as the "show scenario" CLI command.

When "search_type" is omitted, "exact" is applied.

To acquire the specified scenario information, specify the scenario name and apply "exact" to acquire the scenario.

To acquire all the scenario information, the same as "show scenario all" in the CLI command, use "next" and acquire the information according to the following procedure.

Specify a null character in "scenario_name" to acquire the first scenario information.

```
"scenario_name" : "" (null character)
```

```
"search_type" : "next"
```

The information on scenario "/port1" at the head of the scenario tree can be acquired.

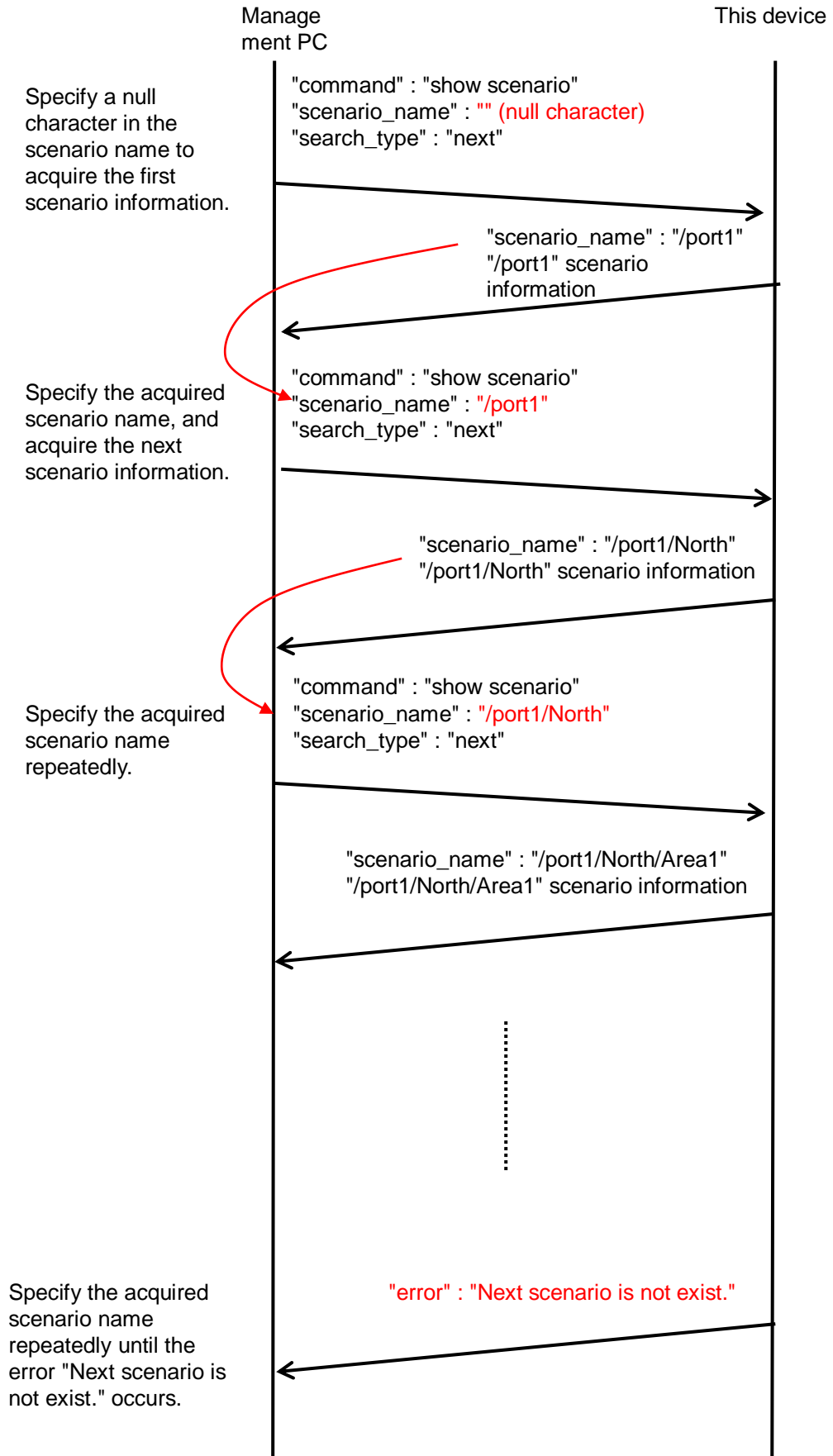
Next, specify the acquired scenario name in "scenario_name".

```
"scenario_name" : "/port1"
```

```
"search_type" : "next"
```

The information on the scenario next to scenario "/port1" in the scenario tree can be acquired.

Specify the acquired scenario name and acquire the information by "next" repeatedly. When specifying the bottom of the scenario tree and acquiring the information by "next", the error "Next scenario is not exist." occurs.



CLI command type	Key	Value	Corresponding CLI command and parameter
Add a filter (Bridge-ctrl)	"command" (Required)	"add filter"	add filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"type" (Required)	"bridge-ctrl"	bridge-ctrl
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
Add a filter (Ethernet)	"command" (Required)	"add filter"	add filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"type" (Required)	"ethernet"	ethernet
	"vid" (Optional)	VLAN ID	[vid {<VID> none}]
	"cos" (Optional)	CoS	[cos <user_priority>]
	"inner-vid" (Optional)	Inner-VLAN ID	[inner-vid {<VID> none}]
	"inner-cos" (Optional)	Inner-CoS	[inner-cos <user_priority>]
	"ethertype" (Optional)	Ethernet Type/Length	[ethertype <type>]
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
Add a filter (IPv4)	"command" (Required)	"add filter"	add filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"type" (Required)	"ipv4"	ipv4
	"vid" (Optional)	VLAN ID	[vid {<VID> none}]
	"cos" (Optional)	CoS	[cos <user_priority>]
	"inner-vid" (Optional)	Inner-VLAN ID	[inner-vid {<VID> none}]
	"inner-cos" (Optional)	Inner-CoS	[inner-cos <user_priority>]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"sip" or "sip list" (Optional)	Source IPv4 address or Rule list name If "sip" and "sip list" are used at the same time, "sip list" is prioritized.	[sip [list] {<src_IP_address> <list_name>}]
	"dip" or "dip list" (Optional)	Destination IPv4 address or Rule list name If "dip" and "dip list" are used at the same time, "dip list" is prioritized.	[dip [list] {<dst_IP_address> <list_name>}]
	"tos" (Optional)	ToS	[tos <type_of_service>]
	"proto" (Optional)	Protocol number	[proto <protocol>]
	"sport" or "sport list" (Optional)	Source port number or Rule list name If "sport" and "sport list" are used at the same time, "sport list" is prioritized.	[sport [list] {<sport> <list_name>}]
	"dport" or "dport list" (Optional)	Destination port number or Rule list name If "dport" and "dportlist" are used at the same time, "dportlist" is prioritized.	[dport [list] {<dport> <list_name>}]
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
Add a filter (IPv6)	"command" (Required)	"add filter"	add filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"type" (Required)	"ipv6"	ipv6
	"vid" (Optional)	VLAN ID	[vid {<VID> none}]
	"cos" (Optional)	CoS	[cos <user_priority>]
	"inner-vid" (Optional)	Inner-VLAN ID	[inner-vid {<VID> none}]
	"inner-cos" (Optional)	Inner-CoS	[inner-cos <user_priority>]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"sip" or "sip list" (Optional)	Source IPv6 address or Rule list name If "sip" and "sip list" are used at the same time, "sip list" is prioritized.	[sip [list] {<src_IP_address> <list_name>}]
	"dip" or "dip list" (Optional)	Destination IPv6 address or Rule list name If "dip" and "dip list" are used at the same time, "dip list" is prioritized.	[dip [list] {<dst_IP_address> <list_name>}]
	"tos" (Optional)	ToS	[tos <type_of_service>]
	"proto" (Optional)	Protocol number	[proto <protocol>]
	"sport" or "sport list" (Optional)	Source port number or Rule list name If "sport" and "sport list" are used at the same time, "sport list" is prioritized.	[sport [list] {<sport> <list_name>}]
	"dport" or "dport list" (Optional)	Destination port number or Rule list name If "dport" and "dport list" are used at the same time, "dport list" is prioritized.	[dport [list] {<dport> <list_name>}]
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
Delete a filter (all specified)	"command" (Required)	"delete filter"	delete filter
	"scenario_name" (Required)	"all"	all
Delete a filter (scenario specified)	"command" (Required)	"delete filter"	delete filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
Delete a filter (filter specified)	"command" (Required)	"delete filter"	delete filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <scenario_name>
Delete filter information	"command" (Required)	"show filter"	show filter
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>

CLI command type	Key	Value	Corresponding CLI command and parameter
	"filter_name" (Required)	Filter name	filter <scenario_name>
	"search_type" (Optional)	Acquisition method "exact": Acquires the information on the specified filter. "next": Acquires the information on the filter next to the specified filter. When omitted or the value is incorrectly spelled, "exact" is applied.	None

About filter information acquisition

To acquire the filter information, there is the "search_type" parameter to specify the acquisition method. Specify "exact" or "next" in "search_type" as the value.

"exact": Acquires the information on the filter specified in "scenario_name" and "filter_name".

"next": Acquires the information on the filter next to the filter specified in "scenario_name" and "filter_name". Acquire the scenarios in the alphabetical order of filter names, the same as the "show filter" CLI command. When the bottom filter of the scenario is specified, acquire the information on the filter at the head of the next scenario.

To acquire the specified filter information, specify the scenario name and filter name, and apply "exact" to acquire the scenario.

To acquire all the filter information on all the scenarios, the same as "show filter all" in the CLI command, use "next". Acquire the filter information by "next" according to the same procedure as scenario acquisition.

CLI command type	Key	Value	Corresponding CLI command and parameter
Add application acceleration	"command" (Required)	"add apl-accel"	add apl-accel
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"protocol " (Required)	Protocol name	protocol smb
	"tcp_port" (Optional)	TCP port number	[tcp <port>]
	"smb-session" (Optional)	Session count	[smb-session <session>]
	"read-attr" (Optional)	Substitute response for the SMB2 QUERY_INFO command in the reading operation "enable": Enables the substitute response for the SMB2 QUERY_INFO command. "disable": Disables the substitute response for the SMB2 QUERY_INFO command. When omitted, "enable" is applied.	[read-attr {enable disable}]
	"read-operation" (Optional)	Substitute response for the SMB2 READ command in the reading operation "enable": Enables the substitute response for the SMB2 READ command. "disable": Disables the substitute response for the SMB2 READ command. When omitted, "enable" is applied.	[read-operation {enable disable}]
"read-cache-size" (Optional)	Cache size of the substitute response in the reading operation	[read-cache-size <size>]	

CLI command type	Key	Value	Corresponding CLI command and parameter
	"write-attr" (Optional)	Substitute response for the SMB2 QUERY_INFO command in the writing operation "enable": Enables the substitute response for the SMB2 QUERY_INFO command. "disable": Disables the attribution substitute response for the SMB2 QUERY_INFO command. When omitted, "enable" is applied.	[write-attr {enable disable}]
	"write-attr-1st" (Optional)	Substitute response for the SMB2 SET_INFO command before the writing operation "enable": Enables the substitute response for the SMB2 SET_INFO command. "disable": Disables the substitute response for the SMB2 SET_INFO command. When omitted, "disable" is applied.	[write-attr-1st {enable disable}]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"write-attr-2nd" (Optional)	Substitute response for the SMB2 SET_INFO command after the writing operation "enable": Enables the substitute response for the SMB2 SET_INFO command. "disable": Disables the substitute response for the SMB2 SET_INFO command. When omitted, "disable" is applied.	[write-attr-2nd {enable disable}]
	"write-operation" (Optional)	Substitute response for the SMB2 WRITE command in the writing operation "enable": Enables the substitute response for the SMB2 WRITE command. "disable": Disables the substitute response for the SMB2 WRITE command. When omitted, "enable" is applied.	[write-operation {enable disable}]
Update application acceleration	"command" (Required)	"update apl-accel"	update apl-accel
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"protocol " (Required)	Protocol name	protocol smb
	"tcp_port" (Optional)	TCP port number	[tcp <port>]
	"smb-session" (Optional)	Session count	[smb-session <session>]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"read-attr" (Optional)	Substitute response for the SMB2 QUERY_INFO command in the reading operation "enable": Enables the substitute response for the SMB2 QUERY_INFO command. "disable": Disables the substitute response for the SMB2 QUERY_INFO command. When omitted, "enable" is applied.	[read-attr {enable disable}]
	"read-operation" (Optional)	Substitute response for the SMB2 READ command in the reading operation "enable": Enables the substitute response for the SMB2 READ command. "disable": Disables the substitute response for the SMB2 READ command. When omitted, "enable" is applied.	[read-operation {enable disable}]
	"read-cache-size" (Optional)	Cache size of the substitute response in the reading operation	[read-cache-size <size>]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"write-attr" (Optional)	Substitute response for the SMB2 QUERY_INFO command in the writing operation "enable": Enables the substitute response for the SMB2 QUERY_INFO command. "disable": Disables the attribution substitute response for the SMB2 QUERY_INFO command. When omitted, "enable" is applied.	[write-attr {enable disable}]
	"write-attr-1st" (Optional)	Substitute response for the SMB2 SET_INFO command before the writing operation "enable": Enables the substitute response for the SMB2 SET_INFO command. "disable": Disables the substitute response for the SMB2 SET_INFO command. When omitted, "disable" is applied.	[write-attr-1st {enable disable}]

CLI command type	Key	Value	Corresponding CLI command and parameter
	"write-attr-2nd" (Optional)	Substitute response for the SMB2 SET_INFO command after the writing operation "enable": Enables the substitute response for the SMB2 SET_INFO command. "disable": Disables the substitute response for the SMB2 SET_INFO command. When omitted, "disable" is applied.	[write-attr-2nd {enable disable}]
	"write-operation" (Optional)	Substitute response for the SMB2 WRITE command in the writing operation "enable": Enables the substitute response for the SMB2 WRITE command. "disable": Disables the substitute response for the SMB2 WRITE command. When omitted, "enable" is applied.	[write-operation {enable disable}]
Delete application acceleration	"command" (Required)	"delete apl-accel"	delete apl-accel
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"protocol " (Required)	Protocol name	protocol smb

CLI command type	Key	Value	Corresponding CLI command and parameter
Add a rule list group	"command" (Required)	"add rulelist group"	add rulelist group
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	Rule list type	{ipv4 ipv6 l4port}
Delete a rule list group (all specified)	"command" (Required)	"delete rulelist group"	delete rulelist group
	"list_name" (Required)	"all"	all
Delete a rule list group (group specified)	"command" (Required)	"delete rulelist group"	delete rulelist group
	"list_name" (Required)	Rule list name	<list_name>
Add a rule list entry (IPv4)	"command" (Required)	"add rulelist entry"	add rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv4"	ipv4
	"IP_address" (Required)	IPv4 address	<IP_address>
Add a rule list entry (IPv6)	"command" (Required)	"add rulelist entry"	add rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv6"	ipv6
	"IP_address" (Required)	IPv6 address	<IP_address>
Add a rule list entry (L4Port)	"command" (Required)	"add rulelist entry"	add rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"l4port"	l4port
	"port" (Required)	L4 port number	<port>
Delete a rule list entry (all specified)	"command" (Required)	"delete rulelist entry"	delete rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"all"	all

CLI command type	Key	Value	Corresponding CLI command and parameter
Delete a rule list entry (IPv4)	"command" (Required)	"delete rulelist entry"	delete rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv4"	ipv4
	"IP_address" (Required)	IPv4 address	<IP_address>
Delete a rule list entry (IPv6)	"command" (Required)	"delete rulelist entry"	delete rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv6"	ipv6
	"IP_address" (Required)	IPv6 address	<IP_address>
Delete a rule list entry (IPv6)	"command" (Required)	"delete rulelist entry"	delete rulelist entry
	"list_name" (Required)	Rule list name	<list_name>
	"type" (Required)	"ipv6"	ipv6
	"port" (Required)	L4 port number	<port>
Get rule list information	"command" (Required)	"show rulelist"	show rulelist
	"list_name" (Required)	Rule list name	[<list_name>]
	"rules" (Required)	Adding a rule list entry (Continued)	None
	"search_type" (Optional)	Acquisition method "exact": Acquires the specified rule list entry. "next": Acquires the rule list entry next to the specified rule list entry. When omitted or the value is incorrectly spelled, "exact" is applied.	None

About rule list information acquisition

To acquire the rule list information, there is the "rules" parameter that does not exist in the "show rulelist" CLI command.

Specify the rule list entry (IP address or L4 port number) in "rules" as the value. Even if the rule list is a single value, use a hyphen to specify the range every time.

IPv4 address 192.168.1.1-192.168.1.1

IPv6 address FE80::0001-FE80::0001

L4 port number 1000-1000

In the rule list without setting the rule list entry, "none" is acquired.

Specify "exact" or "next" in "search_type" as the value to specify the acquisition method.

"exact" Acquires the rule list entry specified in "list_name" and "rules".

"next": Acquires the rule list entry next to the rule list entry specified in "list_name" and "rules". Acquire the rule list entry in the same order as the "show rulelist" CLI command. When the rule list entry at the bottom of the rule list is specified, acquire the rule list entry at the head of the next rule list.

To acquire the specified rule list entry, specify the rule list name and rule list entry, and use "exact" to acquire the entry.

To acquire all the rule list entries of all the rule lists, the same as "show rulelist all" in the CLI command, use "next". Acquire the rule list entries by "next" according to the same procedure as scenario acquisition.

CLI command type	Key	Value	Corresponding CLI command and parameter
Add a channel (Normal channel)	"command" (Required)	"add channel"	add channel
	"channel_name" (Required)	Channel name	<channel_name>
	"lan" (Required)	Port number on the LAN side or Port group	lan {<slot/port> <group_name>}
	"wan" (Required)	Port number on the WAN side or Port group	wan {<slot/port> <group_name>}
	"channel_type" (Required)	Channel type "normal": Adds the normal channel. "default": Adds the default channel.	None
	"vid" (Required)	VLAN ID	vid {<VID> none}
	"inner_vid" (Optional)	Inner-VLAN ID	[inner-vid {<VID> none}]
	"tpid" (Optional)	tpid	[tpid <tpid>]
	"inner_tpid" (Optional)	inner-tpid	[inner-tpid <tpid>]
	"mtu" (Optional)	mtu	[mtu <mtu>]
Add a channel (Default channel)	"command" (Required)	"add channel"	add channel
	"channel_name" (Required)	Channel name	<channel_name>
	"lan" (Required)	Port number on the LAN side or Port group	lan {<slot/port> <group_name>}
	"wan" (Required)	Port number on the WAN side or Port group	wan {<slot/port> <group_name>}
	"channel_type" (Required)	Channel type "normal": Adds the normal channel. "default": Adds the default channel.	None

CLI command type	Key	Value	Corresponding CLI command and parameter
Delete a channel (All specified)	"command" (Required)	"delete channel"	delete channel
	"channel_name" (Required)	"all"	all
Delete a channel (Channel name specified)	"command" (Required)	"delete channel"	delete channel
	"channel_name" (Required)	Channel name	<channel_name>
Show channel information	"command" (Required)	"show channel"	show channel
	"channel_name" (Required)	Channel name	name <channel_name>
	"search_type" (Optional)	Acquisition method "exact": Acquires the specified channel. "next": Acquires the channel next to the specified channel. When omitted or the value is incorrectly spelled, "exact" is applied.	None

About channel information acquisition

To acquire the channel information, there is the "search_type" parameter to specify the acquisition method. Specify "exact" or "next" in "search_type" as the value.

"exact": Acquires the information on the channel specified in "channel_name".

"next": Acquires the information on the channel next to the channel specified in "channel_name". Acquire the scenarios in the alphabetical order of channel names, the same as the "show channel" CLI command.

To acquire the specified channel information, specify the channel name and apply "exact".

To acquire all the channel information, the same as "show channel all" in the CLI command, use "next". Acquire the channel information by "next" according to the same procedure as scenario acquisition.

CLI command type	Key	Value	Corresponding CLI command and parameter
Set interface	"command" (Required)	"set ip channel"	set ip channel
	"channel_name" (Required)	Channel name	<channel_name>
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"netmask" (Required)	IPv4 net mask or IPv6 prefix	netmask <netmask>
Release interface (All specified)	"command" (Required)	"unset ip channel"	unset ip channel
	"channel_name" (Required)	"all"	all
release interface (Channel name specified)	"command" (Required)	"unset ip channel"	unset ip channel
	"channel_name" (Required)	Channel name	<channel_name>
	"type" (Optional)	Setting release target "ipv4": Releases the IPv4 channel interface settings. "ipv6": Releases the IPv6 channel interface settings. When omitted, the settings of the channel interfaces of both IPv4 and IPv6 are released.	{ipv4 ipv6}
Show interface information	"command" (Required)	"show ip channel"	show ip channel
	"channel_name" (Required)	Channel name	name <channel_name>
	"search_type" (Optional)	Acquisition method "exact": Acquires the specified channel. "next": Acquires the channel next to the specified channel. When omitted, "exact" is applied.	None

About interface information acquisition

To acquire the interface information, there is the "search_type" parameter to specify the acquisition method. Specify "exact" or "next" in "search_type" as the value.

"exact": Acquires the information on the interface specified in "channel_name".

"next": Acquires the information on the interface next to the interface specified in "channel_name". Acquire the scenarios in the alphabetical order of channel names, the same as the "show ip interface" CLI command.

To acquire the specified interface information, specify the interface name and apply "exact".

To acquire all the interface information, the same as "show ip interface all" in the CLI command, use "next". Acquire the interface information by "next" according to the same procedure as scenario acquisition.

CLI command type	Key	Value	Corresponding CLI command and parameter
Add a static path (Destination specified)	"command" (Required)	"add route"	add route
	"route_type" (Required)	target	target
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"netmask" (Required)	IPv4 net mask or IPv6 prefix	netmask <netmask>
	"gateway" (Required)	IPv4 address or IPv6 address	gateway <gateway>
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The Transmission Network port is on the LAN side. "wan": The Transmission Network port is on the WAN side.	{lan wan}
Add a static path (Default gateway)	"command" (Required)	"add route"	add route
	"route_type" (Required)	default	default
	"gateway" (Required)	IPv4 address or IPv6 address	gateway <gateway>
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The Transmission Network port is on the LAN side. "wan": The Transmission Network port is on the WAN side.	{lan wan}
Delete a static path (All specified)	"command" (Required)	"delete route"	delete route
	"route_type" (Required)	all	all

CLI command type	Key	Value	Corresponding CLI command and parameter
Delete a static path (Destination specified)	"command" (Required)	"delete route"	delete route
	"route_type" (Required)	target	target
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"netmask" (Required)	IPv4 net mask or IPv6 prefix	netmask <netmask>
	"gateway" (Required)	IPv4 address or IPv6 address	gateway <gateway>
	"channel_name" (Required)	Channel name	channel <channel_name>
Delete a static path (Default gateway)	"command" (Required)	"delete route"	delete route
	"route_type" (Required)	default	default
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The Transmission Network port is on the LAN side. "wan": The Transmission Network port is on the WAN side.	{lan wan}
	"type" (Optional)	Setting release target "ipv4": Deletes the settings of IPv4 static path information. "ipv6": Deletes the settings of IPv6 static path information. When omitted, the settings of static path information of both IPv4 and IPv6 are deleted.	{ipv4 ipv6}

CLI command type	Key	Value	Corresponding CLI command and parameter
Show static path information (Destination specified)	"command" (Required)	"show route target"	show route target
	"IP_address" (Required)	IPv4 address or IPv6 address	<IP_address>
	"netmask" (Required)	IPv4 net mask or IPv6 prefix	netmask <netmask>
	"gateway" (Required)	IPv4 address or IPv6 address	gateway <gateway>
	"channel_name" (Required)	Channel name	channel <channel_name>
	"output_if" (Required)	Transmission Network port "lan": The Transmission Network port is on the LAN side. "wan": The Transmission Network port is on the WAN side.	{lan wan}
	"search_type" (Optional)	Acquisition method "exact": Acquires the specified channel. "next": Acquires the channel next to the specified channel. When omitted or the value is incorrectly spelled, "exact" is applied.	None

About path information acquisition

To acquire the path information, there is the "search_type" parameter to specify the acquisition method. Specify "exact" or "next" in "search_type" as the value.

"exact" Acquires the path information matching all the input parameters.

"next" Acquires the path information next to the path information matching all the input parameters. Acquire the path information in the entry order of channel names, the same as the "show route" CLI command.

To acquire the specified path information, specify all the path information and apply "exact".

To acquire all the path information, the same as "show route all" in the CLI command, use "next". Acquire the path information by "next" according to the same procedure as scenario acquisition.

CLI command type	Key	Value	Corresponding CLI command and parameter
Set traffic acceleration bypass	"command" (Required)	"set wan-accel bypass status"	set wan-accel bypass status
	"status" (Required)	Bypass "enable": Enables bypass. "disable": Disables bypass.	{enable disable}
Set traffic acceleration bypass recovery time	"command" (Required)	"set wan-accel bypass recoverytime"	set wan-accel bypass recoverytime
	"recoverytime" (Required)	Bypass recovery time	<duration>
Set traffic acceleration forced bypass (all specified)	"command" (Required)	"switch wan-accel bypass force"	switch wan-accel bypass force
	"status" (Required)	Forced bypass "enable": Enables forced bypass. "enable": Disables forced bypass.	{enable disable}
	"scenario_name" (Required)	"all"	all
Set traffic acceleration forced bypass (scenario specified)	"command" (Required)	"switch wan-accel bypass force"	switch wan-accel bypass force
	"status" (Required)	Forced bypass "enable": Enables forced bypass. "enable": Disables forced bypass.	{enable disable}
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>

Next, indicates details of Modify Flow Entry (referred to as FlowMod hereafter).

Send the FlowMod (OFPT_FLOW_MOD) message of OpenFlow to register and delete filters. Specify the following in the command field to register or delete filters.

CLI command type	Command field	Relevant CLI command and parameter
Add filter	OFPFC_ADD	add filter
Delete filter	OFPFC_DELETE_STRICT	delete filter

Specify the FlowMod message parameter by the Match field and JSON. Parameters that can be specified in the Match field are shown below: All the Match fields can be omitted. The omitted parameters can be handled as the default value.

CLI command type	Field	Relevant CLI command and parameter	Prior condition
Add filter	OFPXMT_OFB_ETH_TYPE	[ethertype <type>]	None
	OFPXMT_OFB_IP_PROTO	[proto <protocol>]	"0x0800" (ipv4) or "0x86dd" (ipv6) must be specified in ethertype.
	OFPXMT_OFB_IPV4_SRC	[sip <src_IP_address>]	"0x0800" (ipv4) must be specified in ethertype.
	OFPXMT_OFB_IPV4_DST	[dip <dst_IP_address>]	"0x0800" (ipv4) must be specified in ethertype.
	OFPXMT_OFB_IPV6_SRC	[sip <src_IP_address>]	"0x086dd" (ipv6) must be specified in ethertype.
	OFPXMT_OFB_IPV6_DST	[dip <dst_IP_address>]	"0x086dd" (ipv6) must be specified in ethertype.
	OFPXMT_OFB_TCP_SRC	[sport <sport>]	"6" (tcp) must be specified in proto.
	OFPXMT_OFB_TCP_DST	[dport <dport>]	"6" (tcp) must be specified in proto.
	OFPXMT_OFB_UDP_SRC	[sport <sport>]	"17" (udp) must be specified in proto.
OFPXMT_OFB_UDP_DST	[sport <sport>]	"17" (udp) must be specified in proto.	
Delete filter	None	None	None

To delete filters, the above Match fields are ignored. Multiple values cannot be specified in the same CLI command parameters. If a request that does not meet the prior condition was sent, the filter may not be registered correctly.

For details of the specified value, see "PureFlow WSX Unified Network Controller NF7600 Series Command Reference".

Next, the procedure for specifying JSON is described below. Specify OFPIT_APPLY_ACTIONS in the instructions field, specify OFPAT_EXPERIMENTER in the actions field, and specify the JSON-format character strings in the data section of EXPERIMENTER. In addition, specify 0x00000091 in EXPERIMENTER ID.

CLI command type	instructions	actions
Add filter	OFPIT_APPLY_ACTIONS	OFPAT_EXPERIMENTER
Delete filter		

Parameters that can be specified in JSON are shown below: JSON has parameters that must be specified and those that can be omitted. The omitted parameters can be handled as the default value. For details of the JSON format, see Appendix E.

CLI command type	Key	Value	Corresponding CLI command and parameter
Add filter	"type" (Required)	Filter type	{bridge-ctrl ethernet ipv4 ipv6}
	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Required)	Filter name	filter <filter_name>
	"priority" (Optional)	Filter priority	[priority <filter_pri>]
	"vid" (Optional)	VLAN ID	[vid {<VID> none}]
	"inner-vid" (Optional)	INNER VLAN ID	[inner-vid {<VID> none}]
	"sip list" (Optional)	Rule list name	[sip list <list_name>]
	"dip list" (Optional)	Rule list name	[dip list <list_name>]
	"sport list" (Optional)	Rule list name	[sport list <list_name>]
	"dport list" (Optional)	Rule list name	[dport list <list_name>]
Delete filter	"scenario_name" (Required)	Scenario name	scenario <scenario_name>
	"filter_name" (Optional)	Filter name	[filter <filter_name>]

All the keys and values are specified with the character strings. If it is not required to specify the parameters that can be omitted, no description is required. If a key is incorrectly spelled, this parameter is ignored. An error occurs if a parameter that must be specified is incorrectly spelled. However, please note that no error occurs even if a parameter that can be omitted or an undefined parameter is incorrectly spelled.

Appendix I Details of OpenFlow Messages

The responses to OpenFlow messages in this equipment are as follows.

➤ OFPT_HELLO

This message is supported in this equipment.

It exchanges the version of the OpenFlow protocol supported between the OpenFlow controller and this equipment.

Message type

OFPT_HELLO (0)

The bit positions of the version field in the OpenFlow protocol header and the bitmap field of the Hello message are defined as follows.

OpenFlow version	Condition
1.0	0x01
1.1	0x02
1.2	0x03
1.3	0x04
1.4	0x05

This equipment sends Hello messages that support v1.3 only (set 0x04 in the version field, OFPHET_VERSIONBITMAP (0x0001) in the type field of the Hello message, and 0x00000010 in the bitmaps field).

The conditions of error messages are shown below.

Error type	Error code	Support	Condition
OFPET_BAD_REQUEST	OFPBRC_BAD_TYPE	OK	Another Hello message is received again after a Hello message is exchanged with the OpenFlow controller (existing process of OpenvSwitch)
OFPET_HELLO_FAILURE(0)	OFPHFC_INCOMPATIBLE(0)	OK	The OpenFlow controller does not support v1.3
	OFPHFC_EPERM(1)	N/A	Authority error

OFPHFC_INCOMPATIBLE error message

1. In the version field of the OpenFlow protocol header, set the value of the version that did not match. (To allow the OpenFlow controller to identify)
2. In the data section, store a string that indicates the details of the error.

Example:

- When the OpenFlow controller supports v1.0 only
We support version 0x04, you support version 0x01, no common versions.
- When the OpenFlow controller supports v1.4 only
We support version 0x04, you support version 0x05, no common versions.

When a Hello packet with the OpenFlow protocol header only is received from the OpenFlow controller

This equipment regards only the versions set in the version field as supported controllers.

After the error message is sent, disconnect the TCP session with an RST (reset) packet, and connect again.

Number of times of reconnection: Not limited

Retransmission interval: Approx. 10 seconds or less

- OFPT_ECHO_REQUEST
- OFPT_ECHO_REPLY

This message is supported in this equipment.

It sends an Echo request message from this equipment at 5-second intervals after a TCP session with the OpenFlow controller is established.

Message type

OFPT_ECHO_REQUEST (2)

OFPT_ECHO_REPLY (3)

If no Echo response message can be received from the OpenFlow controller (4 times in a row), disconnect the TCP session with an RST (reset) packet, and connect again.

Number of times of reconnection: Not limited

Retransmission interval: Approx. 10 seconds or less

➤ OFPT_EXPERIMENTER

This message is supported in this equipment.

This message is used to respond to setting/display CLI commands.

The Experimenter Multipart message also supports setting/display CLI commands.

Message type

OFPT_EXPERIMENTER (4)

The data structure (struct ofp_experimenter_header) is shown below.

Item	Type	Name	Description
Header	struct ofp_header	header	OpenFlow protocol header
Extended ID	uint32_t	experimenter	IEEE OUI (Anritsu : 0x00000091)
Extension type	uint32_t	exp_type	Arbitrary ID (ignored)
Data section	uint8_t	data[0]	Set JSON-format parameters as in WebAPI. {“command” : “add scenario”, ...}

For details of the settings in the data section, refer to "Appendix H Details of OpenFlow Message Supported for CLI Command".

The response messages are shown below.

Type		Description
Setting CLI command	Normal request	Returns nothing.
	Abnormal request	Sends an error message.
Display CLI command	Normal request	Set and send JSON-format display data as in WebAPI in the data section of the Experimenter message.
	Abnormal request	Sends an error message.

The conditions of error messages are shown below.

Error type	Error code	Condition
OFPET_BAD_REQUEST	OFPBRC_BAD_EXPERIMENTER	Experimenter field is other than 0x000000091 (Anritsu)
OFPET_BAD_REQUEST	OFPBRC_BAD_EXP_TYPE	Not supported because the exp_type field is arbitrary in this equipment
OFPET_EXPERIMENT_ERROR	None	A request that results in an error of the CLI command is received*

* The format of an Experimenter error message is shown below.

Item	Type	Name	Description
Header	struct ofp_header	header	OpenFlow protocol header
Type	uint16_t	type	OFPET_EXPERIMENTER
Extension type	uint16_t	exp_type	Arbitrary in this equipment
Extended ID	uint32_t	experimenter	IEEE OUI (Anritsu : 0x00000091)
Data section	uint8_t	data[0]	Returns an error message of the command in this equipment. Set the following JSON-format. {“error” : “CLI error message”}

- OFPT_FEATURES_REQUEST
- OFPT_FEATURES_REPLY

This message is supported in this equipment.

This message is used to exchange the OpenFlow function with the OpenFlow controller at the time of connection.

The OpenFlow controller sends a Features request message to this equipment when a session is established.

This equipment returns a Features response message to the OpenFlow controller.

Message type

OFPT_FEATURES_REQUEST(5)

OFPT_FEATURES_REPLY(6)

The items of a response message are shown below.

Item	Support	Description
Data path	OK	Unique ID for identifying the switch The lower 48 bits are the MAC address of the management port of WSX The upper 16 bits are 0x0000
Buffer	OK	The number of packets that can be stored in the buffer area simultaneously (fixed at 256)
Table count	OK	The number of flow tables supported by the switch (fixed at 1)
Connection type	OK	The type of connection from the switch to the controller (0 for main connection, other than 0 for auxiliary connection) (fixed at 0)
Function	OK	Set function flags (only port statistics information is supported) (fixed at 0x00000004)
OFPC_FLOW_STATS	N/A	Flow statistics information
OFPC_TABLE_STATS	N/A	Table statistics information
OFPC_PORT_STATS	OK	Port statistics information
OFPC_GROUP_STATS	N/A	Group statistics information
OFPC_IP_REASM	N/A	Reassemble the IP fragment
OFPC_QUEUE_STATS	N/A	Queue statistics information
OFPC_PORT_BLOCKED	N/A	Detects the topology loop and block port to prevent looping of the packet

There is no error handling. A response as defined above is always sent.

- OFPT_GET_CONFIG_REQUEST
- OFPT_GET_CONFIG_REPLY
- OFPT_SET_CONFIG

This message is supported in this equipment.

For setting, this equipment does not return a response message because it is not required to return one.

For acquisition, this equipment returns a GET_CONFIG response message to the OpenFlow controller.

To check if the OpenFlow controller is set correctly, a GET_CONFIG request message is sent.

Message type

OFPT_SET_CONFIG (9)

OFPT_GET_CONFIG_REQUEST (7)

OFPT_GET_CONFIG_REPLY(8)

The items of the SET_CONFIG message and GET_CONFIG response message are shown below.

Item	Support	Description
Flags	△	Set the process of the IP fragment The bitmaps are as follows: OFPC_FRAG_NORMAL (do not process fragment) OFPC_FRAG_DROP (discard) OFPC_FRAG_REASM (reassemble) OFPC_FRAG_MASK This equipment supports OFPC_FRAG_NORMAL only.
Miss send length	△	Define the number of bytes of each packet sent to the OpenFlow controller. Setting range: 0 to 65535 (0xFFFF)

The setting items above are related to the IP fragment process and byte length of messages sent from this equipment to the OpenFlow controller. The settings of the SET_CONFIG message are not used because this equipment does not support Packet-in messages. They are ignored in the internal process of this equipment.

The conditions of error messages are shown below.

Error type	Error code	Support	Condition
OFPET_SWITCH_CONFIG_FAILED(10)	OFPSCFC_BAD_FLAGS(0)	OK	Flags other than OFPC_FRAG_NORMAL are received
	OFPSCFC_BAD_LEN(1)	N/A	Invalid Miss send length is received
	OFPSCFC_EPERM(2)	N/A	Authority error Not supported because there is no Role in this equipment

➤ OFPT_FLOW_MOD

This message is supported in this equipment.

This message is used to add/change flow entries from the OpenFlow controller.

This equipment supports the addition and deletion of filters.

For details of the settings, refer to "Appendix H Details of OpenFlow Message Supported for CLI Command".

Message type

OFPT_FLOW_MOD (14)

The fields are shown below.

Field	Support	Description
cookie	N/A	
cookie_mask	N/A	
table_id	OK	A value other than 0 results in an error.
idle_timeout	OK	A value other than 0 results in an error.
hard_timeout	OK	A value other than 0 results in an error.
priority	N/A	
buffer_id	N/A	OFPP_NO_BUFFER(0xFFFFFFFF)
out_port	N/A	OFPP_ANY(0xFFFFFFFF)
out_group	N/A	OFPG_ANY(0xFFFFFFFF)

The command fields are shown below.

Item	Support	Description
OFPPC_ADD	OK	Addition (corresponding to the add filter of this equipment)
OFPPC_MODIFY	N/A	Correction (for all matching entries)
OFPPC_MODIFY_STRICT	N/A	Correction (for matching entries including wild cards and priority)
OFPPC_DELETE	N/A	Deletion (for all matching entries)
OFPPC_DELETE_STRICT	OK	Deletion (for matching entries including wild cards and priority) (corresponding to delete filter of this equipment)

The flag fields are shown below.

Item	Support	Description
OFPPF_SEND_FLOW_REM	N/A	Whether to enable/disable transmission of the Flow Remove message to the controller when a flow entry disappears due to timeout
OFPPF_CHECK_OVERLAP	N/A	Whether to generate an error in the case of flow table priority conflict
OFPPF_RESET_COUNTS	N/A	Count reset
OFPPF_NO_PKT_COUNTS	N/A	Packet count disabled
OFPPF_NO_BYT_COUNTS	N/A	Byte count disabled

The Flow Match fields are shown below.

Item	Support	Description
OFPXMT_OFB_IN_PORT	OK	Input port of the OpenFlow switch
OFPXMT_OFB_IN_PHY_PORT	N/A	Input physical port of the OpenFlow switch IN_PORT is required
OFPXMT_OFB_METADATA	N/A	Meta data of the flow table
OFPXMT_OFB_ETH_DST	N/A	Destination Ethernet address
OFPXMT_OFB_ETH_SRC	N/A	Source Ethernet address
OFPXMT_OFB_ETH_TYPE	OK	Ethernet frame type
OFPXMT_OFB_VLAN_VID	N/A	VLAN ID
OFPXMT_OFB_VLAN_PCP	N/A	Priority of VLAN VLAN_VID != NONE is required
OFPXMT_OFB_IP_DSCP	N/A	DSCP ETH_TYPE=0x0800 or ETH_TYPE=0x86dd is required
OFPXMT_OFB_IP_ECN	N/A	ECN ETH_TYPE=0x0800 or ETH_TYPE=0x86dd is required
OFPXMT_OFB_IP_PROTO	OK	IP protocol number ETH_TYPE=0x0800 or ETH_TYPE=0x86dd is required
OFPXMT_OFB_IPV4_SRC	OK	Source IPv4 address ETH_TYPE=0x0800 is required
OFPXMT_OFB_IPV4_DST	OK	Destination IPv4 address ETH_TYPE=0x0800 is required
OFPXMT_OFB_TCP_SRC	OK	Source TCP port number IP_PROTO=6 is required
OFPXMT_OFB_TCP_DST	OK	Destination TCP port number IP_PROTO=6 is required
OFPXMT_OFB_UDP_SRC	OK	Source UDP port number IP_PROTO=17 is required
OFPXMT_OFB_UDP_DST	OK	Destination UDP port number IP_PROTO=17 is required
OFPXMT_OFB_SCTP_SRC	N/A	Source SCTP port number IP_PROTO=132 is required

Item	Support	Description
OFPXMT_OFB_SCTP_DST	N/A	Destination SCTP port number IP_PROTO=132 is required
OFPXMT_OFB_ICMPV4_TYPE	N/A	Type of IPv4ICMP IP_PROTO=1 is required
OFPXMT_OFB_ICMPV4_CODE	N/A	Code of IPv4ICMP IP_PROTO=1 is required
OFPXMT_OFB_ARP_OP	N/A	OP code of ARP ETH_TYPE=0x0806 is required
OFPXMT_OFB_ARP_SPA	N/A	Source IPv4 address of ARP ETH_TYPE=0x0806 is required
OFPXMT_OFB_ARP_TPA	N/A	Destination IPv4 address of ARP ETH_TYPE=0x0806 is required
OFPXMT_OFB_ARP_SHA	N/A	Source hardware address of ARP ETH_TYPE=0x0806 is required
OFPXMT_OFB_ARP_THA	N/A	Destination hardware address of ARP ETH_TYPE=0x0806 is required
OFPXMT_OFB_IPV6_SRC	OK	Source IPv6 address ETH_TYPE=0x86dd is required
OFPXMT_OFB_IPV6_DST	OK	Destination IPv6 address ETH_TYPE=0x86dd is required
OFPXMT_OFB_IPV6_FLABEL	N/A	IPv6 flow label ETH_TYPE=0x86dd is required
OFPXMT_OFB_ICMPV6_TYPE	N/A	ICMPv6 type IP_PROTO=58 is required
OFPXMT_OFB_ICMPV6_CODE	N/A	ICMPv6 code IP_PROTO=58 is required
OFPXMT_OFB_IPV6_ND_TARGET	N/A	Target address of Neighbor Discovery ICMPV6_TYPE=135 or ICMPV6_TYPE=136 is required
OFPXMT_OFB_IPV6_ND_SLL	N/A	Source Link-Layer of Neighbor Discovery ICMPV6_TYPE=135 is required
OFPXMT_OFB_IPV6_ND_TLL	N/A	Target Link-Layer of Neighbor Discovery ICMPV6_TYPE=136 is required
OFPXMT_OFB_MPLS_LABEL	N/A	MPLS label ETH_TYPE=0x8847 or ETH_TYPE=0x8848 is required
OFPXMT_OFB_MPLS_TC	N/A	MPLS traffic class ETH_TYPE=0x8847 or ETH_TYPE=0x8848 is required
OFPXMT_OFB_MPLS_BOS	N/A	Bos (bottom of stack) bit included in the first MPLS shim header ETH_TYPE=0x8847 or ETH_TYPE=0x8848 is required
OFPXMT_OFB_PBB_ISID	N/A	I-SID included in the first PBB service instance tag ETH_TYPE=0x88e7 is required
OFPXMT_OFB_TUNNEL_ID	N/A	Meta data associated with the logical port
OFPXMT_OFB_IPV6_EXTHDR	N/A	pseudo field for IPv6 extended header ETH_TYPE=0x86dd is required

The Instruction types are shown below.

Item	Support	Description
FPIT_GOTO_TABLE	N/A	Inherit the process to the specified flow table
FPIT_WRITE_METADATA	N/A	Set meta data that can be referred to in the subsequent tables
FPIT_WRITE_ACTIONS	N/A	Add actions specified in the current action set
FPIT_APPLY_ACTIONS	OK	Immediately apply the specified actions without changing the action set
FPIT_CLEAR_ACTIONS	N/A	Delete all the actions in the current action set
FPIT_METER	N/A	Apply packets to the specified meter
FPIT_EXPERIMENTER	N/A	Area for experimenters

The conditions of error messages are shown below.

Error type	Error code	Support	Condition
OFPET_FLOW_MODIFIED(5)	OFPFMFC_UNKNOWN(0)	N/A	Unknown error
	OFPFMFC_TABLE_FULL(1)	N/A	The maximum number of filters are registered
	OFPFMFC_BAD_TABLE_ID(2)	OK	Table ID is other than 0
	OFPFMFC_OVERLAP(3)	N/A	Overlap error when the CHECK_OVERLAP flag is set
	OFPFMFC_EPERM(4)	N/A	Authority error
	OFPFMFC_BAD_TIMEOUT(5)	OK	Unsupported idle/hard timeout is specified (other than 0)
	OFPFMFC_BAD_COMMAND(6)	OK	Unsupported command is specified
	OFPFMFC_BAD_FLAGS(7)	OK	Unsupported flags are specified

Error type	Error code	Support	Condition
OFPET_BAD_MATCH(4)	OFPBMC_BAD_TYPE(0)	OK	Unsupported match type is specified (other than OFPMT_OXM)
	OFPBMC_BAD_LEN(1)	OK	length error of the match field
	OFPBMC_BAD_TAG(2)	N/A	Unsupported tag/encap
	OFPBMC_BAD_DL_ADDR_MASK(3)	N/A	Data link address mask is not supported
	OFPBMC_BAD_NW_ADDR_MASK(4)	N/A	Network address mask is not supported
	OFPBMC_BAD_WILDCARDS(5)	N/A	Unsupported mask or combination of omission
	OFPBMC_BAD_FIELD(6)	OK	Unsupported Flow Match field is specified
	OFPBMC_BAD_VALUE(7)	N/A	Unsupported value is specified
	OFPBMC_BAD_MASK(8)	N/A	Unsupported mask is specified
	OFPBMC_BAD_PREREQ(9)	OK	Required field is not specified
	OFPBMC_DUP_FIELD(10)	△	Duplicate Flow Match field
	OFPBMC_EPERM(11)	N/A	Authority error

Error type	Error code	Support	Condition
OFPET_BAD_INSTRUCTION(3)	OFPBIC_UNKNOWN_INST(0)	N/A	Unknown instruction
	OFPBIC_UNSUP_INST(1)	OK	Unsupported instruction is received
	OFPBIC_BAD_TABLE_ID(2)	N/A	Table ID is other than 0
	OFPBIC_UNSUP_METADATA(3)	N/A	Meta data value not supported by the data path
	OFPBIC_UNSUP_METADATA_MASK(4)	N/A	Meta data mask value not supported by the data path
	OFPBIC_BAD_EXPERIMENTER(5)	N/A	Experimenter field is other than 0x00000091 (Anritsu)
	OFPBIC_BAD_EXP_TYPE(6)	N/A	Not supported because the exp_type field is arbitrary in this equipment
	OFPBIC_BAD_LEN(7)	N/A	length error of instruction
	OFPBIC_EPERM(8)	N/A	Authority error

Error type	Error code	Support	Condition
OFPET_BAD_ACTION(2)	OFPBAC_BAD_TYPE (0)	OK	Unsupported action type is specified (other than Experimenter)
	OFPBAC_BAD_LEN (1)	OK	length error of the action field
	OFPBAC_BAD_EXPERIMENTER (2)	OK	Unknown Experimenter ID
	OFPBAC_BAD_EXP_TYPE (3)	N/A	Unknown action for Experimenter ID
	OFPBAC_BAD_OUT_PORT (4)	N/A	Invalid output port
	OFPBAC_BAD_ARGUMENT (5)	N/A	Invalid action argument
	OFPBAC_EPERM (6)	N/A	Permission error
	OFPBAC_TOO_MANY (7)	N/A	To many actions to handle
	OFPBAC_BAD_QUEUE (8)	N/A	Invalid output queue
	OFPBAC_BAD_OUT_GROUP (9)	N/A	Invalid group ID of forward action
	OFPBAC_MATCH_INCONSISTENT (10)	N/A	Cannot apply action to this match
	OFPBAC_UNSUPPORTED_ORDER (11)	N/A	Invalid order of Apply-Action action list
	OFPBAC_BAD_TAG (12)	N/A	Action is using unsupported Tag/encap
	OFPBAC_BAD_SET_TYPE (13)	N/A	Invalid type of SET_FIELD action
	OFPBAC_BAD_SET_LEN (14)	N/A	Invalid length of SET_FIELD action
OFPBAC_BAD_SET_ARGUMENT (15)	N/A	Invalid argument of SET_FIELD action	

- OFPT_MULTIPART_REQUEST
- OFPT_MULTIPART_REPLY

This message is supported in this equipment.

Message type	Multipart type
OFPT_MULTIPART_REQUEST(18)	OFPPMP_PORT_STATS (4)
OFPT_MULTIPART_REPLY(19)	

This message is used to acquire statistics information about the port counter.

The frame format of request messages is shown below.

Specify a port number in the port_no field.

port_no	Support	Description
1	OK	Management port (mgmt0)
2	OK	Network port (1/1)
3	OK	Network port (1/2)
4	OK	Network port (1/3)
5	OK	Network port (1/4)
0, 6 to 0xfffffeff	N/A	Out of range
OFPP_MAX (0xfffffff0)	△	Reserved port Do not generate an error. The data section of the response message is empty.
OFPP_IN_PORT (0xfffffff8)	△	Same as above
OFPP_TABLE (0xfffffff9)	△	Same as above
OFPP_NORMAL (0xffffffa)	△	Same as above
OFPP_FLOOD (0xffffffb)	△	Same as above
OFPP_ALL (0xffffffc)	△	Same as above
OFPP_CONTROLLER (0xffffffd)	△	Same as above
OFPP_LOCAL (0xffffffe)	OK	Respond with statistics information of the reserved port and local port.
OFPP_ANY (0xfffffff)	OK	Respond with statistics information of the reserved port and all ports.

The statistics information items (body section as shown above) are shown below. (as in the show counter command)

Item	Support	Description
port_no	OK	Port number. Set OFPP_ANY to specify all ports.
rx_packets	OK	The total number of packets received
tx_packets	OK	The total number of packets sent
rx_bytes	OK	The total number of bytes of packets received
tx_bytes	OK	The total number of bytes of packets sent
rx_dropped	OK	The total number of packets dropped during reception
tx_dropped	N/A	The total number of packets dropped during transmission
rx_errors	OK	The total number of packets resulting in a reception error
tx_errors	N/A	The total number of packets resulting in a transmission error
rx_frame_err	N/A	The number of times a frame allocation error occurred during reception
rx_over_err	N/A	The number of times a packet was lost due to overrun during reception
rx_crc_err	N/A	The number of times a CRC error occurred during reception
collisions	OK	The number of times a collision occurred in the Internet layer
duration_sec	△	Time after the port was enabled (sec) Not supported for the management port (mgmt0) and Network ports (1/1 to 1/4)
duration_nsec	△	Digits below seconds of the time after the port was enabled (nsec) Not supported for the management port (mgmt0) and Network ports (1/1 to 1/4)

Set ALL 0xFF for unsupported items. (32 bits for port_no and duration, 64 bits for others)

The conditions of error messages are shown below.

Error type	Error code	Condition
OFPET_BAD_REQUEST(1)	OFPBRC_BAD_PORT(11)	Port number is out of range
OFPET_BAD_REQUEST(1)	FPBRC_MULTIPART_BUFFER_OVERFLOW(13)	OFPMPPF_REQ_MORE (1) is set in the flags field *

* If OFPMPPF_REQ_MORE (1) is set in the flags field of the request message, the error above occurs. Do not set the flag above.

Item	Support	Description
Advertised	N/A	Advertised characteristics and functions (the bitmaps are the same as Current) Set 0 to all bits because it is not supported.
Supported	N/A	Supported characteristics and functions (the bitmaps are the same as Current) Set 0 to all bits because it is not supported.
Peer	N/A	Characteristics and functions advertised by peer (OpenFlow controller) (the bitmaps are the same as Current) Set 0 to all bits because it is not supported.
Curr speed	OK	Current speed (unit: kb/s) Set from Oper speed (communication speed) of the show port command.
Max speed	OK	Maximum speed OFPP_LOCAL port: 0 Management port: 1G = 1,000,000[kb/s] Set from Port type (port type) of the show port command.

The conditions of error messages are shown below.

Error type	Error code	Condition
OFPET_BAD_REQUEST (1)	FPBRC_MULTIPART_BUFFER_OVERFLOW (13)	OFPPMPF_REQ_MORE (1) is set in the flags field *

* If OFPPMPF_REQ_MORE (1) is set in the flags field of the request message, the error above occurs. Do not set the flag above.

Message type	Multipart type
OFPT_MULTIPART_REQUEST(18)	OFPMMP_EXPERIMENTER (0xffff)
OFPT_MULTIPART_REPLY(19)	

This message is used to respond to setting/display CLI commands.

The Experimenter message also supports setting/display CLI commands.

The data structure after the extended header (struct ofp_experimenter_multipart_header) is shown below.

Item	Type	Name	Description
Extended ID	uint32_t	experimenter	IEEE OUI (Anritsu : 0x00000091)
Extension type	uint32_t	exp_type	Arbitrary ID (ignored)
Data section	uint8_t	data[0]	Set JSON-format parameters as in WebAPI. {“command” : “show scenario counter”, ...}

For details of the settings in the data section, refer to "Appendix H Details of OpenFlow Message Supported for CLI Command".

To acquire statistics information about the scenario counter, set and send the following items in the JSON format in the data section of the response message. (As in show scenario counter)

Item	Display	Description
Scenario	OK	Scenario index and scenario name The scenario index of port scenario is displayed as 40001 for Port 1 and 40002 for Port 2
Rate Control Unit	N/A	Settings about bandwidth control
Default Queue	N/A	Settings about default queue
Attached Filters	N/A	Filter name of the filter added to the scenario
Rx Octets	OK	The number of bytes of packets received
Rx Packets	OK	The number of packets received
Tx Octets	OK	The number of bytes of packets sent
Tx Packets	OK	The number of packets sent
Discard Octets	OK	The number of bytes of packets discarded
Discard Packets	OK	The number of packets discarded

For other display commands, set and send JSON-format display data as in WebAPI in the data section of the response message.

The response messages are shown below.

Type		Description
Setting CLI command	Normal request	Sends a response message of Experimenter Multipart. The data section is empty.
	Abnormal request	Sends an error message.
Display CLI command	Normal request	Set and send JSON-format display data as in WebAPI in the data section of the response message.
	Abnormal request	Sends an error message.

The conditions of error messages are shown below.

Error type	Error code	Condition
OFPET_BAD_REQUEST	OFPBRC_BAD_EXPERIMENTER	Experimenter field is other than 0x00000091 (Anritsu)
OFPET_BAD_REQUEST	OFPBRC_BAD_EXP_TYPE	Not supported because the exp_type field is arbitrary in this equipment
OFPET_BAD_REQUEST	FPBRC_MULTIPART_BUFFER_OVERFLOW	OFPMMPF_REQ_MORE (1) is set in the flags field *1
OFPET_EXPERIMENTER	None	A request that results in an error of the CLI command is received *2

*1 If OFPMMPF_REQ_MORE (1) is set in the flags field of the request message, the error above occurs. Do not set the flag above.

*2 The format of an Experimenter error message is shown below.

Item	Type	Name	Description
Header	struct ofp_header	header	OpenFlow protocol header
Type	uint16_t	type	OFPET_EXPERIMENTER
Extension type	uint16_t	exp_type	Arbitrary in this equipment (ignored)
Extended ID	uint32_t	experimenter	IEEE OUI (Anritsu : 0x00000091)
Data section	uint8_t	data[0]	Returns an error message of the command in WSX. Set the following JSON-format. {"error" : "CLI error message"}

- OFPT_BARRIER_REQUEST
- OFPT_BARRIER_REPLY

This message is supported in this equipment.

Message type

OFPT_BARRIER_REQUEST (20)

OFPT_BARRIER_REPLY(21)

This message is used to check whether the processing of the controller's request is completed.

The OpenFlow controller sends a Barrier request message to check whether the processing of messages that have been received by this equipment has been completed. This equipment sends a Barrier response message after the processing of messages that have been received has been completed.

- OFPT_ROLE_REQUEST
- OFPT_ROLE_REPLY

This message is supported in this equipment.

Message type

OFPT_ROLE_REQUEST (24)

OFPT_ROLE_REPLY(25)

This message is used to notify of the roles of the OpenFlow controller.

The OpenFlow controller sends a Role request message to notify this equipment of the roles. This equipment sends a Role response message.