PureFlow WSX

Unified Network Controller
NF7600 series
Configuration Guide
Traffic Shaping Edition

Third 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 NF7600 series Unified Network Controller Operation Manual (NF7600-W021E). Please also refer to this document before using the equipment.**
- **Keep this manual with the equipment.**

ANRITSU NETWORKS CO., LTD.

Document No.: NF7600-W023E-3.0

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 (Third Edition)

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.

About This Manual

This operation manual describes how to configure and use the software running on 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
- NF7603A
- NF7604A

The manual of the NF7601A consists of the following three manuals. This document is <3>.

<1> Operation Manual　Traffic Shaping Edition (NF7600-W021E)

Describes in detail the installation and handling in this device.

<2> Command Reference　Traffic Shaping Edition (NF7600-W022E)

Describes in detail the commands used in this device.

<3> Configuration Guide　Traffic Shaping Edition (NF7600-W023E)

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

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 Top Counter 2-3
2.7 WebAPI 2-3
2.8 RADIUS 2-3
2.9 Network Bypass 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 3-11

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 Maximum Packet Length Setting 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 7-10
7.4 Configuration Examples 7-11
7.5 Checking Settings and States 7-18

Chapter 8 Traffic Control 8-1

8.1 Overview 8-2
8.2 Hierarchical Shaping 8-3
8.3 Filters and Scenarios 8-4
8.4 Setting Procedure 8-12
8.5 How to Set a Rule List 8-22
8.6 Configuration Examples 8-25
8.7 Advanced Settings 8-30

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

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 12-4

Chapter 13 Top Counter 13-1

13.1 Overview 13-2
13.2 Display Unit of the Top Counter 13-2
13.3 Measurement Range of the Top Counter 13-3
13.4 Traffic Counter 13-3
13.5 Measuring Traffic at Specific Application Ports 13-4
13.6 Operation Command List 13-4
13.7 Operation Procedure 13-5
13.8 Operation Example 13-6
13.9 Cautions 13-8

Chapter 14 WebAPI 14-1

14.1 Overview 14-2
14.2 Communication Protocol 14-3
14.3 HTTP Methods 14-3
14.4 JSON Format 14-4
14.5 API List 14-5
14.6 Common Error Messages 14-6
14.7 List of Error Messages 14-7

Chapter 15 RADIUS 15-1

15.1 Overview 15-2
15.2 Controlling Login Authentication 15-3
15.3 Controlling Login Mode 15-3
15.4 Setting Up the RADIUS Feature 15-4
15.5 RADIUS Server Settings 15-5

Chapter 16 Downloading and Uploading Data 16-1

16.1 Downloading/Uploading Software 16-2
16.2 Downloading the Software Update Patch 16-6
16.3 Downloading/Uploading Configuration Data 16-7
16.4 Restarting the Software 16-11

Chapter 17 Network Bypass Function 17-1

17.1 Overview 17-2
17.2 Setting and Checking the Function 17-3
17.3 Precautions 17-6

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

Appendix F Details of WebAPI F-1

Appendix G WebAPI Sample Programs G-1

(Blank page)

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
- Top counter
- WebAPI
- RADIUS
- Network Bypass Function

(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 Top Counter 2-3
2.7 WebAPI 2-3
2.8 RADIUS 2-3
2.9 Network Bypass Function 2-3

2.1 Traffic Control

Packet loss or communication delays due to insufficient line bandwidth might lower the efficiency of mission-critical tasks such as voice communication and video conferencing, causing major problems. To protect such mission-critical traffic from insufficient line bandwidth or communication delays, the line bandwidth must be split by site, user, or application to allocate the necessary bandwidth, and priority control of traffic is required. This device is installed on the network communication path and performs traffic control such as splitting the line bandwidth to guarantee the minimum bandwidth in the allocated bandwidth and restricting the maximum bandwidth.

For details about traffic control, 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 Top Counter

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

For details about the top counter, see Chapter 13 “Top Counter”.

2.7 WebAPI

The WebAPI feature is used to configure the traffic control feature of the system via HTTP (Hypertext Transfer Protocol: RFC2616).

For details about WebAPI, see Chapter 14 “WebAPI”.

2.8 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 15 “RADIUS”.

2.9 Network Bypass Function

NF7603A and NF7604A 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".

(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 (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 or Network 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 "Command Reference Traffic Shaping Edition (NF7600-W022E)".

(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 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.

(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 or Network 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.

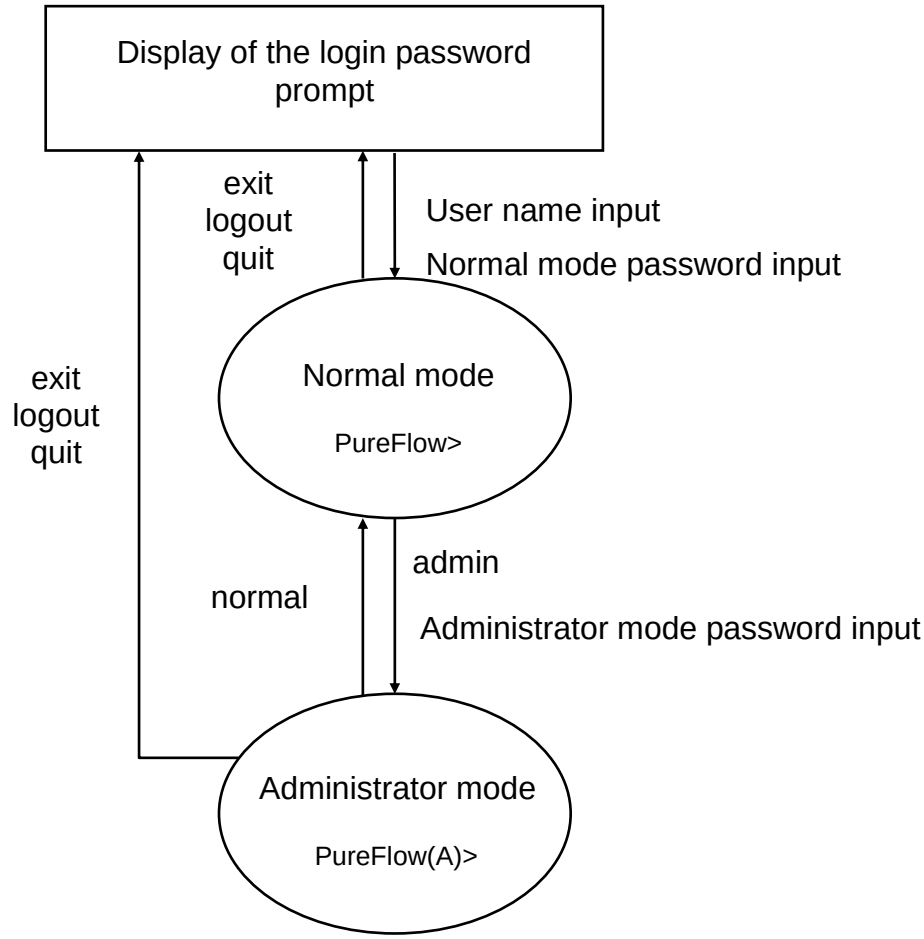
3.2 Command Structure

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	20120630101010

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:

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 media 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-L601A (Traffic Shaping Software)


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

Slot 1 boot up complete
     Medium type 10GBase-R 2 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)>
```

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	-	2 minutes 30 seconds
100 scenarios 100 filters	5 seconds	2 minutes 40 seconds

* For a description of filter, scenario, and rule list settings, 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:.

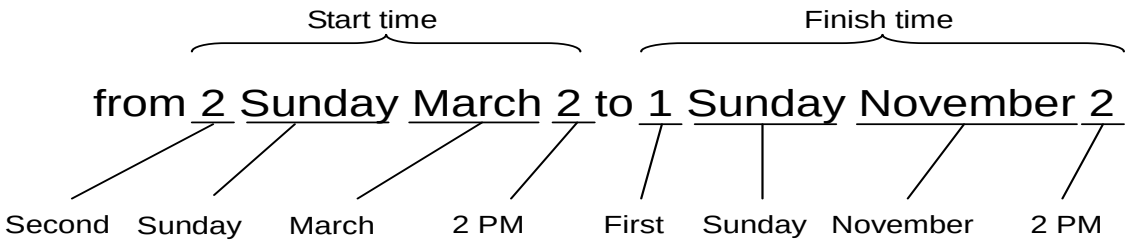
set date <yyyymmddhhmmss>	Sets the date and time.
set timezone <hours-offset> [<minutes-offset>]	Sets the time zone offset from the UTC (Coordinated Universal Time). The default value is +9 [hours] 0 [minutes].
set summertime from <week> <day> <month> <hh> to <week> <day> <month> <hh> [offset]	Sets the 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.

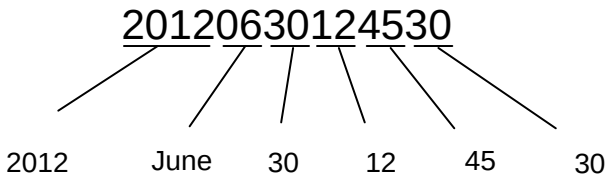
```
PureFlow(A)> set timezone +9
PureFlow(A)> set summertime from 2 Sunday March 2 to 1 Sunday November 2
PureFlow(A)> set date 20120630124530
PureFlow(A)> show date
May 18 2005(Mon) 12:45:32
UTC Offset     :  +09:00
Summer Time    :  From Second Sunday March 02:00
                  To     First Sunday 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)> 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.
Time correction is performed when the time of inquiry interval.

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:
Retype the new Password:
```

Enter the password to be set.

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

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

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.

Ident

The identifier of the program that recorded the system log information.

PID

The identifier of the program 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.

Priority = Facility × 8 + 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
PureFlow(A)> set syslog facility fcpu 19
```

Sets facility of the control CPU to 18.

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 mcessage 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 × 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:

MAC Address

Indicates the MAC address of the device.

Chassis Model Name

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

NF7601A: PureFlow WSX
NF7603A: PureFlow WSX
NF7604A: PureFlow WSX

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 and NF7604A.

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 and NF7604A.

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.

MAC Addres                      : 00-00-91-12-34-56

Chassis Model Name              : NF7603A
Chassis Serial Number           : 1234567890

Control Module Version          : 01A
Shaper Module Version           : 00A
Bypass Module Version           : 00A
Software Version                : 1.1.1
Software License                : NF7600-L601A (Traffic Shaping Software)
Management U-Boot Version       : 1.1.6
Forwarding U-Boot Version       : 1.1.6
MCU-C Version                   : 209
MCU-S Version                   : 209
MCU-B Version                   : 209

Uptime                          : 0 days, 00:27:17
Temperature
  Intake  Temperature           : 32C
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-L614A (Extended Bandwidth 10Gbps)
Updation done.
Enter update scenario command to change port bandwidth.
PureFlow(A)>
PureFlow(A)> show option
    License Key NF7600-L614A available   (Extended Bandwidth 10Gbps)
PureFlow(A)>
```

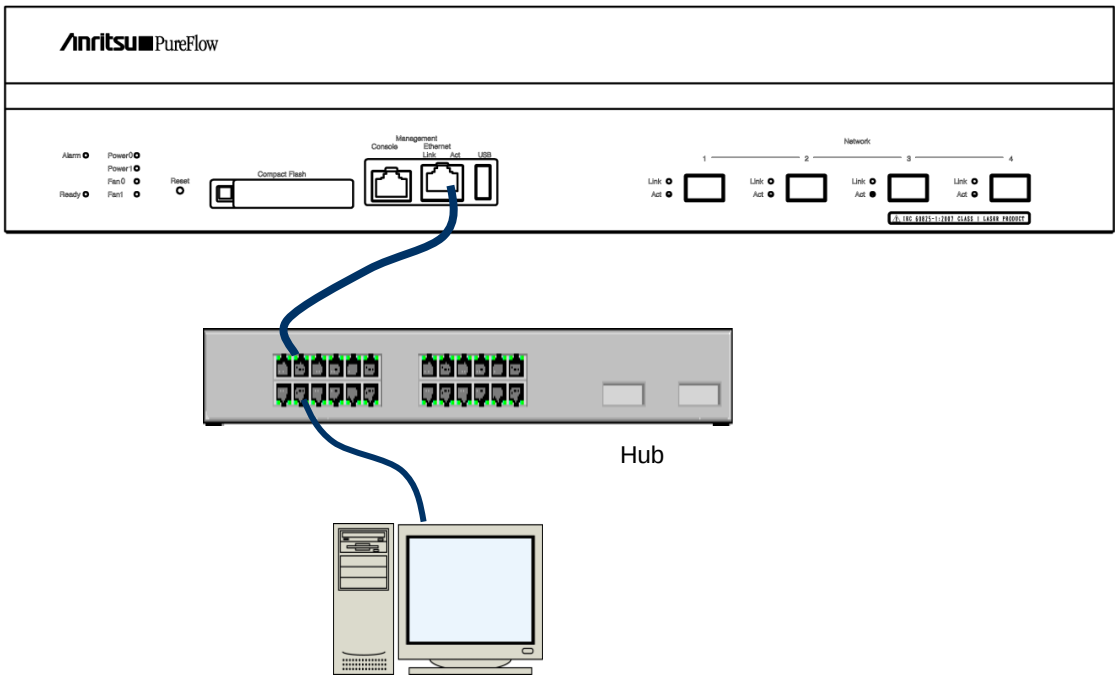
This device has an Ethernet port in the front for remote setting and control over the network. This port, which is a local port for management, is separate from the Network port. This port is a 10/100/1000 BASE-T port that supports Auto-MDIX.

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.)



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 Maximum Packet Length Setting 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 (LC connector)
SFP 1000BASE-SX / 1000BASE-LX (LC connector)
SFP 1000BASE-T (RJ-45 / Auto-MDIX)

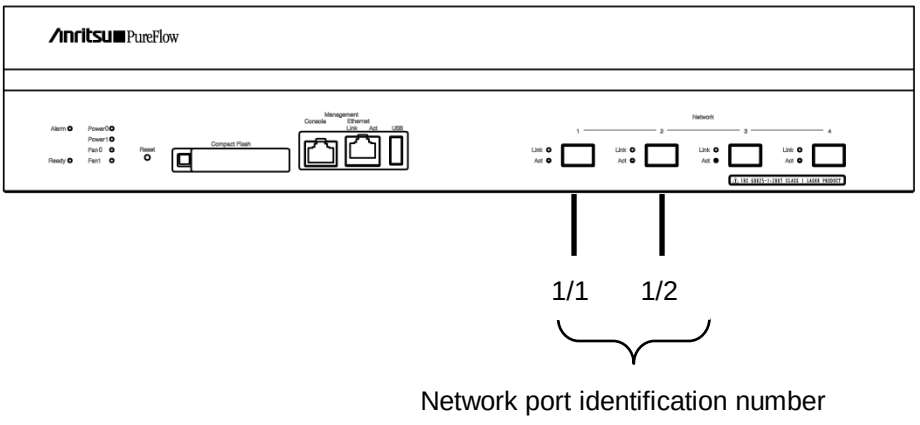
The following settings are available for the Network ports:
Auto negotiation enable/disable (see Notes 2, 4)
Flow control (auto, pause frame send/receive)
Communication speed (10 Mbits/s, 100 Mbits/s, 1 Gbits/s) (see Note 3)
Duplex mode (full, half) (see Note 3)
Maximum packet length (2048 bytes, 10240 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 packet length	2048/10240	2048/10240	2048/10240

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 the PureFlow WSX.

The ports in the slot are numbered as 1/1, 1/2 from the left. Therefore, the ID numbers of the Network ports are as shown below. (Port3 and Port4 are not available.)



For remote setting and control over the network connected to a Network 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:

In PureFlow WSX, 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 setting of maximum packet length is not applied to the system interface. The maximum packet 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
```

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

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

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 Maximum Packet Length Setting

The maximum packet length that can be transfered on the Network port can be changed via CLI. For the maximum packet length, specify the full packet length from the start of the Ethernet header to FCS. Note that the sizes of VLAN Tag and duplex VLAN Tag are excluded. For a packet with VLAN Tag added, the setting value plus 4 bytes can be transfered; for a packet with duplex VLAN Tag added, the setting value plus 8 bytes can be transfered.

set port maxpacketlen {2048 | 10240}	Sets the maximum packet length for the Network port. The default value is 2048 bytes.

The maximum packet length is a common setting value for each Network port. If it is changed to 10240 bytes, valid values for traffic attributes in the bandwidth control settings are changed as follows:

	2048 bytes	10240 bytes
Minimum bandwidth	0, 1 k[bit/s] to 10 G[bit/s] 1 k[bit/s] unit	0, 5 k[bit/s] to 10 G[bit/s] 5 k[bit/s] unit
Peak bandwidth	2 k[bit/s] to 10 G[bit/s] 1 k[bit/s] unit	10 k[bit/s] to 10 G[bit/s] 5 k[bit/s] unit
Buffer size	2 k[Byte] to 100 M[Byte]	11 k[Byte] to 100 M[Byte]

If a configured traffic attribute becomes out of range due to the maximum packet length change, the traffic attribute is automatically rounded within the range. Also, when adding a traffic attribute, a warning message indicating that rounding will be performed is displayed. In either case, bandwidth control is performed after the rounding.

To set the maximum packet length to 10240 bytes, you need to restart the device. To set the maximum packet length to 10240 bytes, execute the following commands:

```
PureFlow(A)> set port maxpacketlen 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.
  minimum value of minimum bandwidth    1k ->   5k
  minimum value of peak bandwidth       2k -> 10k
  bandwidth resolution                  1k ->   5k
  buffer size minimum                   2k -> 11k
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.

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      1000BASE-T        Enabled    Up     Enabled    100M    Full
1/2      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                :1000BASE-T
Admin status             :Enabled
Oper status              :Up
Auto negotiation         :Enabled
Admin speed              :100M
Admin duplex             :Full
Tx Flow control          :Auto
Rx Flow control          :Auto
Admin Max Packet Len     :2048
Admin Max Packet Len     :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
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
system             5                0              10                0

Port        Err Packets     Collision       Discard
-------     -----------     ----------      ----------
 1/1               0                0              0
 1/2               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
```

(Blank page)

Chapter 7 System Interface Settings

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

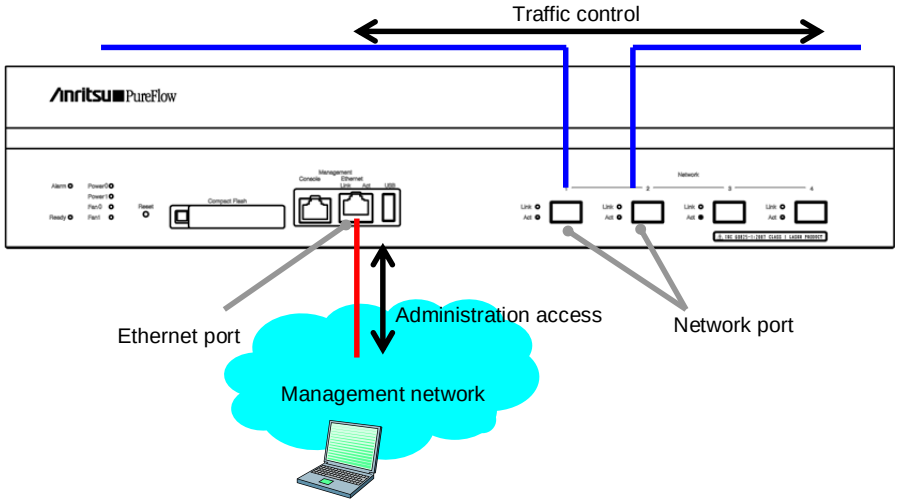
7.1 Overview 7-2
7.2 System Interface Communication 7-3
7.3 System Interface Filter 7-10
7.4 Configuration Examples 7-11
7.5 Checking Settings and States 7-18

7.1 Overview

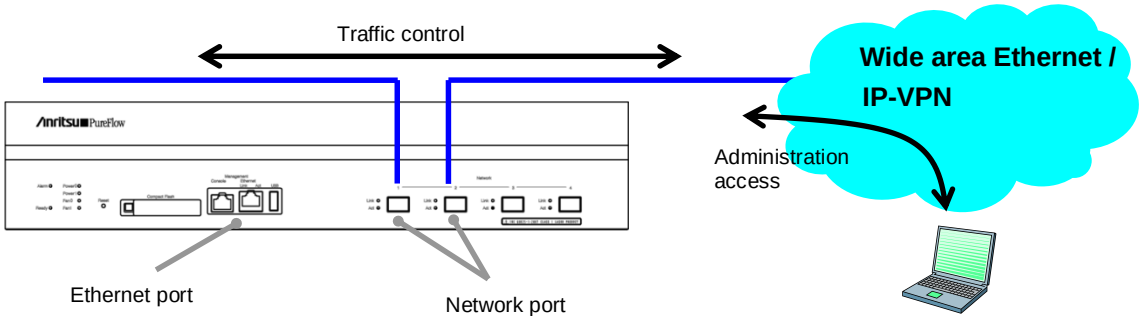
The system interface is an IP network interface for administrators to perform remote access to this device over the network. To control this device remotely, you can use methods such as Telnet and SNMP for setting and state monitoring of this device.

For the system interface, you can select either access via the Ethernet port or access via the Network port, as described below.

(1) Remote management via the Ethernet port
The administrator terminal is located on a separate management network from the network on which traffic control is performed (I/O from the Network port) and performs control via the Ethernet port. This method is effective if you do not want any access from within the network on which traffic control is performed for security reasons.



(2) Remote management via the Network port
The administrator terminal is located within the network on which traffic control is performed and performs control via the Network port. The network configuration can be simplified because no dedicated management network is required.



7.2 System Interface Communication

Communication to the system interface can be performed via the Ethernet port or Network port. When communication is performed via the Ethernet port, communication of packets without VLAN Tag can be performed. When communication is performed via the Network port, the Network port used for communication can be specified (1/1 only, 1/2 only, or All), and communication of packets with or without VLAN Tag or duplex VLAN Tag can be performed. Also, 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	✓	✓
WebAPI	✓	✓
NF7201A Monitoring Manager 2	✓	–
System interface filter	✓	✓

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
51967	TCP	–	Connection to Monitoring Manager 2

Note 1:

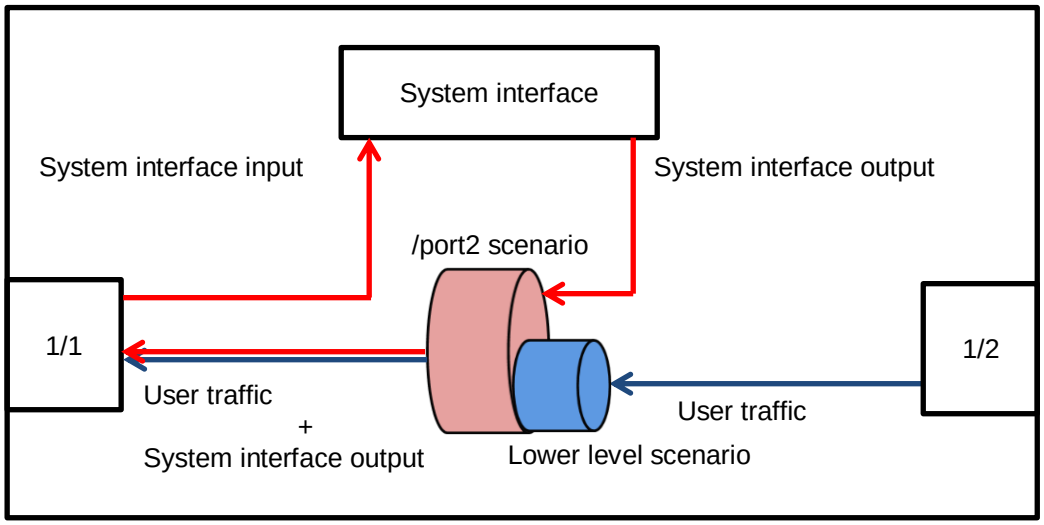
Communication can be performed via either the Ethernet port or Network port.

Note 2:

For communication via the Ethernet port, only packets without VLAN Tag can be communicated.

Note 3:

For communication via the Network port, the bandwidth of the Network port is used during communication to the system interface. The output traffic of the system interface uses the port scenario bandwidth of the opposite port (e.g. "/port2" scenario in the case of port 1/1), and is assigned the top priority of Class 1. Also it is counted by the I/O counter of the relevant scenario. The input traffic does not use the scenario bandwidth and is not counted by the scenario counter.



When allocating a bandwidth to control traffic on the network, consider the bandwidth for system interface communication. For a description of the traffic control settings, see Chapter 8 "Traffic Control".

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 port ethernet set ip system port network in {<slot/port> | all} vid {<VID> | none} [tpid <tpid>] inner-vid {<VID> | none} [inner-tpid <tpid>]	Specifies the communication port (Ethernet port/Network port) of the system interface. Also specifies the following if the Network port is specified as the communication port to the system interface. - Network port ID number (1/1,1/2,all) - VLAN ID (0 to 4094/none), output Tag Protocol ID - Inner-VLAN ID (0 to 4094/none), output Tag Protocol ID The default Network port ID number is "all" (all Network ports). The default VLAN ID and Inner-VLAN ID is "none" (communication of packets without VLAN Tag). Specify the output Tag Protocol ID when specifying Tag Protocol ID of the packet transmitted by the system interface in the case of communication of packets with VLAN Tag or duplex VLAN Tag. 0x8100 is used for both by default. The default communication port is the Ethernet port.
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 port ethernet
PureFlow(A)> set ip system gateway 192.168.10.1
```

To set the IPv4 address (192.168.10.3), subnet mask (255.255.255.0), communication port (Network port (1/1 only)), VLAN ID (10), duplex VLAN Tag none, 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 port network in 1/1 vid 10 inner-vid none
PureFlow(A)> set ip system gateway 192.168.10.1
```

To set the IPv6 address (2001:DB8::1), the prefix length (32), and the default gateway (2001:DB8::FE) to the system interface, execute the following commands:
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:

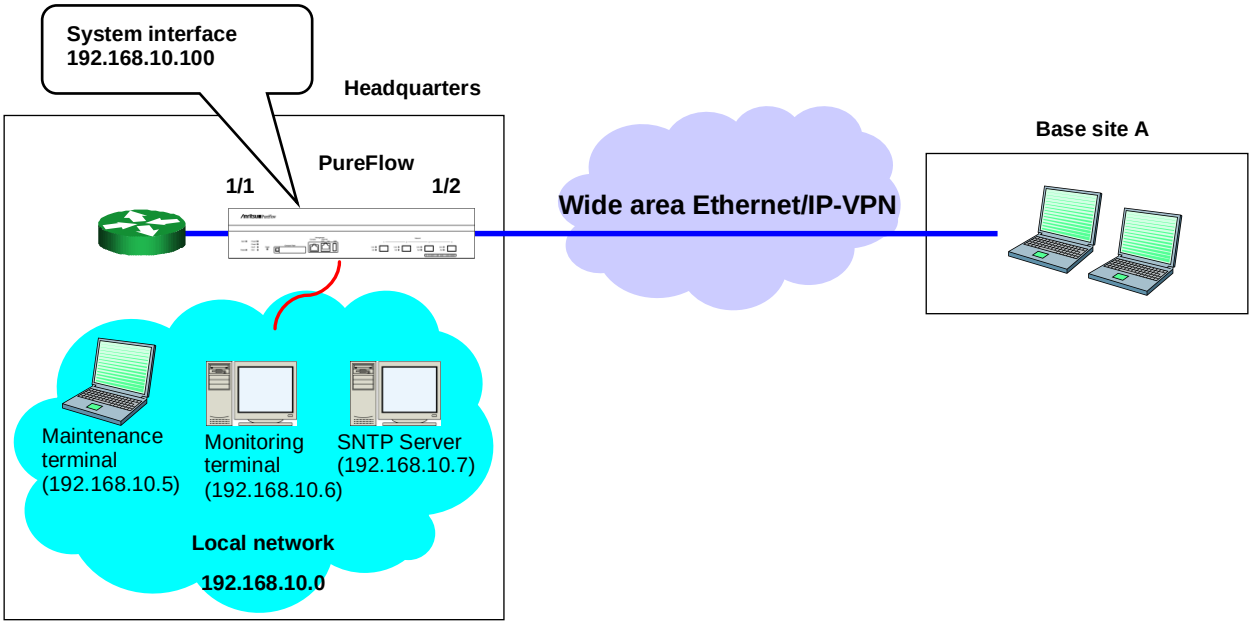
Be careful when setting a system interface filter.
To enable the filter, set permit first, and set deny after that. To delete the filter, delete deny first, and then delete permit. Or delete all by using the “delete ip system filter all” command.

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:

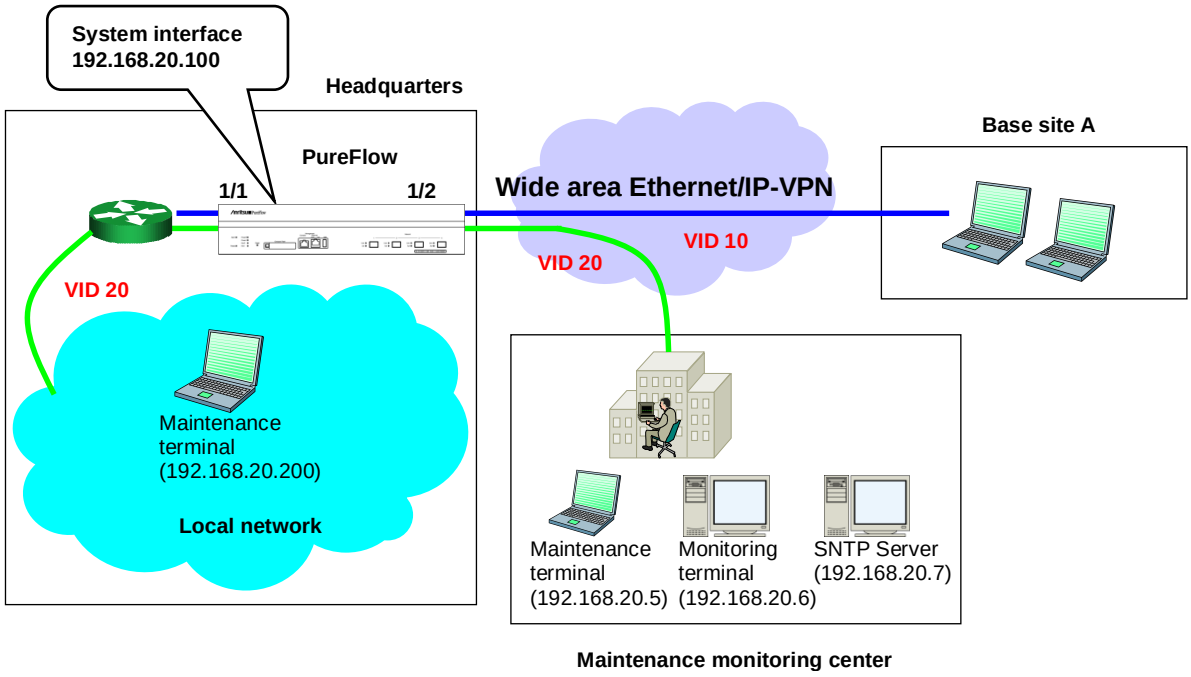
```
PureFlow(A)> add syslog host 192.168.10.6
PureFlow(A)> set syslog host enable
```

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 Wide area Ethernet / IP-VPN network via the Network port (communication of packets without VLAN Tag)

- The network to Site A is VLAN ID 10.
- The network to the maintenance monitoring center is VLAN ID 20.
- The IPv4 address of the system interface is 192.168.20.100, and the subnet mask is 255.255.255.0.
- The default gateway address of the system interface is 192.168.20.1.
- Communication to the system interface is done from all Network ports.
- The IPv4 addresses of the maintenance terminal (CLI, download/upload) are 192.168.20.5 and 192.168.20.200.
- The IPv4 address of the monitoring terminal (SNMP, Syslog) is 192.168.20.6.
- The IPv4 address of the SNTP server is 192.168.20.7.



Execute the following commands.

System interface setting:

```
PureFlow(A)> set ip system 192.168.20.100 netmask 255.255.255.0 up
PureFlow(A)> set ip system port network in all vid 20 inner-vid none
PureFlow(A)> set ip system gateway 192.168.20.1
```

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 192.168.20.6 version v2c
                    community honsya_system_management trap
```

Syslog host setting:

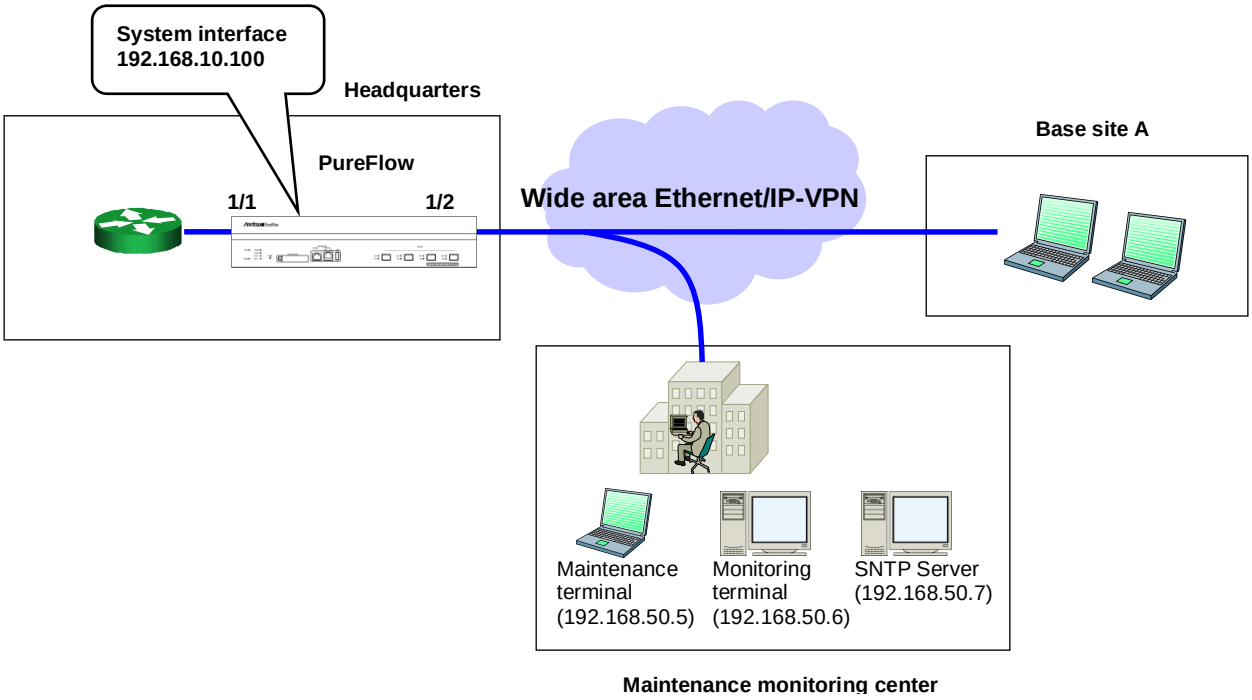
```
PureFlow(A)> set syslog host ip 192.168.20.6
PureFlow(A)> set syslog host enable
```

SNTP server setting:

```
PureFlow(A)> set sntp server 192.168.20.7
PureFlow(A)> set sntp enable
```

Case 3 Performing maintenance and monitoring from the Wide area Ethernet / IP-VPN network via the Network port (communication of packets without VLAN Tag)

- The network to Site A is 192.168.2.0/255.255.255.0.
- The network to the maintenance monitoring center is 192.168.50.0/255.255.255.0.
- 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.
- Communication to the system interface is done only via the Network port 1/2.
- The IPv4 address of the maintenance terminal (CLI, download/upload) is 192.168.50.5.
- The IPv4 address of the monitoring terminal (SNMP, Syslog) is 192.168.50.6.
- The IPv4 address of the SNTP server is 192.168.50.7.



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 port network in 1/2 vid none inner-vid none
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 192.168.50.6 version v2c
                    community honsya_system_management trap
```

Syslog host setting:

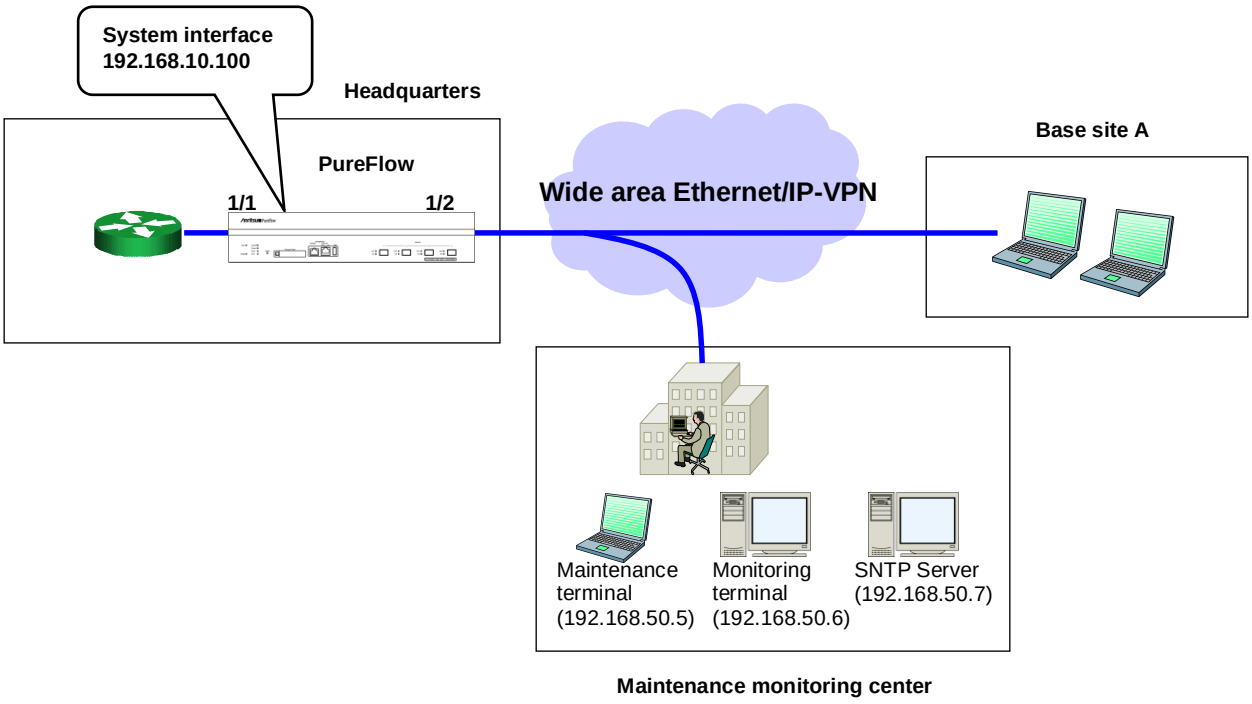
```
PureFlow(A)> set syslog host ip 192.168.50.6
PureFlow(A)> set syslog host enable
```

SNTP server setting:

```
PureFlow(A)> set sntp server 192.168.50.7
PureFlow(A)> set sntp enable
```

Case 4 Performing maintenance and monitoring from the Wide area Ethernet / IP-VPN network via the Network port (only access from the specified network allowed)

- The network to Site A is 192.168.2.0/255.255.255.0.
- The network to the maintenance monitoring center is 192.168.50.0/255.255.255.0.
- 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.
- Communication to the system interface is done only via the Network port 1/2.
- The IPv4 address of the maintenance terminal (CLI, download/upload) is 192.168.50.5.
- The IPv4 address of the monitoring terminal (SNMP, Syslog) is 192.168.50.6.
- The IPv4 address of the SNTP server is 192.168.50.7.
- Communication to the system interface is allowed only from the maintenance monitoring center.



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 port network in 1/2 vid none inner-vid none
PureFlow(A)> set ip system gateway 192.168.10.1
```

System interface filter setting:

```
PureFlow(A)> add ip system filter 10 sip 192.168.50.0/255.255.255.0 permit
PureFlow(A)> add ip system filter 20 deny
```

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 192.168.50.6 version v2c
                    community honsya_system_management trap
```

Syslog host setting:

PureFlow(A)> set syslog host ip 192.168.50.6

PureFlow(A)> set syslog host enable

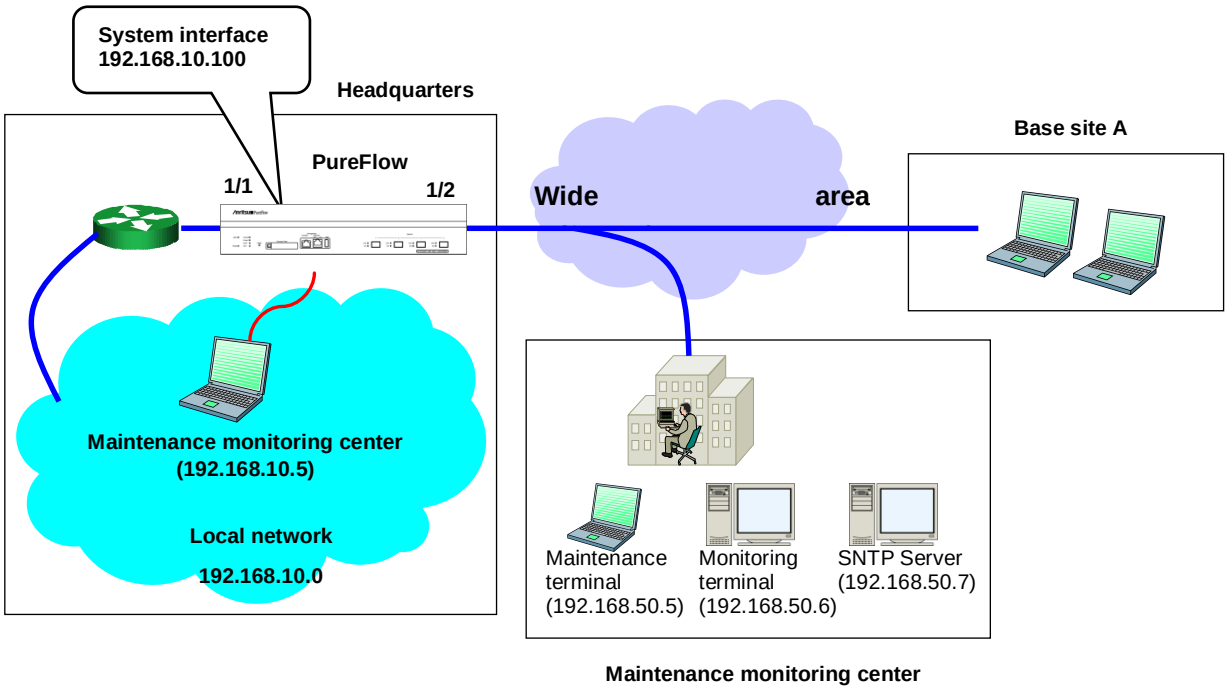
SNTP server setting:

PureFlow(A)> set sntp server 192.168.50.7

PureFlow(A)> set sntp enable

Case 5 Performing maintenance and monitoring from the Wide area Ethernet / IP-VPN network and local network via the Ethernet port

- The local network within the headquarters is 192.168.10.0/255.255.255.0.
- The network to Site A is 192.168.2.0/255.255.255.0.
- The network to the maintenance monitoring center is 192.168.50.0/255.255.255.0.
- 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 addresses of the maintenance terminal (CLI, download/upload) are 192.168.50.5 and 192.168.10.5.
- The IPv4 address of the monitoring terminal (SNMP, Syslog) is 192.168.50.6.
- The IPv4 address of the SNTP server is 192.168.50.7.



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 192.168.50.6 version v2c
                    community honsya_system_management trap
```

Syslog host setting:

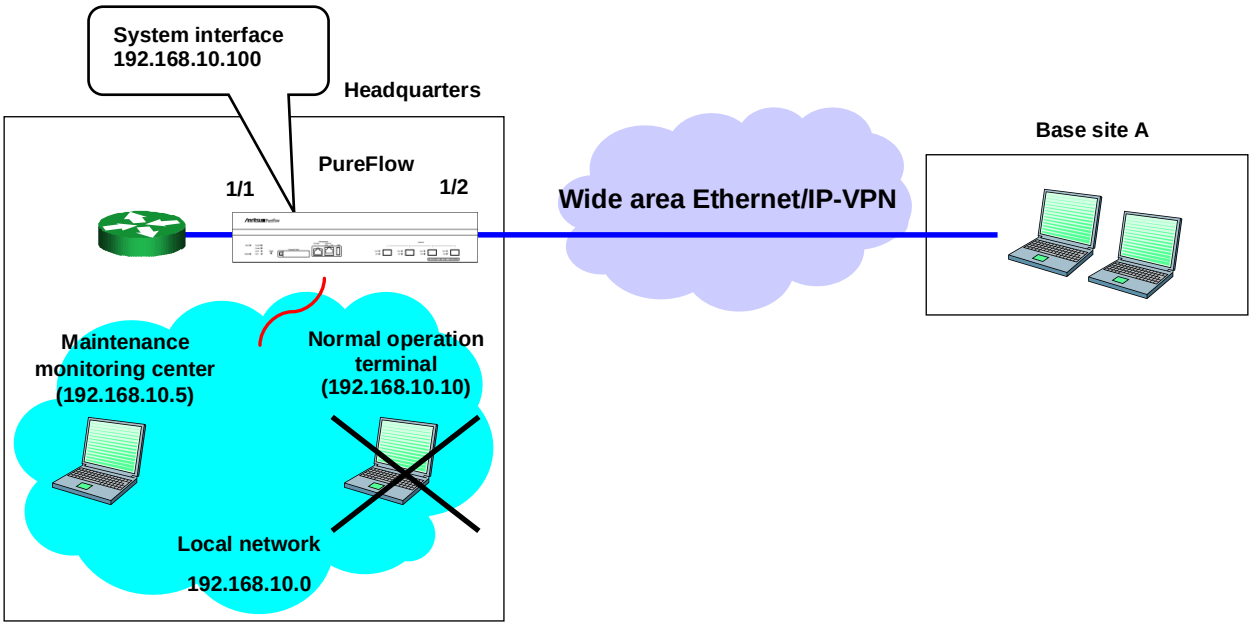
```
PureFlow(A)> set syslog host ip 192.168.50.6
PureFlow(A)> set syslog host enable
```

SNTP server setting:

```
PureFlow(A)> set sntp server 192.168.50.7
PureFlow(A)> set sntp enable
```

Case 6 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 sip 192.168.20 permit
```

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
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
system            N/A             N/A           N/A             N/A

Port        Err Packets    Collision      Discard
-------     -----------    ----------     ----------
 1/1                0               0             0
 1/2                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
```

(Blank page)

Chapter 8 Traffic Control

This chapter describes the traffic control feature and settings.

8.1 Overview 8-2
8.2 Hierarchical Shaping 8-3
8.3 Filters and Scenarios 8-4
8.3.1 Filters 8-5
8.3.2 Hierarchical relation of filters 8-7
8.3.3 Scenarios 8-8
8.3.4 Relationship between filters and scenarios 8-9
8.3.5 Rule list 8-11
8.4 Setting Procedure 8-12
8.5 How to Set a Rule List 8-22
8.6 Configuration Examples 8-25
8.7 Advanced Settings 8-30
8.7.1 Flow identification mode 8-31
8.7.2 Queues 8-35
8.7.3 Communication gap mode 8-42

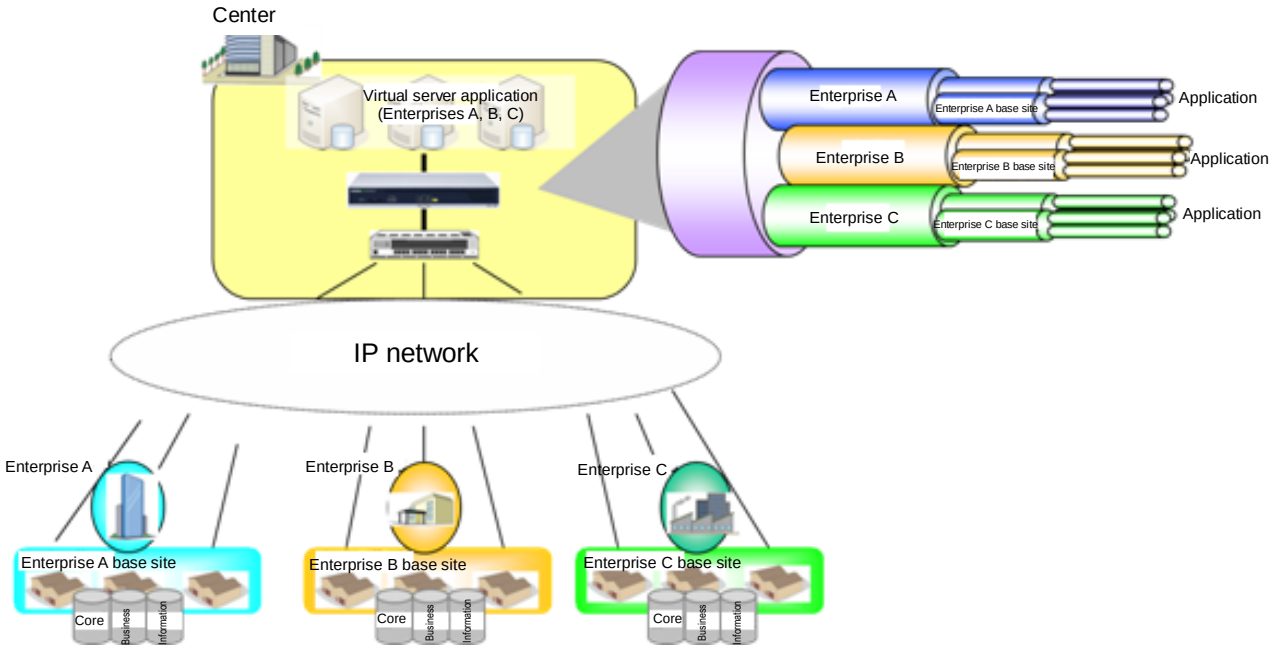
8.1 Overview

PureFlow WSX provides bandwidth control over a maximum 10 GBit/s broadband network. It is suitable for the core network of a large business that has many business bases, or a large-sized data center operated by a service provider.

Bandwidth control for a business core network generally uses hierarchical control for each business base to secure and allocate or restrict the bandwidth for each service or host. The bandwidth control for a service provider's data center also uses hierarchical control for each base or service host.

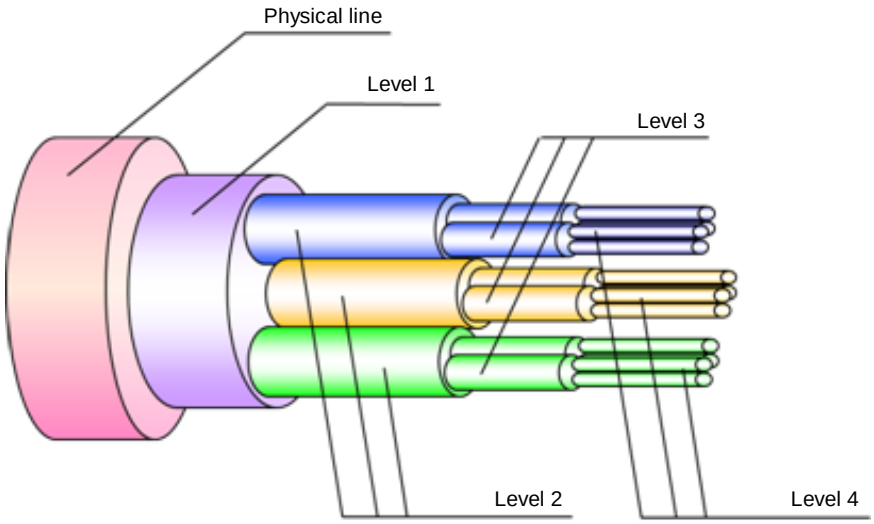
In cloud services, the servers, storage devices, and networks that make up the facility are virtualized and shared throughout the enterprise, and therefore securing an adequate bandwidth for the enterprise is needed before business base grouping. In this way, control can be performed in a hierarchy of up to 8 levels (the enterprise level, business base level, service host level, and so on). The performance, the number of hierarchy levels, queues, and grouping conditions are designed to be suitable for the cloud base.

For example, PureFlow WSX can easily set a virtual circuit with an adequate bandwidth for an enterprise, and set a virtual circuit with adequate bandwidths for business bases under the enterprise, and then secure or restrict bandwidths for services and hosts under the base.



8.2 Hierarchical Shaping

The hierarchical shaping is a layered traffic control method that divides the line bandwidth and allocates necessary bandwidths to groups such as enterprises, business bases, or users, and then allocates bandwidths to hosts and services.
This device can add a scenario and a filter to each level so that hierarchical shaping of up to 8 levels can be realized. On the first level (Level 1), the physical line bandwidth is controlled (shaped) at any bandwidth. On the second level (Level 2), the traffic of business bases and users is classified and set on the Level 2 circuit, and therefore the line bandwidth is divided into virtual circuits and separate bandwidths are allocated to them. On the third level (Level 3) and later, like the above, the bandwidth allocated to the upper level can be divided and controlled.
Following is a conceptual diagram of hierarchical shaping:



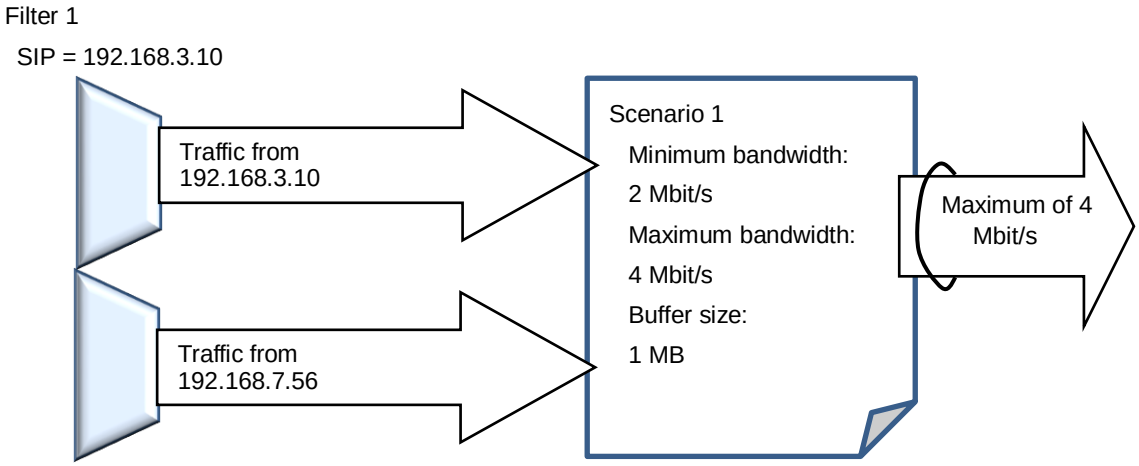
Level 1 (the first hierarchy)
The entire bandwidth of Level 1 can be controlled (shaped).
Level 1 can comprise one or more Level 2 bandwidths.
Level 2 (the second hierarchy)
The bandwidth of Level 1 is divided and controlled.
Level 2 can comprises one or more Level 3 bandwidths.
Level 3 (the third hierarchy)
The bandwidth of Level 2 is divided and controlled.
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.

8.3 Filters and Scenarios

This device uses a filter to classify packets and extract traffic. Classified traffic is controlled according to a scenario called a “traffic attribute” (minimum bandwidth, maximum bandwidth, buffer size).

Multiple filters can be associated with the same scenario, and a bandwidth shared by multiple users and applications can be allocated.



SIP: Source IP address

8.3.1 Filters

A filter defines a filter rule for classifying packets.
Use a Bridge-ctrl frame, Ethernet frame, IP packet or a combination of these to define a filter to extract traffic.

- Bridge-ctrl frame
  Specific destination MAC address packet for switch control (Bridge-ctrl filter)

- Ethernet frame
  VLAN ID, CoS (for duplex VLAN frames, filters can be set to both)
  The length/type field of the Ethernet header (Ethernet filter)

- IPv4/IPv6 packet (IP filter)
  VLAN ID, CoS (for duplex VLAN frames, filters can be set to both.)
  IPv4 IP address, protocol number, ToS, port number
  IPv6 IP address, protocol number, traffic class, port number

This device recognizes a VLAN Tag whose Tag Protocol ID is 0x8100 or 0x88A8. For IEEE standards, 0x88A8 is used for Outer Tag, and 0x8100 is used for Inner Tag, but for this device, both 0x8100 and 0x88A8 can be used for either tag.

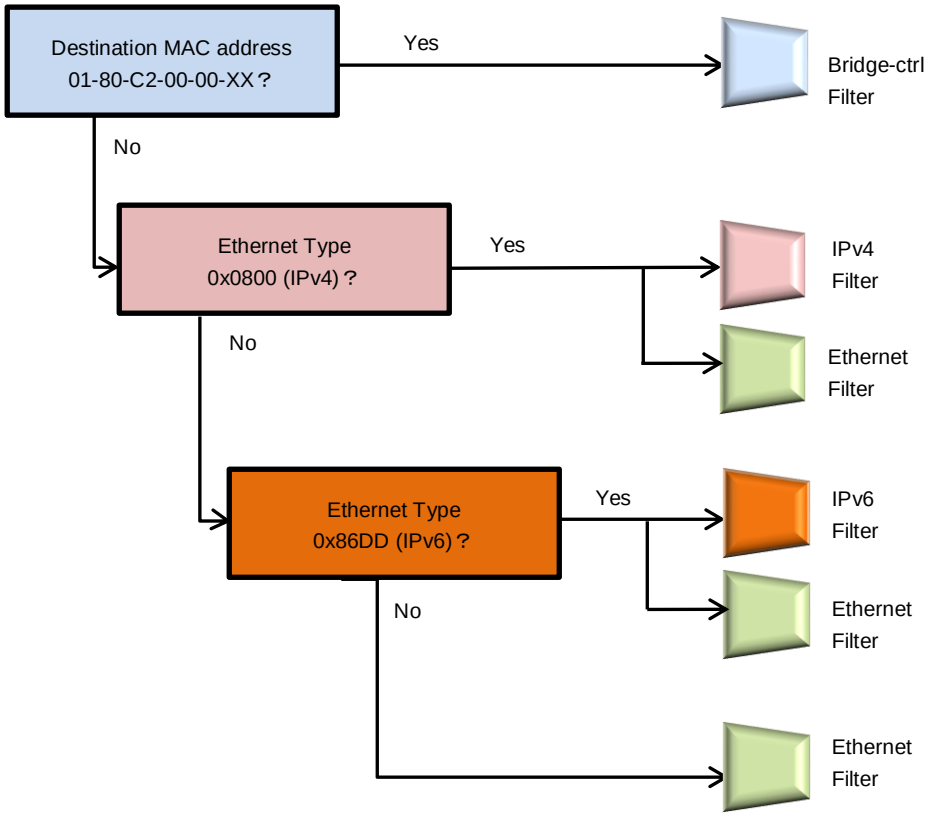
There are 3 types of filters: the Bridge-ctrl filter that classifies the Bridge-ctrl frames only, the Ethernet filter that classifies the length/type fields of the Ethernet header, and the IP filter that classifies the VLAN Tag fields, IP headers, TCP/UDP headers.

1. The Bridge-ctrl filter targets MAC addresses reserved for switch control such as the spanning tree protocol BPDU and link aggregation LACP.
   For example, use this to prioritize BPDU or secure the bandwidth in 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 whole Ethernet frame.
   Use this to classify packets by VLAN or packet type.
   For example, to enable bandwidth control per VLAN, specify VLAN alone.
   To prioritize ARP packets or secure a bandwidth, specify Ethernet Type “0806”.

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 (whether VLAN Tag is added is also identified)
   - CoS
   - Source IP address (SIP)
   - Destination IP address (DIP)
   - ToS or traffic class
   - Protocol number
   - Source port number (Sport)
   - Destination port number (Dport)

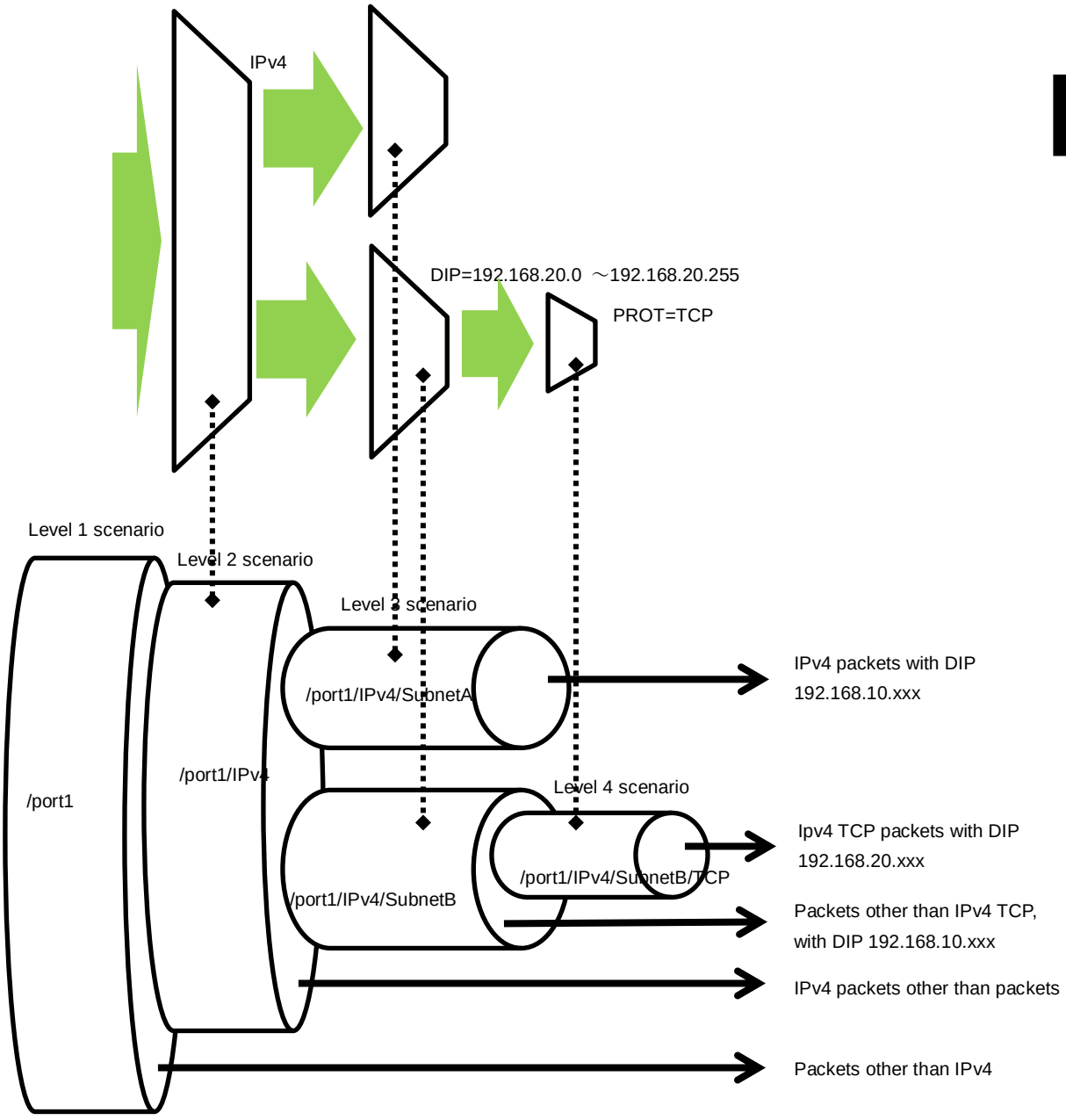
For traffic classification by filter, the applicable filter type is fixed according to the packet. For frames with the MAC address 01-80-C2-00-00-XX, only the Bridge-ctrl filter is applied regardless of other fields' contents. Provided the MAC address is not 01-80-C2-00-00-XX, only the IPv4 filter and 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.



8.3.2 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 SubnetA packets, SubnetB packets, and other Subnet IPv4 packets. For the filter of the Level 4 scenario, specify Protocol TCP to classify packets into SubnetB TCP packets and other packets.



8.3.3 Scenarios

A scenario is a setting (traffic attribute) for controlling traffic extracted by filters.

Two modes are available for scenarios: the Aggregate queue mode that controls the entire traffic, and the Individual queue mode that identifies each flow (the minimum identifiable unit of traffic in this device) to control traffic per flow. Traffic can be considered as a group consisting of multiple flows (see Section 8.7.1 for details of flows).

Traffic attributes need to be set for scenarios on Level 1 to 8, and the following parameters can be specified:

- class: Specifies the priority of the scenario. Priorities can be specified within the same level.
  The highest priority is given to class 1, the second highest priority is given to class 2, and so on.
  (For details about the class, see Section 8.7.2 Queues.)
- Guaranteed bandwidth: This is the guaranteed bandwidth allocated to a scenario. The bandwidth is secured even when traffic flows on the same level.
- Maximum bandwidth: This is the maximum bandwidth allocated to a scenario. All available bandwidths can be used.
- Buffer size: This is the allowable input burst length assigned to a scenario.
  If a lower level scenario exists under the scenario, for flows that match the filter criteria of the lower level scenario, the buffer size set for the lower level scenario is the allowable input burst length.
  If packets build up beyond the buffer size, the packets are discarded.
  (To specify an optimal buffer size, see Section 12.2.4 Determining the scenario parameters.)

For scenarios in the individual queue mode, the following parameters can be added:

- Maximum number of queues:
  Specifies the maximum number of queues generated for the scenario.
  A total of 300000 individual queues are available for all scenarios in the individual queue mode.
- Queue division target: Specifies the division target of the queues to be generated. Like the flow identification mode, this is specified as a packet field (for details about the flow identification mode, see Section 8.7.1 Flow identification mode).
  The specified field is identified, and an individual queue is assigned to a flow with a different field.
- Action when maximum queue count reached:
  Specifies the operation applied to a flow other than an IP flow if the number of queues generated exceeds the maximum queue count for the scenario or 300000 for all individual scenarios, or 5tuple (sip, dip, tos, proto, sport, dport) are included in the queue division target.
  discard Performs discard.
  forwardbesteffort Performs best effort transfer (class 8).

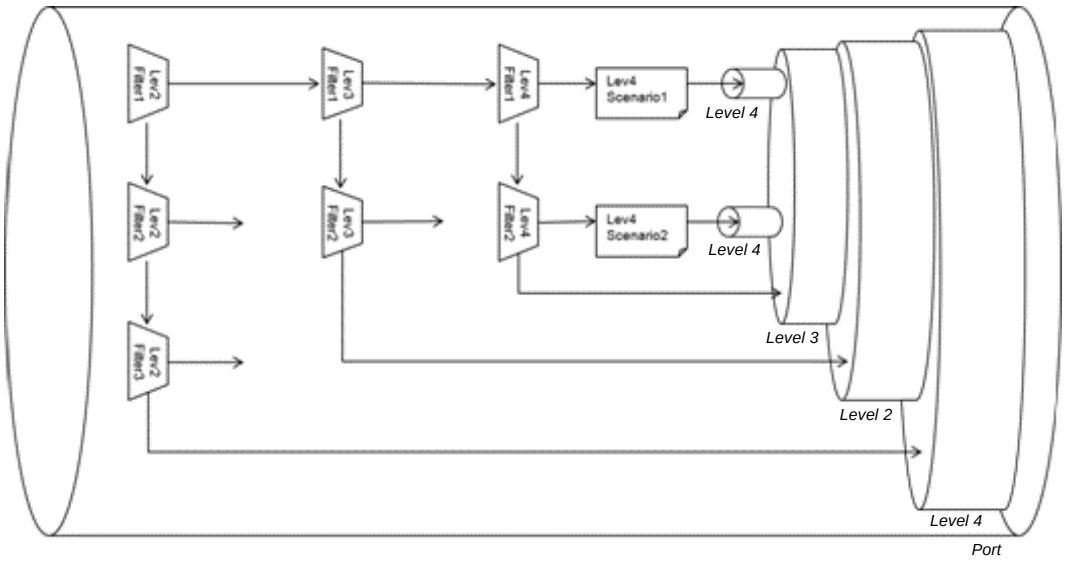
forwardattribute Transfers a specified traffic attribute.

- Minimum bandwidth when the maximum number of queues is exceeded
- Maximum bandwidth when the maximum number of queues is exceeded
- Class when the maximum number of queues is exceeded:

This is the traffic attribute when forwardattribute is specified for the action when the maximum number of queues is exceeded.

8.3.4 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 and buffer size.



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

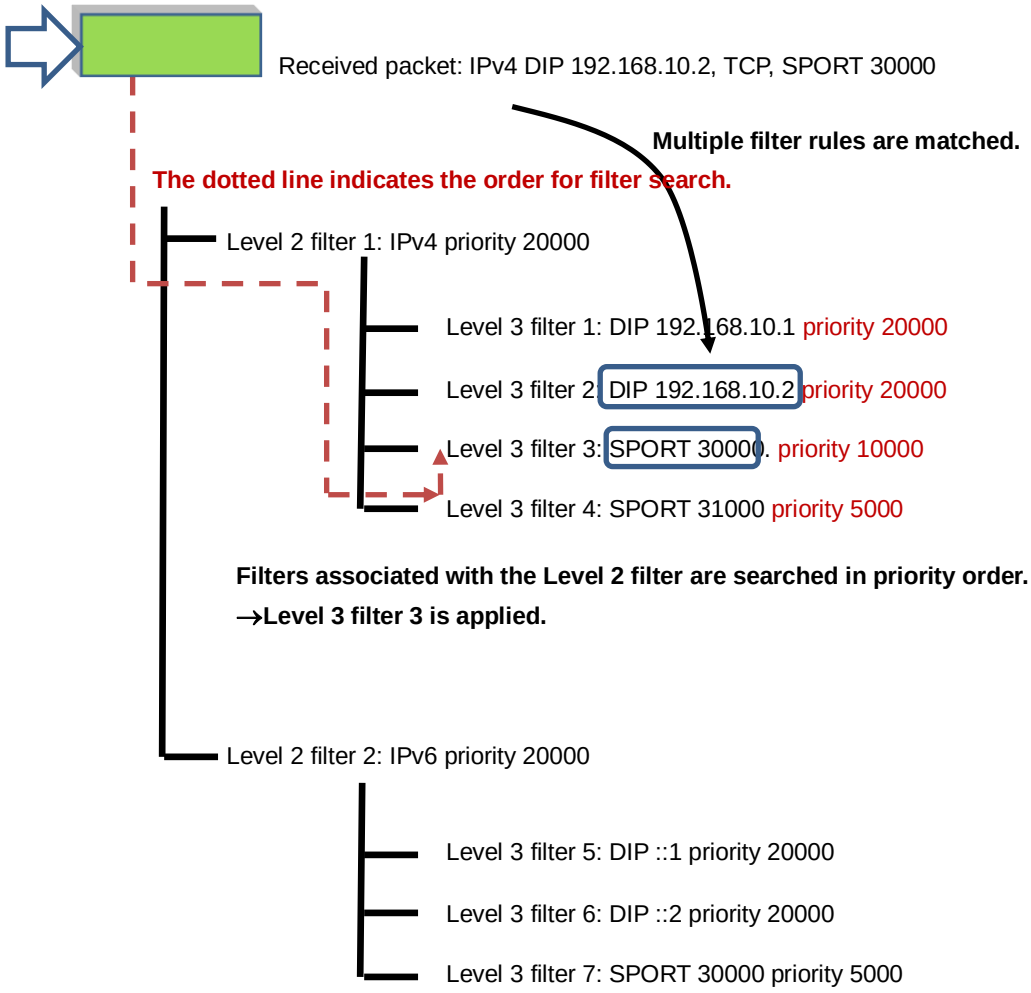
Bandwidth control from Level 1 to Level 4, and discard and transfer control by the filter setting are available.
For the filter action, “aggregate”, “individual”, and “discard” can be specified. Packets that match the filter rules follow the configured operation.
This device uses Level 2 filters in a priority order to verify whether received packets match the filer 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 “individual” scenario, and filters 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.

8.3.5 Rule list

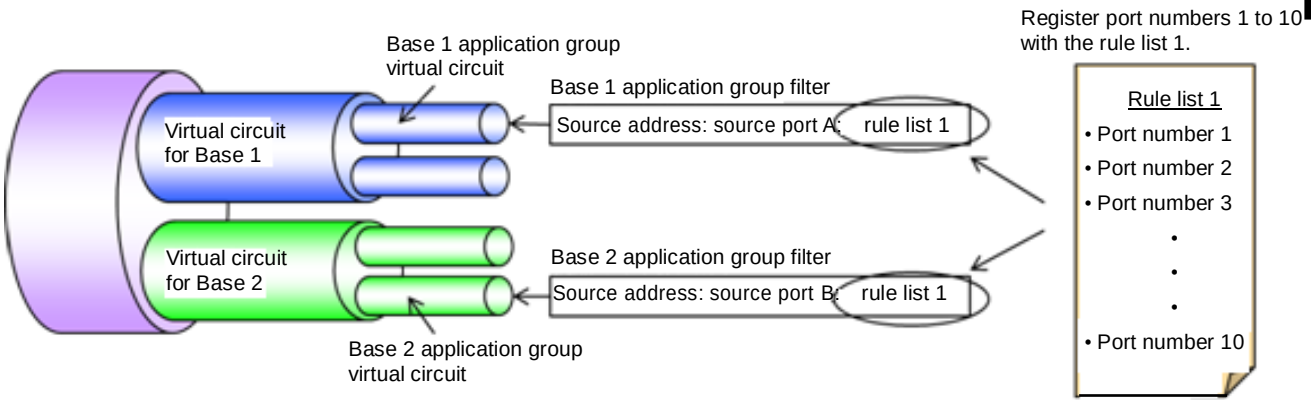
The rule list is a feature to group multiple classification conditions for traffic (IP address, port number, etc). By using this feature, multiple classification conditions for traffic can be specified by single rule list name.

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

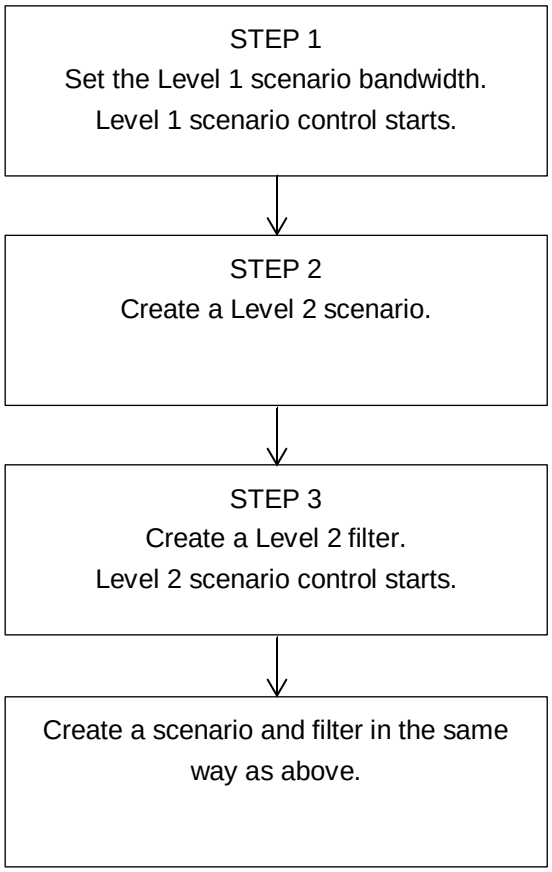
Rule lists can be specified for multiple filters repeatedly. Using rule lists reduces the number of filters and lines for configuration.



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.

8.4 Setting Procedure

The figure below describes the setting procedure.



Details of each step are described below.

STEP 1: Set the Level 1 scenario bandwidth

This device assigns a traffic attribute for each virtual circuit according to the scenario setting. The following parameters can be specified for the Level 1 scenario.

Parameter	Setting range	Optional/required	Description
Scenario name (scenario_name)	"/port1" or "/port2"	Required	Cannot be omitted or changed.
Minimum bandwidth (min_bandwidth)	0,1 k[bit/s] to 10 G[bit/s]	Optional	When omitted: Minimum bandwidth is not guaranteed * Can be set but does not affect the operation.
Maximum bandwidth (peak_bandwidth)	2 k[bit/s] to 10 G[bit/s]	Optional	When omitted: 1 Gbit/s
Buffer size (bufsize)	2 k[Byte] to 100 M[Byte]	Optional	When omitted: 10 M[Byte]

The following CLI command is for a Level 1 scenario:

update scenario <scenario_name> action aggregate [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [class <class>] [bufsize <bufsize>]	This command can modify a traffic attribute of the traffic transmitted from the Network port. For a Level 1 scenario, specify "/port1" or "/port2" for <scenario_name>. Classes of a Level 1 scenario cannot be changed.

The Level 1 scenario ("/port1" or "/port2") is set by default, and cannot be added or changed. Use the "update scenario" command to change parameters for the Level 1 scenario ("/port1" or "/port2").

An example of a Level 1 scenario is as follows:

Sample 1) Set the maximum bandwidth to 5 Gbit/s for traffic from port1 to port2.

```
PureFlow(A)> update scenario "/port1" action aggregate peak_bw 5G
```

Sample 2) Set the maximum bandwidth to 3 Gbit/s for traffic from port2 to port1.

```
PureFlow(A)> update scenario "/port2" action aggregate peak_bw 3G
```

STEP 2: Setting the bandwidth for Level 2 and lower scenarios

This device assigns a traffic attribute for each virtual circuit according to the scenario setting. 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) "/port1/xxxx/xxxx" (Level 3) "/port2/xxxx/xxxx" (Level 3) And so forth (to Level 8)	Required	Cannot be omitted or changed. For the first level, specify port number such as "/port1" or "/port2" for the Network port number, and then specify a scenario name to be added for the second level or lower. Valid values are from 1 to 128 characters for all levels (/port1, /port2).
Action mode	discard/aggregate/individual	Required	
Class (class)	1 to 8	Optional	When omitted: 2 1 (high) ⇔ (low) 8
Minimum bandwidth (min_bandwidth)	0 or 1 k[bit/s] to 10 G[bit/s]	Optional	When omitted: Minimum bandwidth is not guaranteed
Maximum bandwidth (peak_bandwidth)	2 k[bit/s] to 10 G[bit/s]	Optional	When omitted: Maximum bandwidth is not guaranteed.
Buffer size (bufsize)	2 k[Byte] to 100 M[Byte]	Optional	When omitted: 1 M[Byte]
Scenario index	1 to 40000	Optional	When omitted: Auto assignment in the device
Maximum number of queues (maxquenum)	1 to 300000	Optional	When omitted: 300000 Enabled only for the individual queue mode.

Parameter	Setting range	Optional/required	Description
Queue division target (quedivision)	default: A combination of 5 tuple (sip, dip, tos, proto, sport, dport) is used to divide queues. vlan: Divides queues based on VLAN ID. cos: Divides queues based on CoS. inner-vlan: Divides queues based on inner VLAN ID. inner-cos: Divides queues based on inner CoS. ethertype: Divides queues based on Ethernet Type/Length. sip: Divides queues based on the source IP address. dip: Divides queues based on the destination IP address. tos: Divides queues based on ToS or Traffic Class. proto: Divides queues based on the protocol number. sport: Divides queues based on the source port number. dport: Divides queues based on the destination port number.	Optional	When omitted: default Enabled only for the individual queue mode.
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 Enabled only for the individual queue mode.
Minimum bandwidth when the maximum number of queues is exceeded (fail_min_bw)	0 or 1 k[bit/s] to 10 G[bit/s]	Optional	When omitted: Minimum bandwidth is not guaranteed Available only when “forwardattribute” is specified in the individual queue mode

Parameter	Setting range	Optional/required	Description
Maximum bandwidth when the maximum number of queues is exceeded (fail_peak_bw)	2 k[bit/s] to 10 G[bit/s]	Optional	When omitted: Maximum bandwidth is not guaranteed. Available only when “forwardattribute” is specified in the individual queue 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

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

add scenario <scenario_name> action discard [scenario <scenario_id>]	Registers a discard mode scenario. A scenario index is automatically assigned, and normally need not be set.
add scenario <scenario_name> action aggregate [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [class <class>] [bufsize <bufsize>] [scenario <scenario_id>]	Registers a scenario in the aggregate queue mode. Traffic attributes such as the bandwidth and buffer size are set to control traffic. A scenario index is automatically assigned, and normally need not be set.
add scenario <scenario_name> action individual [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>]	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 set. A scenario index is automatically assigned, and normally need not be set.
update scenario <scenario_name> action aggregate [min_bw <min_bandwidth>] [peak_bw <peak_bandwidth>] [class <class>] [bufsize <bufsize>]	Changes a scenario in the aggregate queue mode. 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. The scenario name, action mode, and scenario index cannot be changed.
update scenario <scenario_name> action individual [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>]	Changes a scenario in the individual queue mode. 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. The scenario name, action mode, and scenario index cannot be changed.

An example of Level 2 scenario is shown below.

Sample 1) Set the maximum bandwidth to 3 Gbit/s for the aggregate queue mode scenario of the "Tokyo" base received from port1.

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

Sample 2) Change the maximum bandwidth to 1 Gbit/s for the aggregate queue mode scenario of the "Osaka" base received from port1.

```
PureFlow(A)> update scenario "/port1/Osaka" action aggregate peak_bw 1G
```

Sample 3) Set the maximum bandwidth to 500 Kbits/s and the maximum number of queues to 20 for the individual queue mode scenario of the "Nagoya" base received from port1.

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

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

Sample 4) Register the "Shinjuku" area under the "Tokyo" base as an aggregate queue mode scenario, and set the maximum bandwidth to 100 Mbit/s.

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

STEP 3: Setting the filter for Level 2 and lower scenarios

This device uses filters to identify the Bridge-ctrl frame, Ethernet frame, IPv4 packet, and IPv6 packet traffic.

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

Parameter		Setting range	Optional/required
Filter name		1 to 48 characters	Required
Scenario name		Total of 1 to 128 characters for all levels (set by the “add scenario” command)	Required
Filter type		bridge-ctrl, ethernet, ipv4, ipv6	Required
Ethertype		Specifies the Type field in the Ethernet header. 0x0000 to 0xFFFF	Optional Enabled only for Ethernet filters
VLAN ID		Specifies IEEE802.1Q VLAN ID. 0 to 4094 (range specification available), none (without VLAN tag)	Optional
Inner VLAN ID		Specifies the inner VLAN ID for QinQ. 0 to 4094 (range specification available), none (without VLAN tag)	Optional
CoS		Specifies CoS in IEEE802.1Q VLAN ID. 0 to 7	Optional
Inner CoS		Specifies the CoS in inner VLAN for QinQ. 0 to 7	Optional
Source IP address	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	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
ToS, or Traffic Class	IPv4	0 to 255 (The range “start-end” can be specified.)	Optional Enabled only for IP filters
	IPv6	0 to 255 (The range “start-end” can be specified.)	Optional Enabled only for IP filters

Parameter	Setting range	Optional/required
Protocol number	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	0 to 65535 (The range "start-end" can be specified.) Rule list name	Optional Enabled only for IP filters
Destination port number	0 to 65535 (The range "start-end" can be specified.) Rule list name	Optional Enabled only for IP filters
Filter priority	1 to 40000	Optional When omitted: 20000

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

add filter scenario <scenario_name> filter <filter_name> bridge-ctrl [priority <filter_pri>]	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)).
add filter scenario <scenario_name> filter <filter_name> ethernet [vid {<VID> | none}] [cos <user_priority>] [inner-vid {<VID> | none}] [inner-cos <user_priority>] [ethertype <type>] [priority <filter_pri>]	Identifies frames based on the length/type field of the Ethernet header. This can also be specified for VLAN ID and CoS in the VLAN tag. Each of the parameters can be omitted but you cannot omit all the parameters. Specify at least one parameter other than "priority".
add filter scenario <scenario_name> filter <filter_name> ipv4 [vid {<VID> | none}] [cos <user_priority>] [inner-vid {<VID> | none}] [inner-cos <user_priority>] [sip [list] {<src_IP_address> | <list_name>}] [dip [list] {<dst_IP_address> | <list_name>}] [tos <type_of_service>] [proto <protocol>] [sport [list] {<sport> | <list_name>}] [dport [list] {<dport> | <list_name>}] [priority <filter_pri>]	Identifies IPv4 packets based on the IP address, protocol number, port number, etc. This can also be specified for VLAN ID and CoS in the VLAN tag. Each parameter can be omitted. If all parameters are omitted, all IPv4 packets are targeted.

add filter scenario <scenario_name> filter <filter_name> ipv6 [vid {<VID> | none}] [cos <user_priority>] [inner-vid {<VID> | none}] [inner-cos <user_priority>] [sip [list] {<src_IP_address> | <list_name>}] [dip [list] {<dst_IP_address> | <list_name>}] [tos <type_of_service>] [proto <protocol>] [sport [list] {<sport> | <list_name>}] [dport [list] {<dport> | <list_name>}] [priority <filter_pri>]	Identifies IPv6 packets based on the IP address, protocol number, port number, etc. This can also be specified from VLAN ID and CoS in the VLAN tag. Each parameter can be omitted. If all parameters are omitted, all IPv6 packets are targeted.

An example of Level 2 filters is shown below.

Sample 1) Set BPDU 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) Set ARP as a filter for the Level 2 scenario "/port1/arp".

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

Sample 3) Set VLAN ID for IPv4 to "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) Set VLAN ID for IPv6 to "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 4) Set the source IP address for IPv4 within 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
```

8.5 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: Set the rule list.
Step 2: Add 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 as follows:

add rulelist group <list_name> {ipv4 | ipv6 | l4port}	Adds a rule list. Either ipv4 or ipv6 or l4port is targeted.
add rulelist entry <list_name> ipv4 <IP_address>	Adds the IPv4 address to the rule list.
add rulelist entry <list_name> ipv6 <IP_address>	Adds the IPv6 address to the rule list.
add rulelist entry <list_name> l4port <port>	Adds the l4 port number to the rule list.
delete rulelist group {<list_name> | all}	Deletes a rule list.
delete rulelist entry <list_name> ipv4 <IP_address>	Deletes the IPv4 address from the rule list.
delete rulelist entry <list_name> ipv6 <IP_address>	Deletes the IPv6 address from the rule list.
delete rulelist entry <list_name> l4port <port>	Deletes the l4 port number from the rule list.
show rulelist [<list_name>]	Displays the 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 procedure for setting a rule list is shown below.

Step 1) Register the rule list “TVCservers”.

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

Step 2) Add 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) Specify the rule list name “TVCservers” for “sip” of the add filter command.

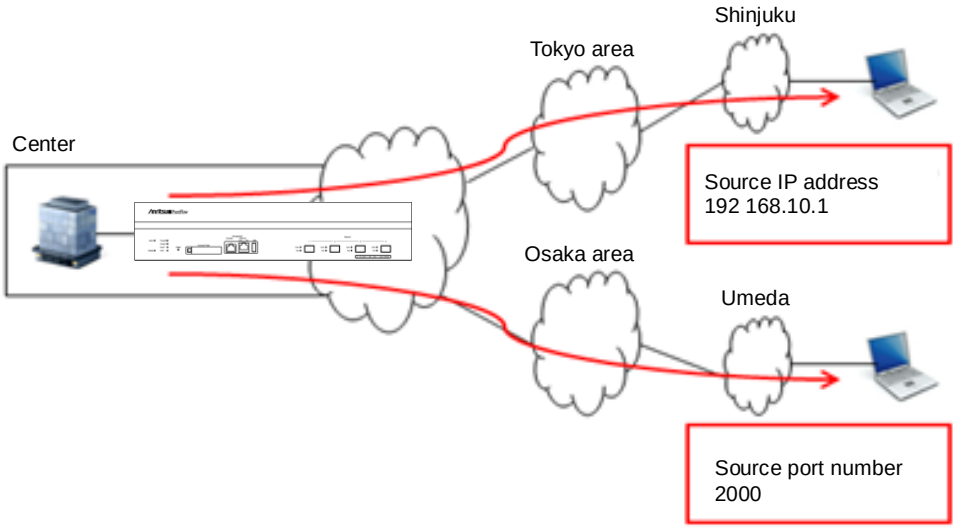
```
PureFlow(A)> add filter scenario “/port1/Tokyo/TVC” filter “TVC” ipv4 sip list
                    “TVCservers”
```

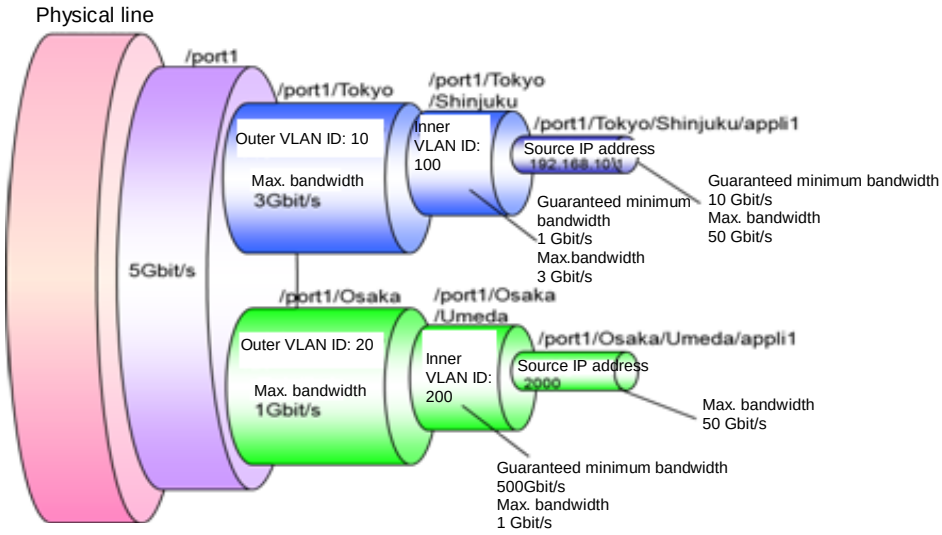
8.6 Configuration Examples

This section shows configuration examples of setting the following network environments.
In this example, an area is assigned to Level 2, a network to Level 3, and an application to Level 4.

Case 1 Securing the bandwidths for the area and application

- The network for Tokyo is Outer VLAN ID 10. (Level 2 filter setting)
- The network for Osaka is Outer VLAN ID 20. (Level 2 filter setting)
- The network for Shinjuku in Tokyo is Inner VLAN ID 100. (Level 3 filter setting)
- The network for Umeda in Osaka is Inner VLAN ID 200. (Level 3 filter setting)
- The source IP address of the controlled application in Shinjuku is “192.168.10.1”. (Level 4 filter setting)
- The source port number of the controlled application in Umeda is “2000”. (Level 4 filter setting)
- The maximum bandwidth from WSX is set to 5 Gbit/s. (Level 1 circuit setting)
- The maximum bandwidth from the center to Tokyo is set to 3 Gbit/s, and the maximum bandwidth to Osaka is set to 1 Gbit/s. (Level 2 scenario setting)
- In the 3 Gbit/s guaranteed bandwidth for Tokyo area, the minimum guaranteed bandwidth is set to 1 Gbit/s and the maximum bandwidth to 3 Gbit/s for Shinjuku area. (Level 3 scenario setting)
- In the 1 Gbit/s guaranteed bandwidth for Osaka area, the minimum guaranteed bandwidth is set to 500 Mbit/s and the maximum bandwidth to 1 Gbit/s for Umeda area. (Level 3 scenario setting)
- For the controlled application in Shinjuku area, the minimum guaranteed bandwidth is set to 10 Mbit/s, and the maximum bandwidth to 50 Mbits/s. (Level 4 scenario setting)
- For the controlled application in Umeda area, the maximum bandwidth is set to 50 Mbit/s. (Level 4 scenario setting)





Execute the following commands.

Level 1 scenario setting:

PureFlow(A)> update scenario "/port1" action aggregate peak_bw 5G

Level 2 scenario setting:

PureFlow(A)> add scenario "/port1/Tokyo" action aggregate peak_bw 3G
PureFlow(A)> add scenario "/port1/Osaka" action aggregate peak_bw 1G

Level 2 filter setting:

PureFlow(A)> add filter scenario "/port1/Tokyo" filter "Tokyo" ipv4 vid 10
PureFlow(A)> add filter scenario "/port1/Osaka" filter "Osaka" ipv4 vid 20

Level 3 scenario setting:

PureFlow(A)> add scenario "/port1/Tokyo/shinjuku" action aggregate min_bw 1G peak_bw 3G
PureFlow(A)> add scenario "/port1/Osaka/Umeda" action aggregate min_bw 500M peak_bw 1G

Level 3 filter setting:

PureFlow(A)> add filter scenario "/port1/Tokyo/Shinjuku" filter "Shinjuku" ipv4 inner-vid 100
PureFlow(A)> add filter scenario "/port1/Osaka/Umeda" filter "Umeda" ipv4 inner-vid 200

Level 4 scenario setting:

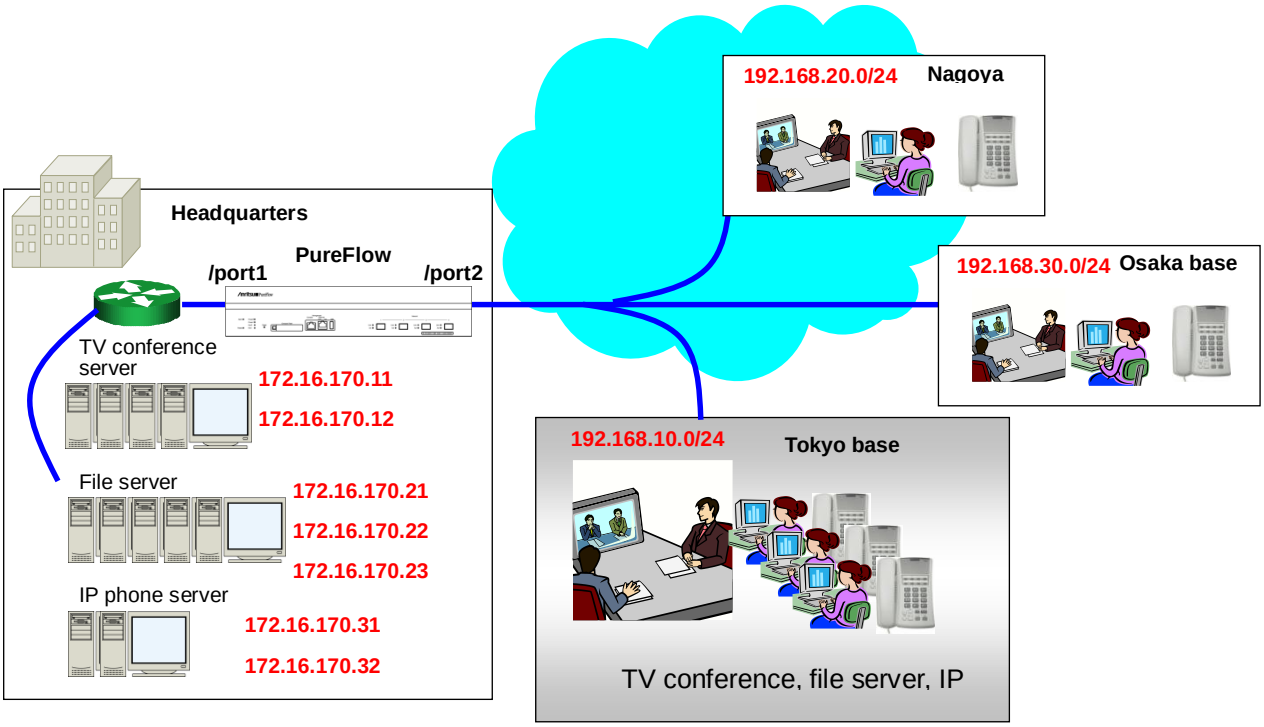
PureFlow(A)> add scenario "/port1/Tokyo/Shinjuku/appli1" action aggregate min_bw 10M peak_bw 50M
PureFlow(A)> add scenario "/port1/Osaka/Umeda/appli1" action aggregate peak_bw 50M

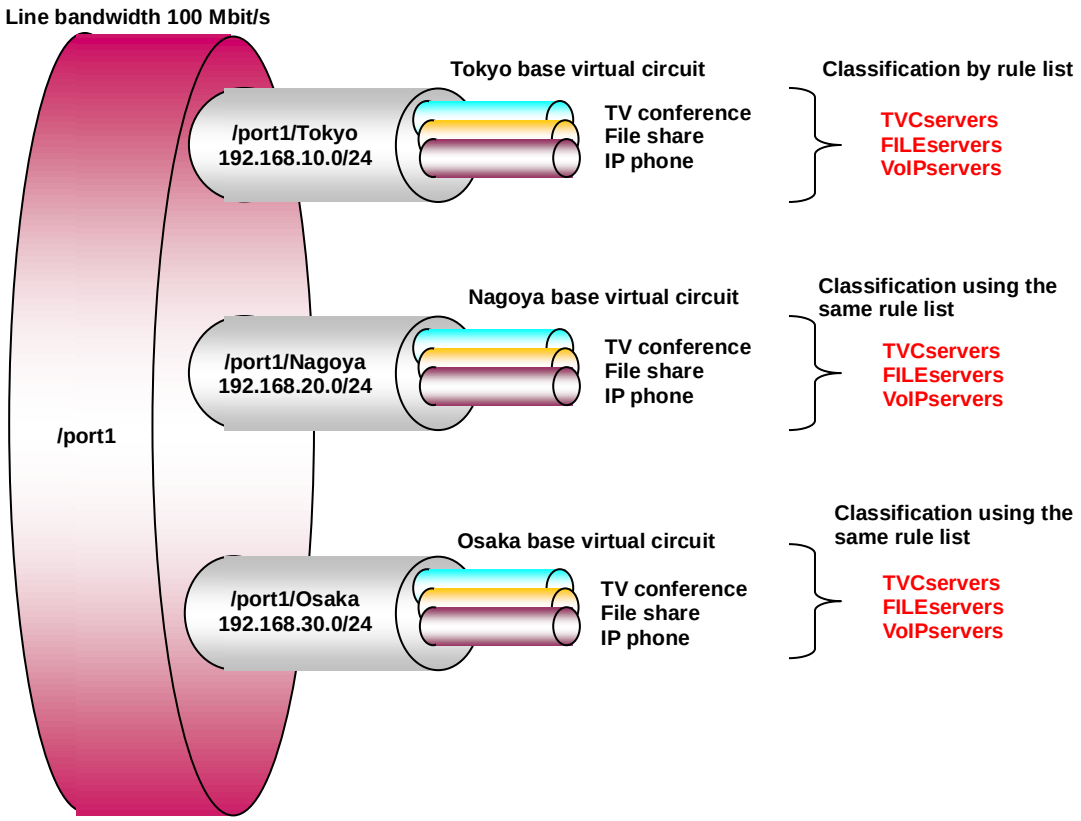
Level 4 filter setting:

PureFlow(A)> add filter scenario "/port1/Tokyo/Shinjuku/appli1" filter "Shin_appli1" ipv4 sip 192.168.10.1
PureFlow(A)> add filter scenario "/port1/Osaka/Umeda/appli1" filter "Ume_appli1" ipv4 sport 2000

Case 2 Using a rule list to simplify the filter setting

- Each base uses the server in the headquarters. (TV conference, file server, VoIP)
- PureFlowWSX allocates communication bandwidths to each base, and then allocates bandwidths to each service.





Register services to each rule list.

- Register the IP addresses of TV conference servers to the rule list.

  PureFlow(A)> add rulelist group “TVCservers” ipv4
  PureFlow(A)> add rulelist entry “TVCservers” ipv4 172.16.170.11
  PureFlow(A)> add rulelist entry “TVCservers” ipv4 172.16.170.12

- Register the IP addresses of file servers to the rule list.

  PureFlow(A)> add rulelist group “FILEservers” ipv4
  PureFlow(A)> add rulelist entry “FILEservers” ipv4 172.16.170.21
  PureFlow(A)> add rulelist entry “FILEservers” ipv4 172.16.170.22
  PureFlow(A)> add rulelist entry “FILEservers” ipv4 172.16.170.23

- Register the IP addresses of IP phone server to the rule list.

  PureFlow(A)> add rulelist group “VoIPservers” ipv4
  PureFlow(A)> add rulelist entry “VoIPservers” ipv4 172.16.170.31
  PureFlow(A)> add rulelist entry “VoIPservers” ipv4 172.16.170.32

Register a virtual circuit for Tokyo base.

- Set the total amount of traffic to Tokyo base.

  PureFlow(A)> add scenario “/port1/Tokyo” action aggregate peak_bw 10M
  PureFlow(A)> add filter scenario “/port1/Tokyo” filter “Tokyo” ipv4 dip
  192.168.10.0-192.168.10.255

- Register traffic by using the rule list of TV conference servers.

  PureFlow(A)> add scenario “/port1/Tokyo/TVC” action aggregate min_bw 5M
  PureFlow(A)> add filter scenario “/port1/Tokyo/TVC” filter “Tokyo_TVC” ipv4
  sip list “TVCservers”

- Register traffic by using the rule list of file servers.

  PureFlow(A)> add scenario “/port1/Tokyo/FILE” action aggregate min_bw 4M
  PureFlow(A)> add filter scenario “/port1/Tokyo/FILE” filter “Tokyo_FILE” ipv4
  sip list “FILEservers”

- Register traffic by using the rule list of IP phone servers.

  PureFlow(A)> add scenario “/port1/Tokyo/VoIP” action aggregate min_bw 1M
  PureFlow(A)> add filter scenario “/port1/Tokyo/VoIP” filter “Tokyo_VoIP” ipv4
  sip list “VoIPservers”

Use the same rule lists to register traffic for Nagoya and Osaka bases.

Register a virtual circuit for Nagoya base.

- Set the total amount of traffic to Nagoya base.

```
PureFlow(A)> add scenario "/port1/Nagoya" action aggregate peak_bw 10M
PureFlow(A)> add filter scenario "/port1/Nagoya" filter "Nagoya" ipv4 dip
                192.168.20.0-192.168.20.255
```

- Register traffic by using the rule list of TV conference servers.

```
PureFlow(A)> add scenario "/port1/Nagoya/TVC" action aggregate min_bw 5M
PureFlow(A)> add filter scenario "/port1/Nagoya/TVC" filter "Nagoya_TVC" ipv4
                sip list "TVCservers"
```

- Register traffic by using the rule list of file servers.

```
PureFlow(A)> add scenario "/port1/Nagoya/FILE" action aggregate min_bw 4M
PureFlow(A)> add filter scenario "/port1/Nagoya/FILE" filter "Nagoya_FILE" ipv4
                sip list "FILEservers"
```

- Register traffic by using the rule list of IP phone servers.

```
PureFlow(A)> add scenario "/port1/Nagoya/VoIP" action aggregate min_bw 1M
PureFlow(A)> add filter scenario "/port1/Nagoya/VoIP" filter "Nagoya_VoIP" ipv4
                sip list "VoIPservers"
```

Register a virtual circuit for Osaka base.

- Set the total amount of traffic to Osaka base.

```
PureFlow(A)> add scenario "/port1/Osaka" action aggregate peak_bw 10M
PureFlow(A)> add filter scenario "/port1/Osaka" filter "Osaka" ipv4 dip
                192.168.30.0-192.168.30.255
```

- Register traffic by using the rule list of TV conference servers.

```
PureFlow(A)> add scenario "/port1/Osaka/TVC" action aggregate min_bw 5M
PureFlow(A)> add filter scenario "/port1/Osaka/TVC" filter "Osaka_TVC" ipv4
                sip list "TVCservers"
```

- Register traffic by using the rule list of file servers.

```
PureFlow(A)> add scenario "/port1/Osaka/FILE" action aggregate min_bw 4M
PureFlow(A)> add filter scenario "/port1/Osaka/FILE" filter "Osaka_FILE" ipv4
                sip list "FILEservers"
```

- Register traffic by using the rule list of IP phone servers.

```
PureFlow(A)> add scenario "/port1/Osaka/VoIP" action aggregate min_bw 1M
PureFlow(A)> add filter scenario "/port1/Osaka/VoIP" filter "Osaka_VoIP" ipv4
                sip list "VoIPservers"
```

8.7 Advanced Settings

This device provides the following advanced settings:

- Flow identification mode
- Queue
- Communication gap mode

8.7.1 Flow identification mode

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 whose 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 VLAN Tag is added is also identified)
- CoS
- Inner VLAN ID
- Inner CoS
- Ethernet Type.

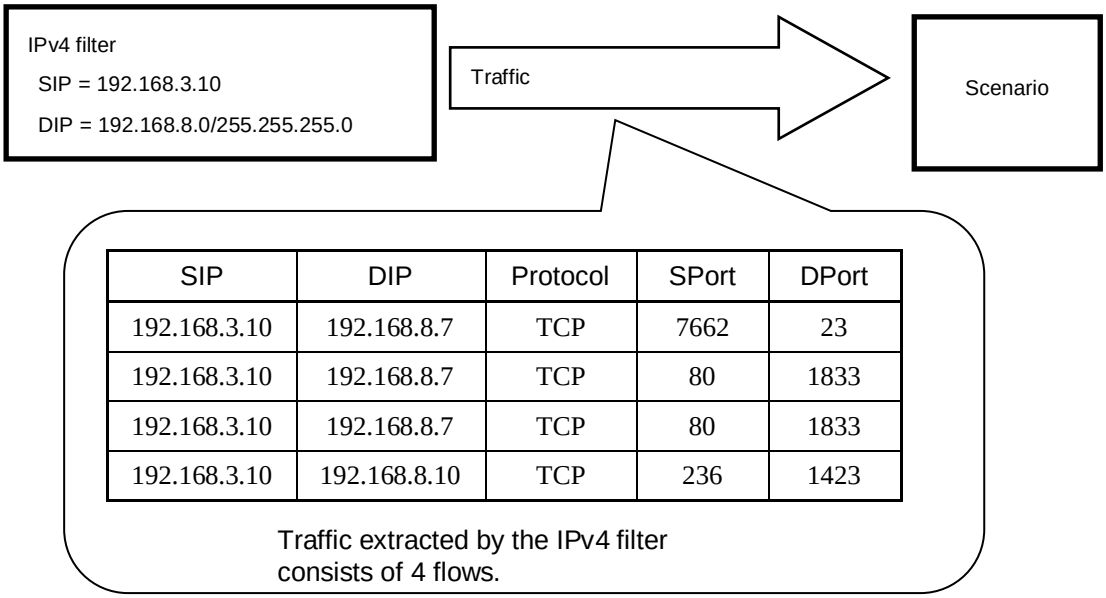
(3) IPv4/IPv6 flow

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

- VLAN ID (whether VLAN Tag is added is also identified)
- CoS
- Inner VLAN ID
- Inner CoS
- Source IP address (SIP)
- Destination IP address (DIP)
- ToS or traffic class
- Protocol number
- Source port number (Sport)
- Destination port number (Dport)

In the flow identification mode, fields for identifying EthernetType flows and IPv4/IPv6 flows can be selected.

For example, normally, IPv4 flows are traffic that matches the source IP address (SIP), destination IP address (DIP), protocol number (Protocol), source port number (SPort), and destination port number (DPort).



This device can change the combination of these fields for flow identification (flow identification mode). You can therefore transfer packets that have different fields as different flows or as the same flow.

The following parameters can be specified for the flow identification mode:

Parameter	Setting range	Optional/required
Input Network port	1/1 / 1/2	Required
Field name	default: Sets the flow identification fields to the default values. The source IP address, destination IP address, protocol number, source port number, and destination port number are used for flow identification. vid: Identifies flows based on VLAN ID. cos: Identifies flows based on CoS. inner-vid: Identifies flows based on inner VLAN ID. inner-cos: Identifies flows based on inner CoS. sip: Identifies flows based on the source IP address. dip: Identifies flows based on the destination IP address. tos: Identifies flows based on ToS or Traffic Class. proto: Identifies flows based on the protocol number. sport: Identifies flows based on the source port number. dport: Identifies flows based on the destination port number.	Required

Multiple parameters can be specified by delimiting them with commas (,).

The following CLI command is used for flow identification:

set filter mode in <slot/port> <field>	Selects the identification field of the flow. The default value of “field” is “default”.

The following is a command execution example:

PureFlow(A)> set filter mode in 1/1 cos
PureFlow(A)> set filter mode in 1/2 sip,dip
PureFlow(A)>

Note:

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. Parameters not used for packet classification by filter rules can be excluded from the flow identification mode to reduce internal resource consumption.

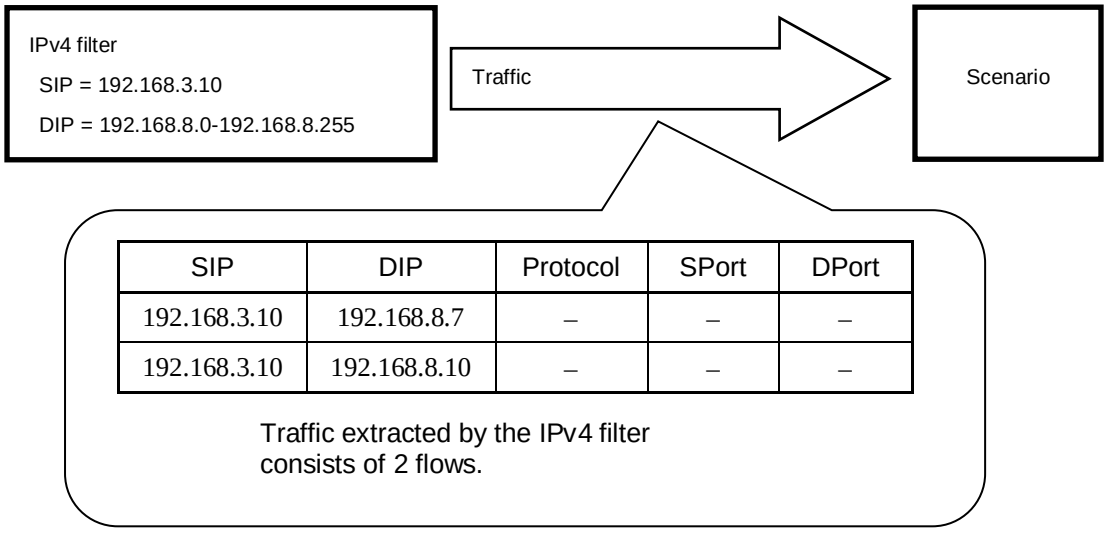
Note:

BridgeControl flows are created on a one-per-port basis regardless of the flow identification mode setting.

Note:

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 it's flow is not identified.

For example, to identify flows based on the source and destination IP addresses to control IPv4 packets with different fields as the same IPv4 flow, enable sip and dip. In this flow identification mode, the source and destination IP addresses are targeted as the IPv4 filter criteria registered by the "add filter" command. IPv4 filters that contain fields other than those specified in the flow identification mode are considered invalid.



The following table shows the relationships between the specified field names and fields identified for flows.

Specified field name	Flow identification field									
	VLAN ID	CoS	Inner VLAN ID	Inner CoS	SIP	DIP	ToS	Protocol number	Sport number	Dport number
default	–	–	–	–	✓	✓	–	✓	✓	✓
vid	✓	–	–	–	–	–	–	–	–	–
cos	–	✓	–	–	–	–	–	–	–	–
inner-vid	–	–	✓	–	–	–	–	–	–	–
inner-cos	–	–	–	✓	–	–	–	–	–	–
sip	–	–	–	–	✓	–	–	–	–	–
dip	–	–	–	–	–	✓	–	–	–	–
tos	–	–	–	–	–	–	✓	–	–	–
proto	–	–	–	–	–	–	–	✓	–	–
sport	–	–	–	–	–	–	–	–	✓	–
dport	–	–	–	–	–	–	–	–	–	✓

✓: Flow identification performed

–: Flow identification not performed

8.7.2 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 9).

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

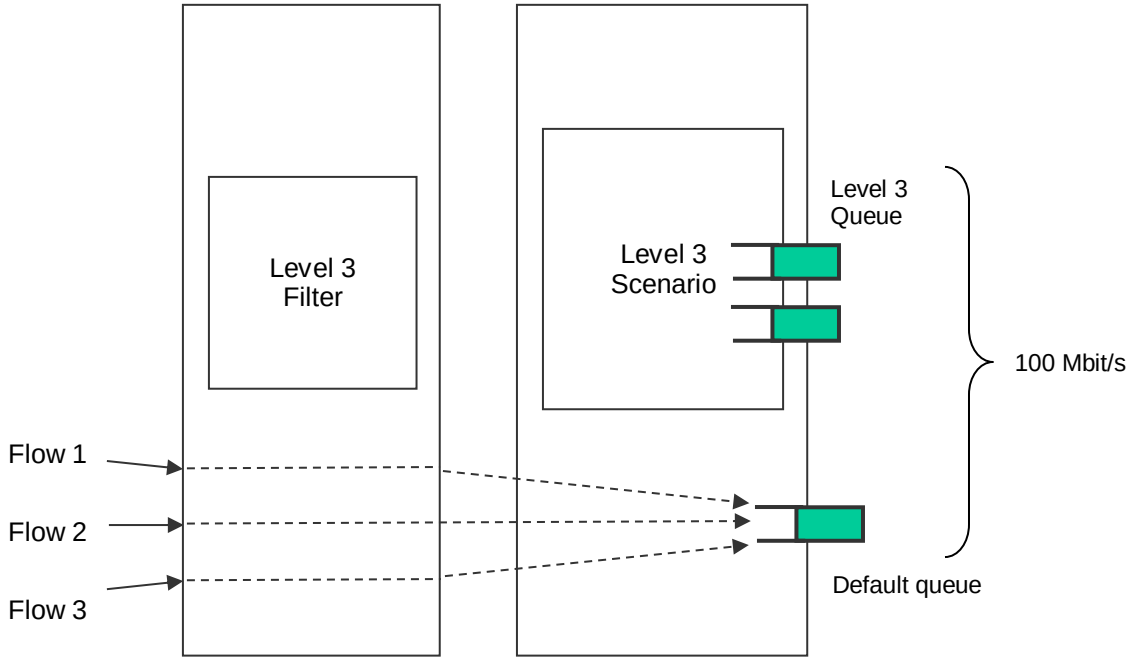
Also, assume the following three types of traffic were input:

- 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 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

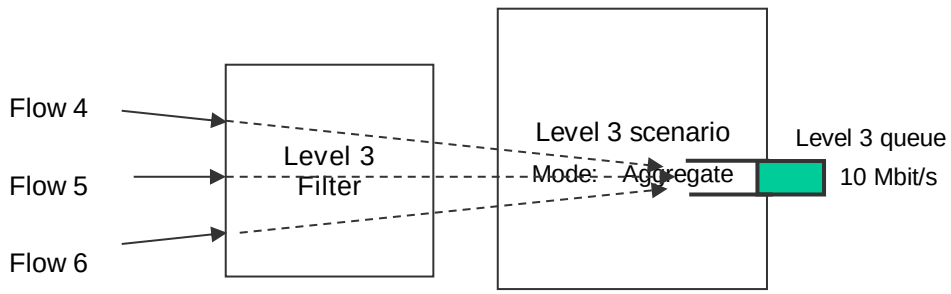
Also, assume the following three types of traffic were input:

- 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

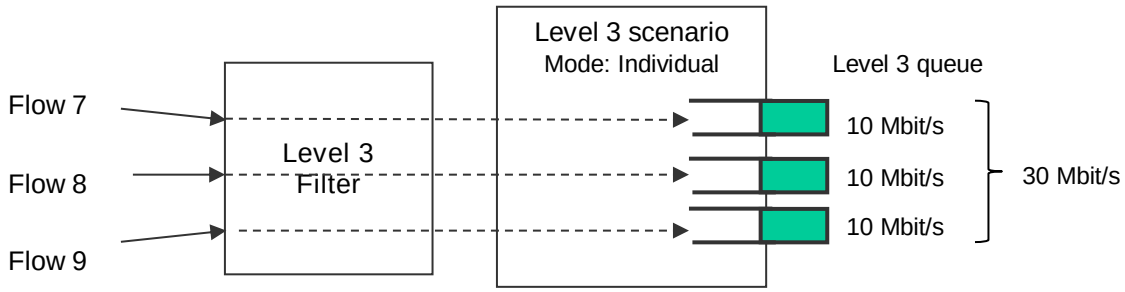
Also, assume the following three types of traffic were input:

- 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.



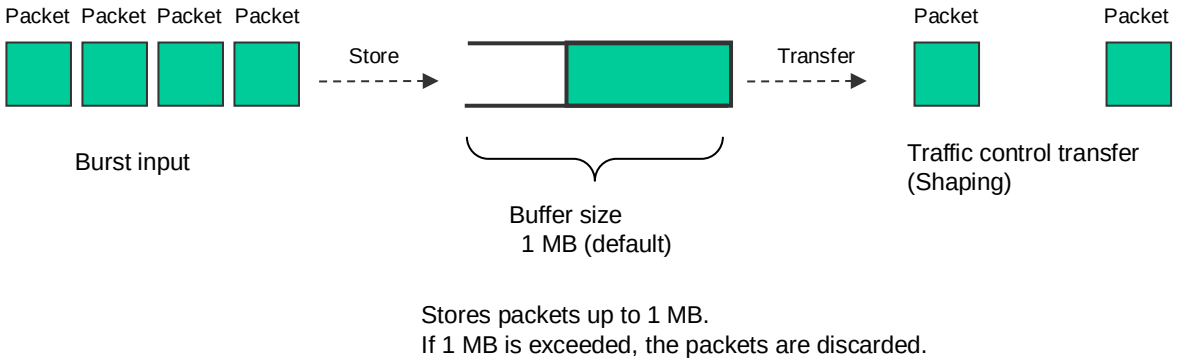
Note:

Monitoring Manager 2 displays a scenario in the individual queue mode as one queue in the same way as the aggregate queue mode. Individual queues are 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 1 MB.
If 1 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 10.)

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

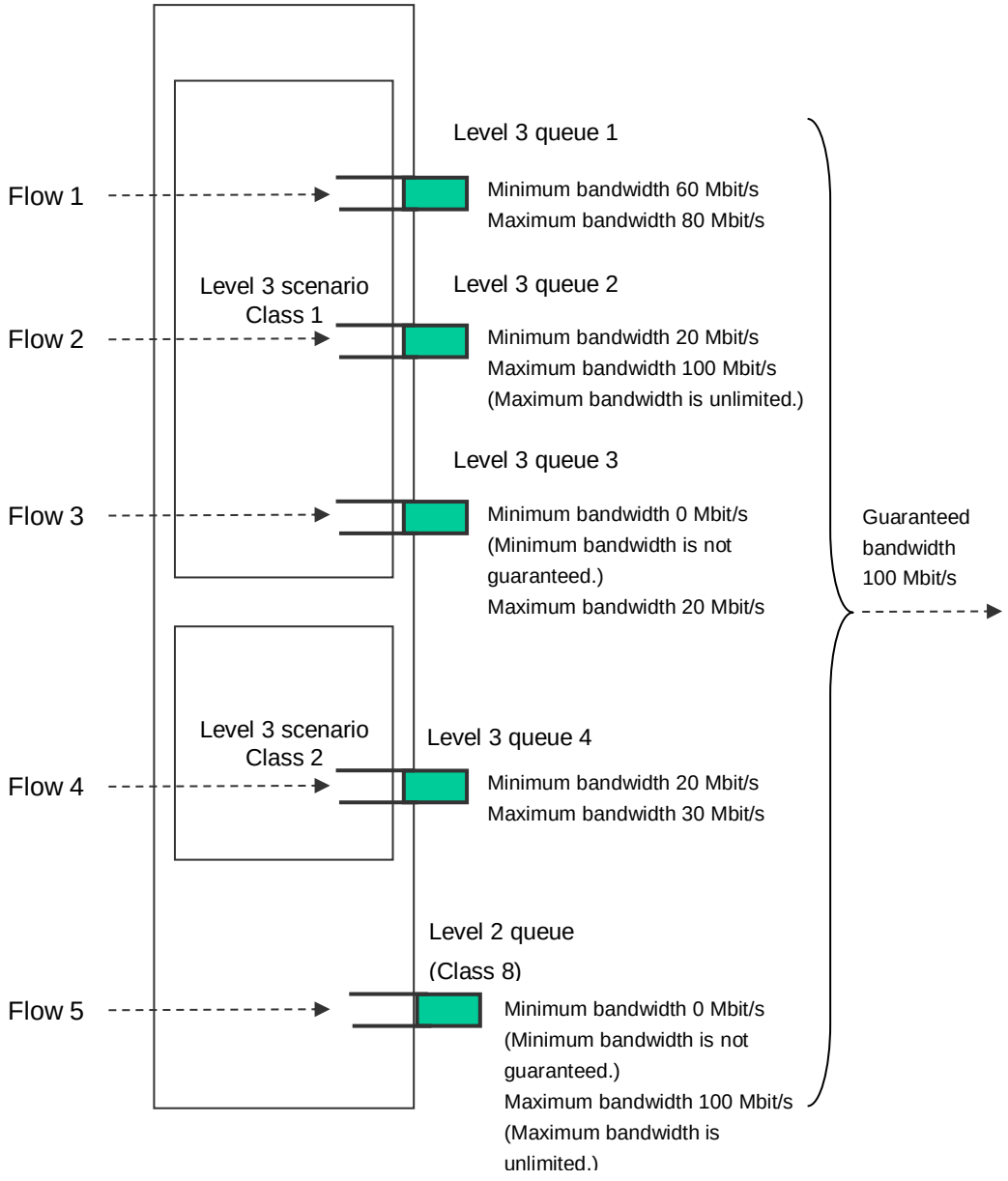
A class (queue priority) can be specified for Level 2 or lower queues.

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 9, 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 10 Mbit/s, maximum bandwidth unlimited)
- Level 3 queue 3 (class 1, minimum bandwidth not guaranteed, maximum bandwidth 10 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 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 not limited 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 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).

8.7.3 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.

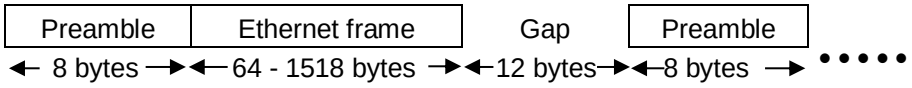


Figure Gaps and preambles of the Ethernet frame

The following CLI commands are available for communication gap mode:

set bandwidth mode {gap [<size>] | no_gap}	Enables/disables inter-frame gaps and preambles in the communication bandwidth settings. The default value is "no_gap (disabled)". 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.

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.

(Blank page)

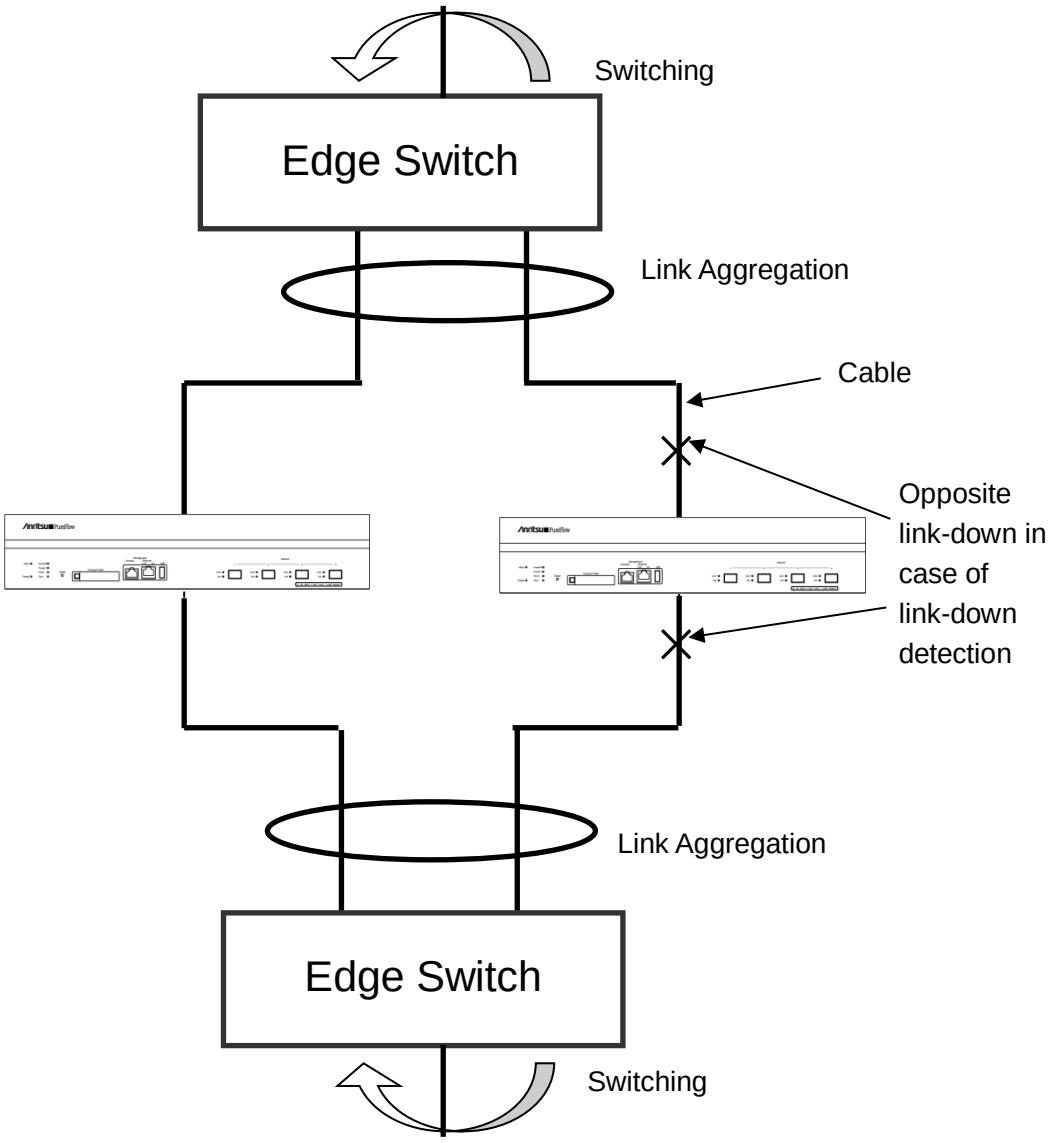
Chapter 9 Link-down Transfer

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:

set lpt {enable | disable}	Enables and disables the function to transfer an alarm by bringing down the link of the communicating Port (1/2) when the link of Port (1/1) is down, and the function to transfer an alarm by bringing down the link of the communicating Port (1/1) when the link of Port (1/2) is down.
show lpt	Displays whether link down transfer is enabled or disabled.

The following is a command execution example:

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

(Blank page)

Chapter 10 SSH

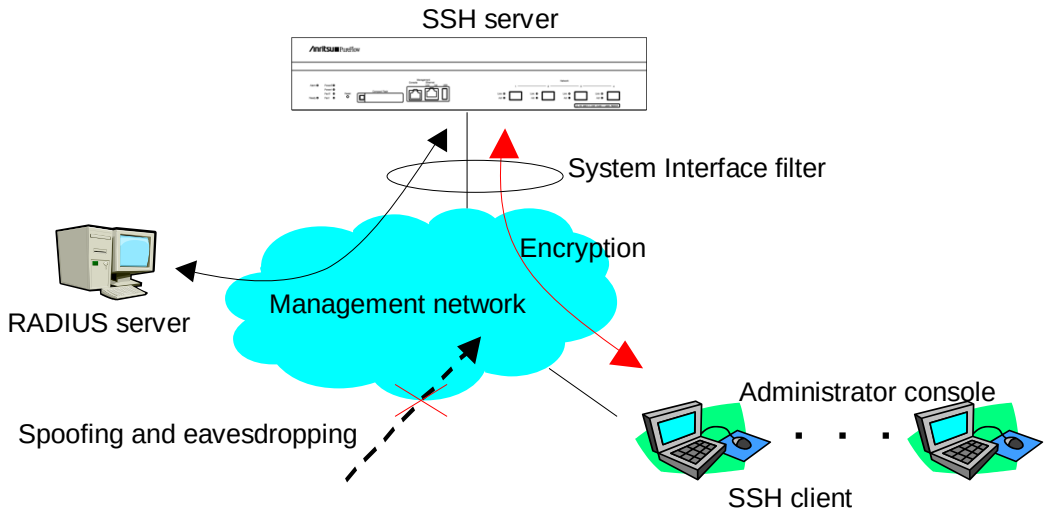
This chapter describes the SSH (Secure Shell) feature.

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 15 “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 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)

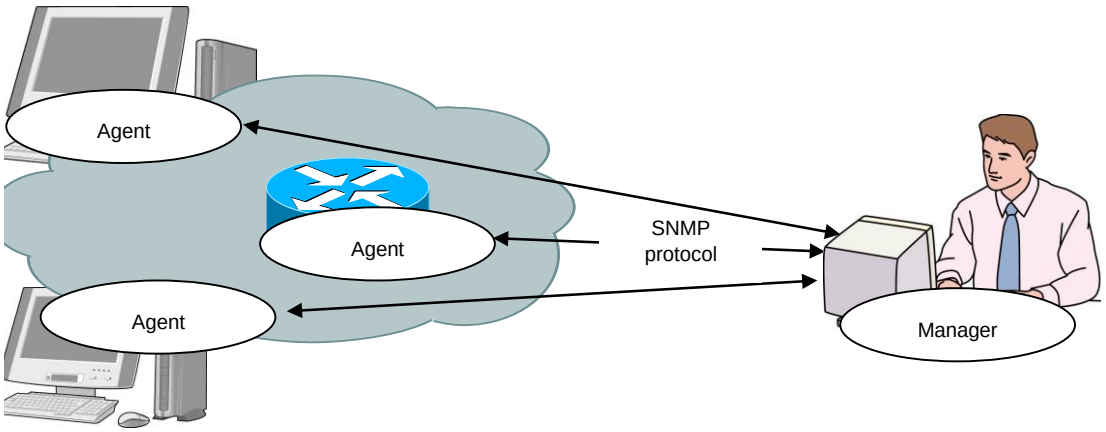
Chapter 11 SNMP Setting

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.

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:

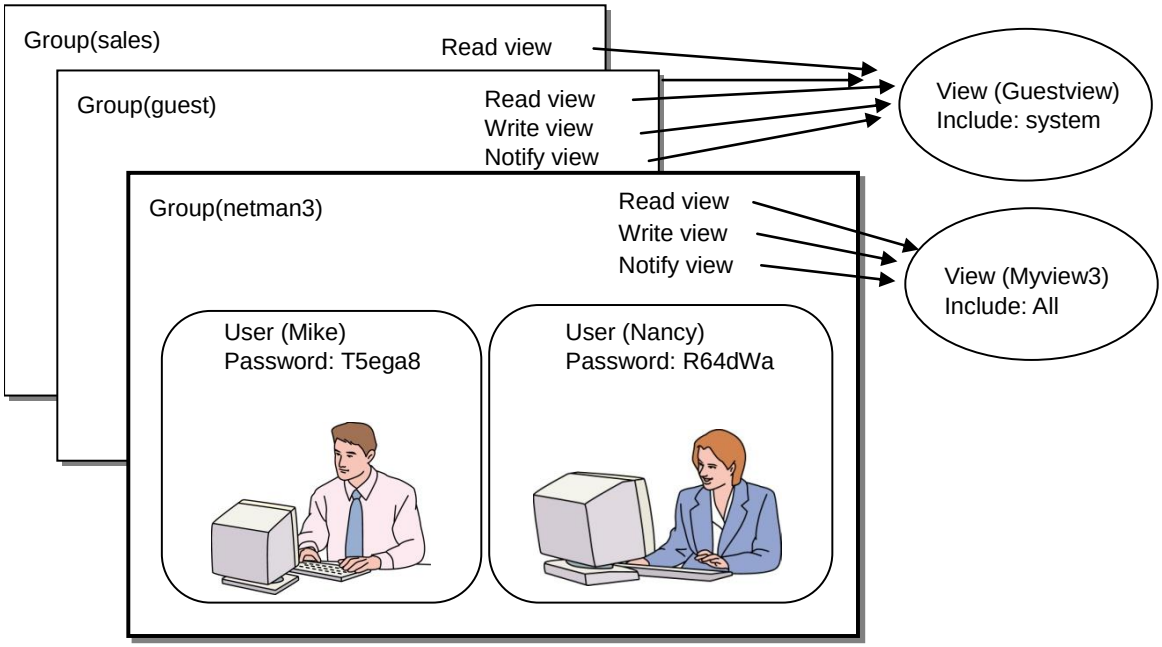
PureFlow(A)> add snmp view All iso included
PureFlow(A)> add snmp view myview1 interface included
PureFlow(A)> add snmp community netman1 version v1 view myview1 permission rw

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:

add snmp group <group_name> [auth_type {auth | noauth}] [read <readview>] [write <writeview>] [notify <notifyview>]	Adds an SNMPv3 group.
delete snmp group <group_name>	Deletes a group.
add snmp user <user_name> <group_name> [auth_type {auth | noauth}] [password <auth_password>]	Adds an SNMPv3 user. To set the password, ensure the length is from 8 to 24 characters.
delete snmp user <user_name>	Deletes a user.
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 group [<group_name>]	Shows the set group.
show snmp user [<user_name>]	Shows the set user.
show snmp view [<view_name>]	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 | queueallocrecovery | 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” “queueallocrecovery” “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 12-4
12.2.1 Scenario counter 12-4
12.2.2 Scenario buffer information 12-5
12.2.3 Rate measurement 12-6
12.2.4 Determining the scenario parameters 12-7

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:

show counter [brief]	Shows the counter for all Network ports and system interface. Specify “brief” to show an overview.
show counter {<slot/port> | system}	Displays the counter of the specified Network port or system interface.
clear counter [<slot/port> | system]	Clears the Network port/system interface counter.

12.2 Scenario Statistics

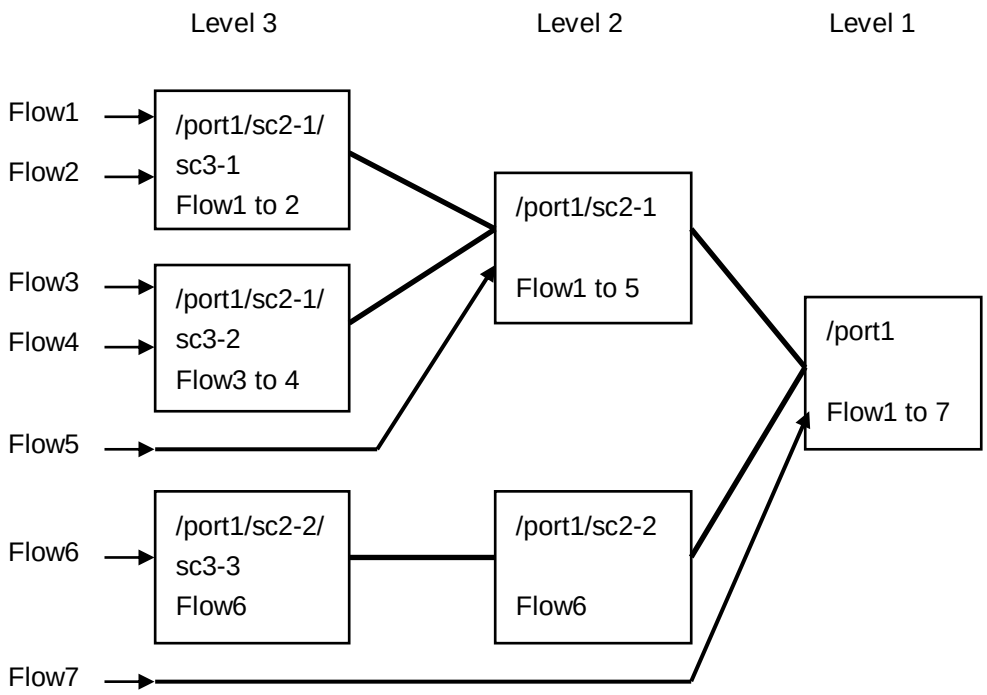
The scenario statistics contain the scenario counter, scenario buffer information, and rate measurement.
This information is the statistics for each scenario.

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:

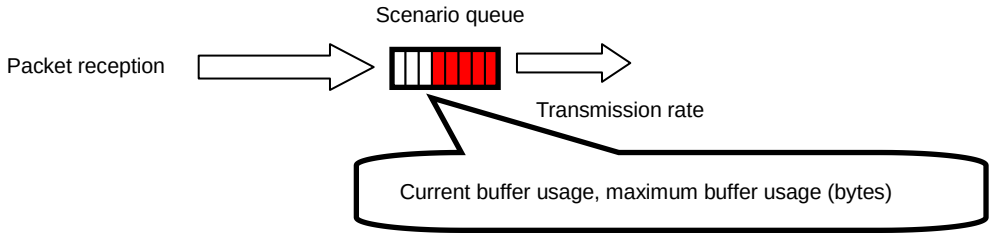
show scenario counter name <scenario_name>	Shows a scenario counter.
show scenario counter summary	Displays a list of scenario counters.
clear scenario counter name <scenario_name>	Deletes a scenario counter.
clear scenario counter all	Clears all scenario counters.

For <scenario_name>, specify the scenario set by the “add scenario” command.

12.2.2 Scenario buffer information

This is the buffer information per scenario.
The scenario buffer information shows the following:

- Buffer usage and use rate
- Buffer peak hold (the maximum value of buffer usage)
- Peak transmission rate (the peak transmission rate over the last 1 minute)
- Average transmission rate (the average transmission rate over the last 1 minute)



The following CLI commands can be used for the scenario buffer information:

show scenario info name <scenario_name>	Shows the buffer information for the scenario.
show scenario info summary	Displays a list of buffer information for the scenario.
clear scenario peakhold buffer name <scenario_name>	Clears the maximum buffer usage of 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:

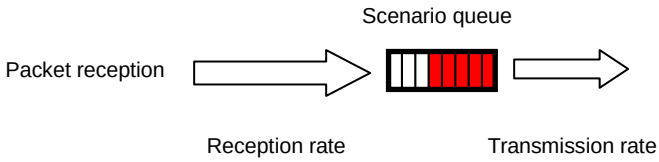
Monitoring Manager 2 V1.1.1 does not show the transmission rate information. To show the transmission rate information, use Monitoring Manager 2 V1.2.1 or later.

Note:

In the individual queue mode scenario, the CLI commands display the buffer information of the fail action queue instead of the individual queues.
Use the aggregate queue mode scenario in measurement of the following pages.

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:

monitor rate <scenario_name> [<num>]	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)>

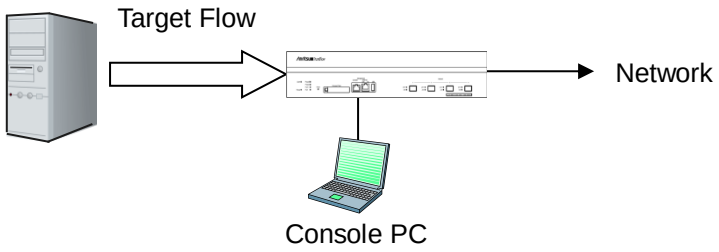
Note:
"bps" in CLI means bits per second.

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/measscenario action aggregate bufsize 100M
PureFlow(A)> add filter scenario /port1/measscenario filter measflow ipv4 sip 192.168.10.9
```

Start the flow, and measure the rate for the target scenario.

```
PureFlow(A)> monitor rate /port1/measscenario 3
Scenario Name : "/port1/measscenario"

  Times[s]          Rcv Rate[kbps]          Trs Rate[kbps]
-------------------------------------------------------------
         1              3587.562                3587.562
         2              3482.826                3482.826
         3              3624.692                3624.692
-------------------------------------------------------------
   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 appox. 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.

Chapter 13 Top Counter

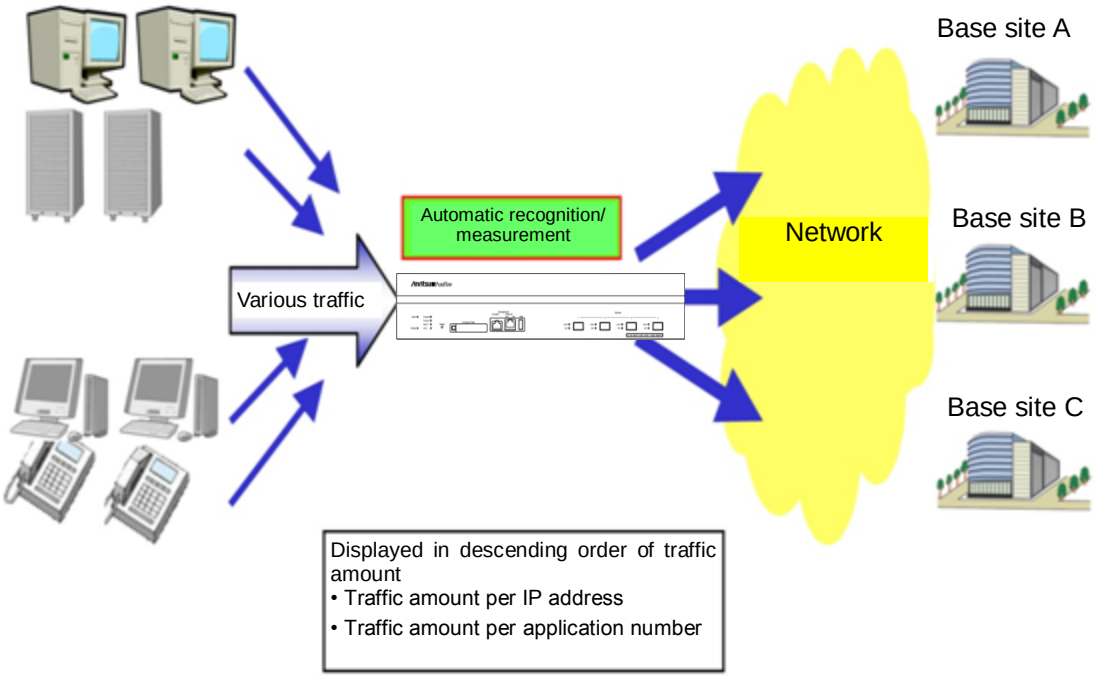
This chapter describes the top counter feature.

13.1 Overview 13-2
13.2 Display Unit of the Top Counter..... 13-2
13.3 Measurement Range of the Top Counter..... 13-3
13.4 Traffic Counter 13-3
13.5 Measuring Traffic at Specific Application Ports 13-4
13.6 Operation Command List 13-4
13.7 Operation Procedure..... 13-5
13.8 Operation Example 13-6
13.9 Cautions 13-8

13.1 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.



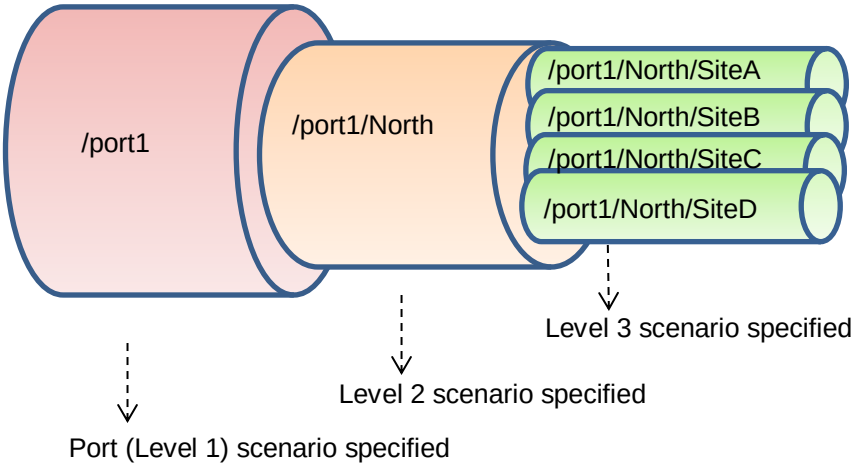
13.2 Display Unit of the Top Counter

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)

13.3 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.

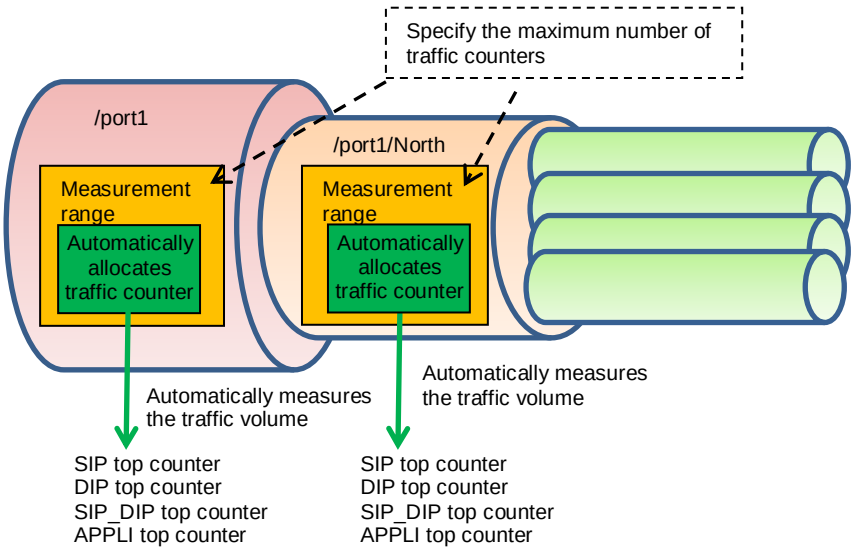


For example, to observe the traffic that consumes the most communication bandwidth in the traffic passing through the level n scenario, specify the level n scenario as the measurement range. This allows you to grasp the traffic with the largest amount of transmission in the traffic input to the scenario.

13.4 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.



13.5 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.

13.6 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.

13.7 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 13.9 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.

13.8 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 13.9 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)>
```

13.9 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) Monitoring Manger 2 V1.1.1 does not show the top counter information.
To show the top counter information, use Monitoring Manager 2 V1.2.1 or later.

(4) 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.

(5) 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.

(6) 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.

(7) The top counter is updated about 1 minute after the collection cycle of the top counter is reached.

(8) 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.

Chapter 14 WebAPI

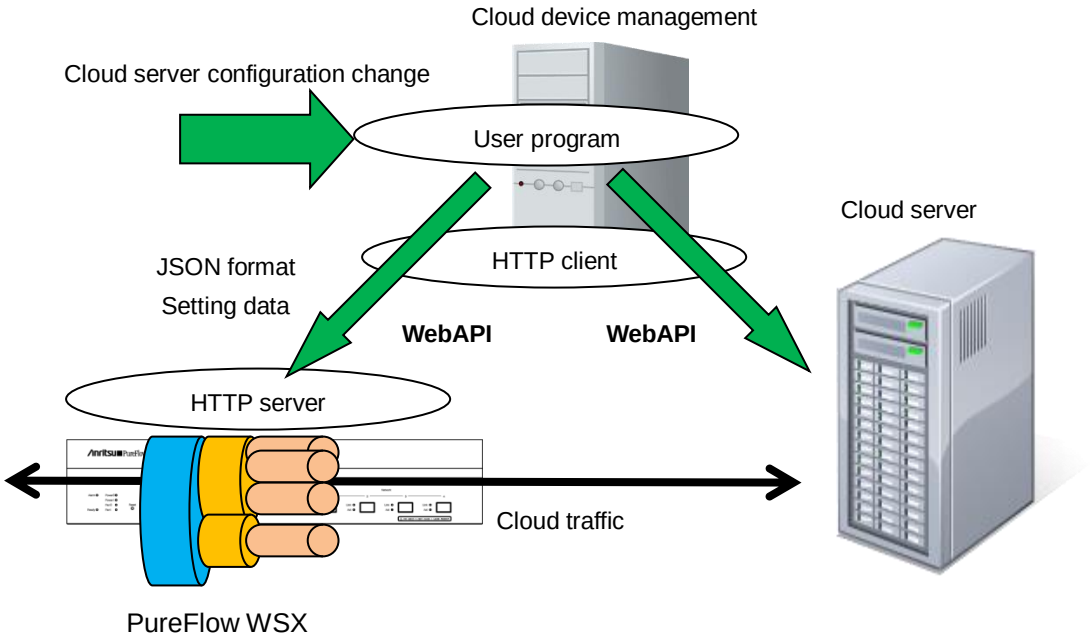
This chapter describes the WebAPI (Web Application Program Interface) feature.

14.1 Overview 14-2
14.2 Communication Protocol 14-3
14.3 HTTP Methods 14-3
14.4 JSON Format 14-4
14.5 API List 14-5
14.6 Common Error Messages 14-6
14.7 List of Error Messages 14-7

14.1 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 bandwidth control 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.

14.2 Communication Protocol

The WebAPI feature use HTTP or HTTPS as the communication protocol. To set up the communication protocol, use the following commands.

set webapi protocol {normalhttp | httpsecure}	Sets the communication protocol of the WebAPI. normalhttp: Use HTTP. httpsecure: Use HTTPS. It is not possible to simultaneously use of HTTP and HTTPS.
show webapi	Displays the settings of the WebAPI.

14.3 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.

14.4 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”

14.5 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
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
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”.

14.6 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.
Required parameter is not specified.	The required parameter is not specified.
Specified command is invalid when GET request.	The command (add/update/delete) cannot be specified by the GET method.
Specified command is invalid when POST request.	The command (get) cannot be specified by the POST method.
WebAPI session is full.	Maximum WebAPI sessions exceeded.
Failed to create pipe.	Cannot create the pipe for internal communication.
No response message from LR.	No response from internal communication.

14.7 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 Minimum Bandwidth is invalid. (Valid from 0, 1 k to 10 G) ・The specified Minimum Bandwidth is invalid.
	Specified Peak Bandwidth is invalid. (Valid from 2 k to 10 G) ・ The specified 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 100 M) ・ 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.
	Could not Add the Scenario. ・ The scenario cannot be registered.
	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 300000) ・ The specified maxquenum is out of range.
	Extended number of scenario is not licensed. ・ It is not possible to register a scenario exceeding the limit of the scenario license. ・ It is not possible to set the maxquenum parameter exceeding the limit of the scenario license.
	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.
	failaction is not specified. ・ fail_min_bw, fail_peak_bw, and fail_class cannot be specified without specifying failaction.

Add a scenario (Continued)	Specified Failaction is invalid. • fail_min_bw, fail_peak_bw, and fail_class can be specified only when forwardattribute is specified as failaction.
	Specified scenario has packets in buffer. Please wait until the buffer becomes empty, and try again. • The specified scenario is sending packets. Wait until it completes, and retry.

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 Minimum Bandwidth is invalid. (Valid from 0, 1 k to 10 G) ・The specified Minimum Bandwidth is invalid.
	Specified Peak Bandwidth is invalid. (Valid from 2 k to 10 G) ・The specified 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 100 M) ・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.
	Could not Update the Scenario. ・The scenario cannot be changed.
	Specified Max Q Num is invalid. (Valid from 1 to 300000) ・The specified 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.
	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
Delete 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.
	Down level hierarchy scenario exists. ・A lower level scenario exists.
	Could not Delete the Scenario. ・The scenario cannot be deleted.

API	Error Messages
Get scenario information	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 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 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.
	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.
	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 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.

API	Error Messages
Add a filter (Continued)	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, Or Start - End) ・ 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.
	Could not Delete the Filter. ・ The filter cannot be deleted.

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.
	Could not add the rulelist. • The rule list cannot be registered.

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.
	Could not delete the rulelist. • The rule list cannot be deleted.

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.) • 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 number of entries for the specified rule list exceeds the limit (512 records).
	Maximum number of total rulelist entry was exceeded. • The number of entries of all rule lists exceeds the registration limit (64000 records).
	Specified rulelist entry is already in use. • The specified rule list entry had already been registered.
	Rulelist entry and rulelist is not same type. • The type is different from that of the target rule list.
	Could not add the rulelist entry. • The rule list entry cannot be registered.

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.) ・ 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 that of the target rule list.
	Specified rulelist entry is not used. ・ The specified rule list entry does not exist.
	Could not delete the rulelist entry. ・ The rule list entry cannot be deleted.

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.) ・ The rule list name is invalid.
	Specified rulelist name is not used. ・ The specified rule list does not exist.

API	Error Messages
Save configuration	configuration save is in progress ・ The configuration save is in progress

API	Error Messages
Get configuration information	None

(Blank page)

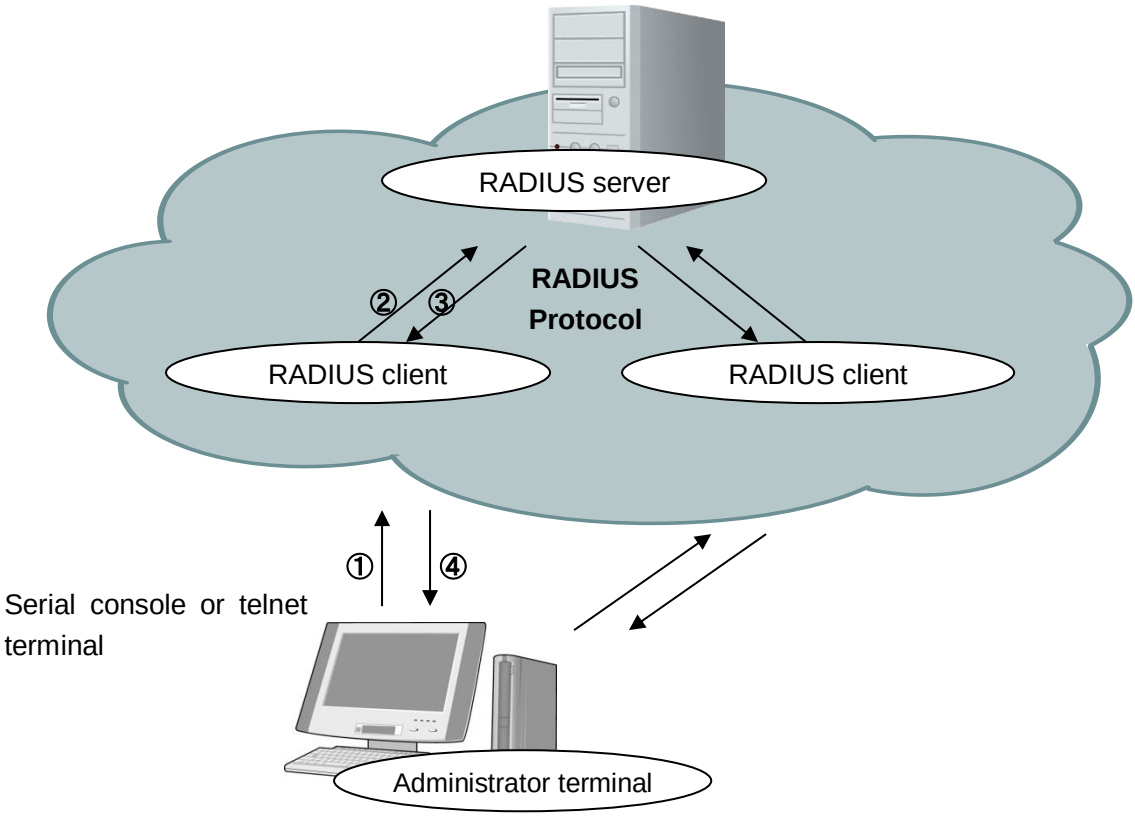
Chapter 15 RADIUS

This chapter describes the RADIUS (Remote Authentication Dial In User Service) feature.

15.1 Overview 15-2
15.2 Controlling Login Authentication 15-3
15.3 Controlling Login Mode 15-3
15.4 Setting Up the RADIUS Feature 15-4
15.5 RADIUS Server Settings 15-5

15.1 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 PureFlowWSX to the RADIUS server.

③ An authentication response is sent from the RADIUS server to the RADIUS client.

④ PureFlowWSX permits connection from the administrator terminal based on the received authentication response.

15.2 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.

15.3 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.

15.4 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)>
```

15.5 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 PureFlowWSX.

User ID

Set the user ID.

Authentication method

Specify the same authentication method (CHAP or PAP) as the authentication method set for PureFlowWSX.

Password

Set the login password.

Service type

Specify this parameter as required. If the RADIUS server does not give notification of any service type, PureFlowWSX 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, PureFlowWSX 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 PureFlowWSX 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
```

Chapter 16 Downloading and Uploading Data

This chapter describes how to download or upload software and configuration data.

16.1 Downloading/Uploading Software 16-2
16.1.1 Downloading software from a CF card 16-2
16.1.2 Uploading software to a CF card 16-2
16.1.3 Downloading software from a USB flash drive 16-3
16.1.4 Uploading software to a USB flash drive 16-3
16.1.5 Downloading software via TFTP 16-4
16.1.6 Downloading software via FTP 16-4
16.2 Downloading the Software Update Patch 16-6
16.2.1 Downloading software Update Patch from a CF card 16-6
16.2.2 Downloading software Update Patch from a USB flash drive 16-6
16.3 Downloading/Uploading Configuration Data 16-7
16.3.1 Downloading configuration data from a CF card 16-7
16.3.2 Uploading configuration data to the CF card 16-7
16.3.3 Downloading configuration data from a USB flash drive 16-8
16.3.4 Uploading configuration data to a USB flash drive 16-8
16.3.5 Downloading configuration data via TFTP 16-9
16.3.6 Uploading configuration data via TFTP 16-9
16.3.7 Downloading configuration data via FTP 16-10
16.3.8 Uploading configuration data via FTP 16-10
16.4 Restarting the Software 16-11

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”.

16.1 Downloading/Uploading Software

16.1.1 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 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.

16.1.2 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 nf7600.bin
Upload as “nf7600.bin” to Flash Memory Card (y/n)? y
Loading .....
Done.
PureFlow(A)>
```

16.1.3 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.

16.1.4 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)>
```

16.1.5 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.

16.1.6 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:ftpuser    (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.

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 how to obtain the proper object file, contact the supplier.

16.2 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.

16.2.1 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
Appling file system patch ..... done
Appling apps patch ..... done
Appling 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.

16.2.2 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
Appling file system patch ..... done
Appling apps patch ..... done
Appling 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.

16.3 Downloading/Uploading Configuration Data

16.3.1 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.

16.3.2 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.

16.3.3 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.

16.3.4 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.

16.3.5 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.

16.3.6 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.

16.3.7 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.

16.3.8 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.

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.

16.4 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)

Chapter 17 Network Bypass Function

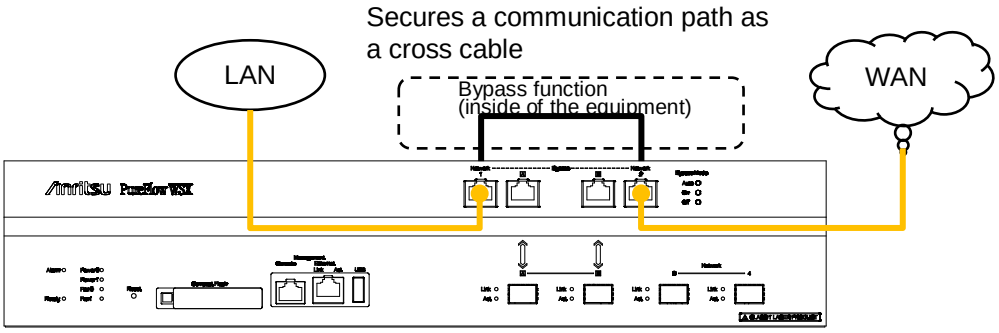
This chapter describes the bypass function and setting.

17.1 Overview 17-2
17.2 Setting and Checking the Function 17-3
17.3 Precautions 17-6

17.1 Overview

NF7603A and NF7604A 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.

When the system interface communication port is set to the Ethernet port:
PureFlow(A)> set bypass on
PureFlow(A)>

When the system interface communication port is set to the Network port:
PureFlow(A)> set bypass on
System interface might be disconnected from the network, ok (y/n)? y
Done
PureFlow(A)>

To switch the Network port to the bypass state temporarily for 300 seconds, execute the following command.

When the system interface communication port is set to the Ethernet port:
PureFlow(A)> bypass time 300 on
Current time : Feb 29 17:38:47

```
Expiring time:    Feb 29    17:43:47
PureFlow(A)>
```

When the system interface communication port is set to the Network port:

```
PureFlow(A)> bypass time 300 on
System interface might be disconnected from the network, ok (y/n)? y
Current time : Feb 29    17:38:47
Expiring time:    Feb 29    17:43:47
Done
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.

If device management is performed via the Network port, remote management of this device cannot be performed in the bypass connection state. To perform remote management in the bypass connection state, manage the device via the Ethernet port.

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 “Command Reference Traffic Shaping Edition(NF7600-W022E)”.

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-TS FP 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 packet length	set port maxpacketlen	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 port (ethernet/network)	ethernet	ethernet/network
	set ip system port network (in <slot/port>/all)	all (all Network ports)	1/1,1/2,all * When communication port is Network
	set ip system port network (vid <VID>/none)	none (no VLAN Tag)	0 to 4094/none * When communication port is Network
	set ip system port network (tpid <tpid>)	0x8100 (IEEE802.1Q VLAN Tag)	0x0000 to 0xffff * When communication port is Network
	set ip system port network (inner-vid <VID>/none)	none (No Inner-VLAN Tag)	0 to 4094/none * When communication port is Network
	set ip system port network (inner-tpid <tpid>)	0x8100 (IEEE802.1Q VLAN Tag)	0x0000 to 0xffff * When communication port is Network
	set ip system gateway(IPv4)	(None)	IPv4 Address
	set ip system gateway(IPv6)	(None)	IPv6 Address
Auto reboot	set autoreboot	enable	enable/disable
Flow identification mode	set filter mode	default	default,vid,cos,inner-vid,inner-cos,sip,dip,tos,proto
Flow aging time	set agingtime	300 seconds	1 to 1800 sec
Communication gap mode setting	set bandwidth mode	no_gap	gap/no_gap
Link-down transfer feature	set lpt	disable	enable/disable

Setting item	Command	Default value	Setting range
Telnet connection setting	set telnet	enable	enable/disable
SSH connection setting	set ssh	enable	enable/disable
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

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 AC0127 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.
	Fan #N removed	Fan unit is removed. (#N is 0 or 1)	No recovery measure is required.

Severity	Syslog message	Occurs when	Action
Alert (1) (Continued)	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 recover	The usage of System Buffer %s exceeded 90% and then dropped below 50%.	No recovery measure is required.
	Critical error on FCPU Core[#N], Code[#M] Data1[0xxxxxxxxx] Data2[0xxxxxxxxx]	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 or 2)	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.
	Detected Queue blocking.	The packet stagnation was detected in individual queues in the device.	Contact Anritsu Networks support.
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.

Severity	Syslog message	Occurs when	Action
Notice (5) (Continued)	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.
	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.
	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. Cycle / Scenarios / Monitoring Manager 2 connections 10 sec / 2000 / 2 10 sec / 4000 / 1 30 sec / 10000 / 2 30 sec / 20000 / 1 60 sec / No limit / 4

Severity	Syslog message	Occurs when	Action
Notice (5) (Continued)	Session limits between monitoring manager is released.	The number of Monitoring Manager 2 connections exceeded the limit, and then dropped below it.	No recovery measure is required.
	Terminate monitoring manager session due to System Packet Buffer full.	A Monitoring Manager 2 connection is disconnected due to the System Packet Buffer exceeds the limit when the communication port of the system interface is Network.	The system cannot communicate with a Monitoring Manager 2 using the system interface in a state in which the usage of the System Packet Buffer has exceeded the limit. Check the traffic state and various settings. The connection will be reconnected automatically after recovery of the System Packet Buffer usage.
	Monitoring manager session connected. (xxx.xxx.xxx.xxx)	Session connected to the monitoring manager 2. (xxx.xxx.xxx.xxx)	No recovery measure is required.
	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)	Pipe #N changed Operate from Down.	Pipe link-up occurred. (#N is 1)	No recovery measure is required.
	Pipe #N changed Down from Operate.	Pipe link-down occurred. (#N is 1)	Check the following. ・If any link-down occurred at the specified location
	Port #N/#M changed Up from Down.	Port link-up occurred (#N is 1) (#M is 1 to 2)	No recovery measure is required.

Severity	Syslog message	Occurs when	Action
Informational (6) (Continued)	Port #N/#M changed Down from Up.	Port link-down occurred (#N is 1.\) (#M is 1 to 2)	Check the following. ・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 2)	Check the following. ・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
	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. ・If any cable disconnection occurred ・If the right cable is used
	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.
	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 “Command Reference Traffic Shaping Edition (NF7600-W022E)”.

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. • 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. • 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. • If the power cable is connected • If the supply voltage is within the valid range (AC 100 V to 240 VAC0127 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. • 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.
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.

MIB object name	Name of command setting	Occurs when	Action
pfGsxSystemHeatAlarmEvent(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 40°C or lower.
pfGsxSystemHeatRecoveryEvent(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)	queueallocrecovery	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.
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.

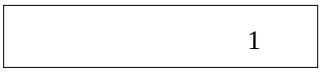
(Blank page)

Appendix D Enterprise MIB List

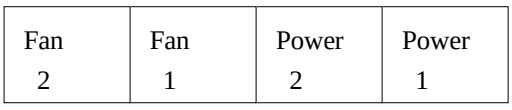
The table 4 shows a list of Enterprise MIB objects of PureFlow WSX.

Table 4 List of Enterprise MIB Objects of PureFlow WSX

MIB group	MIB object name	Description
pureFlowGsMib		PureFlow GS Enterprise MIB tree. The object ID is 1.3.6.1.4.1.1151.2.1.7. Objects in the tree and their IDs (in parentheses) are as follows: PureFlow GS Enterprise MIB tree is a common MIB tree for the PureFlow GS series. This manual describes PureFlow WSX MIB objects.
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)	The table for the CPU and memory use rates of the fowarding-system CPU. This table contains the following objects:
	pfGsSystemFcpuEntry(1.3.6.1.4.1.1151.2.1.7.1.12.1)	The entry table for the CPU and memory use rates of the fowarding-system CPU. The table index is pfSystemFcpuIndex. This table contains the following objects:

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 
	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.
pfGsBypass(1.3.6.1.4.1.1151.2.1.7.13)	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.
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 1

MIB group	MIB object name	Description
pfGsModule(1.3.6.1.4.1.1151.2.1.7.2) (Continued)	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): 10GE/2SFP xge4sfp(6): 10GE/4SFP ge4sfp(7): GE/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.
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 

MIB group	MIB object name	Description
pfGsPower(1.3.6.1.4.1.1151.2.1.7.3) (Continued)	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 pfGsFanIndex. 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 Fan 2 | Fan 1 | Power 2 | Power 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.
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.

MIB group	MIB object name	Description
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
	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.
	pfGsxScenarioStatisticsHCRxOctets(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.
	pfGsxScenarioStatisticsHCRxPackets(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.

MIB group	MIB object name	Description
pfGsxScenarioStatistics(1.3.6.1.4.1.1151.2.1.7.9) (Continued)	pfGsxScenarioStatisticsHCTxOctets(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.
	pfGsxScenarioStatisticsHCTxPackets(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.
	pfGsxScenarioStatisticsHCDiscardOctets(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.
	pfGsxScenarioStatisticsHCDiscardPackets(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.
	pfGsxScenarioStatisticsDefaultQueRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.16)	Shows the number of received octets of of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.17)	Shows the number of received packets of of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueTxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.18)	Shows the number of transmitted octets of of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueTxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.19)	Shows the number of transmitted packets of of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueDiscardOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.20)	Shows the number of discarded octets of of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueDiscardPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.21)	Shows the number of discarded packets of of the scenario default queue.
	pfGsxScenarioStatisticsDefaultQueHCRxOctets(1.3.6.1.4.1.1151.2.1.7.9.1.1.22)	Shows the number of received octets of 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
pfGsxScenarioStatistics(1.3.6.1.4.1.1151.2.1.7.9) (Continued)	pfGsxScenarioStatisticsDefaultQueHCRxPackets(1.3.6.1.4.1.1151.2.1.7.9.1.1.23)	Shows the number of received packets of 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 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 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 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 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.
	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
	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.

MIB group	MIB object name	Description
pfGsxScenarioInformation(1.3.6.1.4.1.1151.2.1.7.10) (Continued)	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.
	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.

MIB group	MIB object name	Description
pfGsxScenarioInformation(1.3.6.1.4.1.1151.2.1.7.10) (Continued)	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.
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
	pfGsxScenarioStatByScIdRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.4)	Shows the number of received octets of the scenario.
	pfGsxScenarioStatByScIdRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.5)	Shows the number of received packets of the scenario.

MIB group	MIB object name	Description
pfGsxScenarioStatByScId(1.3.6.1.4.1.1151.2.1.7.11) (Continued)	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.
	pfGsxScenarioStatByScIdHCRxOctets(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.
	pfGsxScenarioStatByScIdHCRxPackets(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.
	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.
	pfGsxScenarioStatByScIdDefaultQueRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.16)	Shows the number of received octets of of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.17)	Shows the number of received packets of of the scenario default queue.

MIB group	MIB object name	Description
pfGsxScenarioStatByScId(1.3.6.1.4.1.1151.2.1.7.11) (Continued)	pfGsxScenarioStatByScIdDefaultQueTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.18)	Shows the number of transmitted octets of of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueTxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.19)	Shows the number of transmitted packets of of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueDiscardOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.20)	Shows the number of discarded octets of of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueDiscardPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.21)	Shows the number of discarded packets of of the scenario default queue.
	pfGsxScenarioStatByScIdDefaultQueHCRxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.22)	Shows the number of received octets of of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdDefaultQueHCRxPackets(1.3.6.1.4.1.1151.2.1.7.11.1.1.23)	Shows the number of received packets of of the scenario default queue. Note: This object cannot be accessed via SNMPv1. Use v2c or higher for access.
	pfGsxScenarioStatByScIdDefaultQueHCTxOctets(1.3.6.1.4.1.1151.2.1.7.11.1.1.24)	Shows the number of transmitted octets of 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 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 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 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
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
	pfGsxScenarioInfoByScIdDefFlowNum(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.
	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.

MIB group	MIB object name	Description
pfGsxScenarioInfoByScId(1.3.6.1.4.1.1151.2.1.7.12) (Continued)	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.
	pfGsxScenarioInfoByScIdDefBuffRatio(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 %.
	pfGsxScenarioInfoByScIdDefBuff(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.
	pfGsxScenarioInfoByScIdDefPeakBuffRatio(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 %.
	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.

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

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

ScenarioSortIndex: Scenario ID of the scenario The length is 1. Use the same way as scenario ID acquisition in pfGsxScenarioStatByScIdTable.

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 PureFlow WSX.

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 Traffic Shaping Edition (NF7600-W022E)”.

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
	"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>]

API	Key	Value	Relevant 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
	"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>]
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>]

API	Key	Value	Relevant CLI command and parameter
	"peak_bandwidth" (Optional)	Peak bandwidth	[peak_bw <peak_bandwidth>]
	"class " (Optional)	Class	[class <class>]
	"bufsize" (Optional)	Buffer size	[bufsize <bufsize>]
Add 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>]
Delete a scenario (all specified)	"command" (Required)	"delete scenario"	delete scenario
	"scenario_name" (Required)	"all"	all

API	Key	Value	Relevant CLI command and parameter
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, "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".

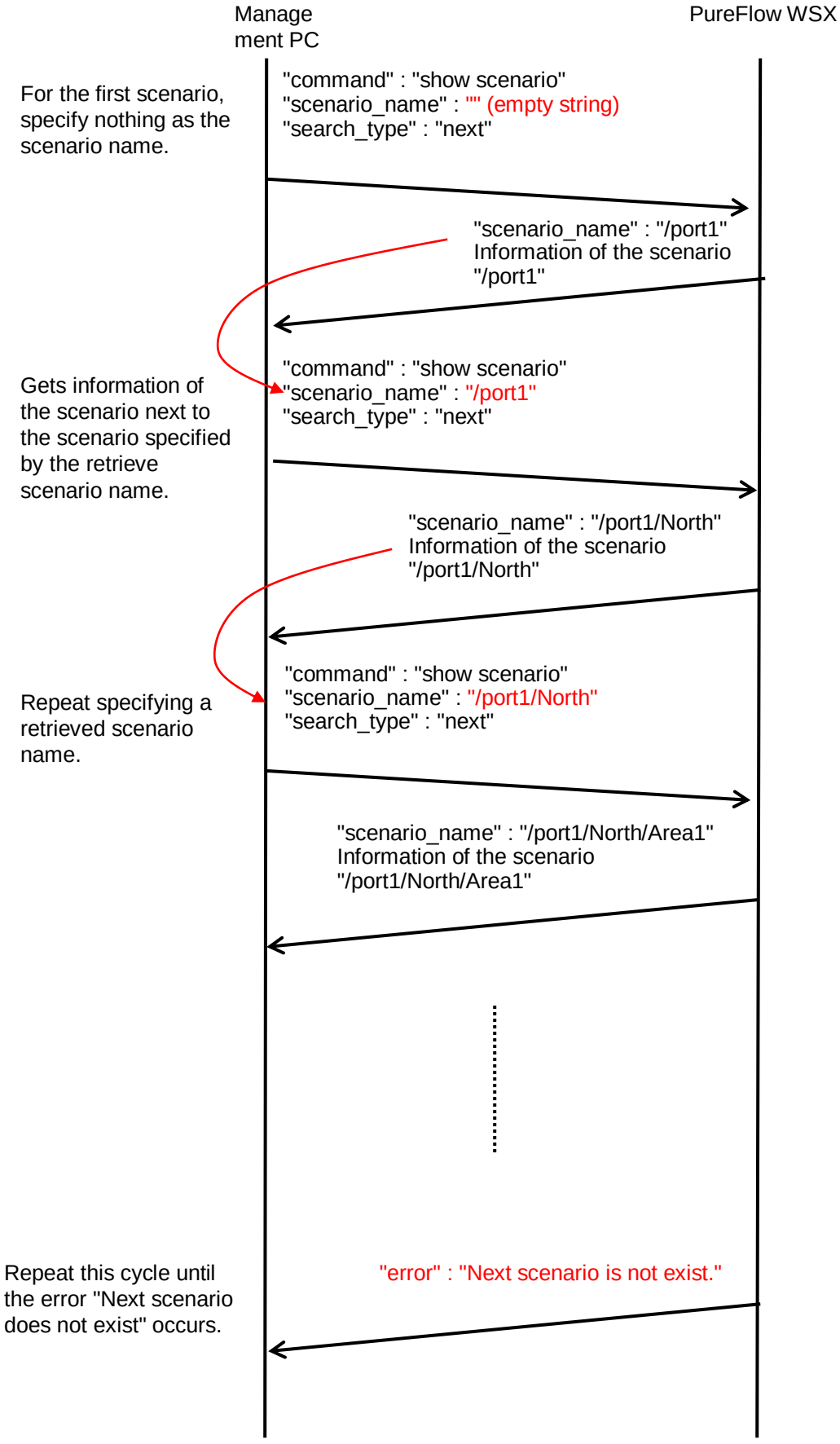
"exact": Gets information of the scenario specified by "scenario_name".
"next": Gets information of the scenario next to the scenario specified by "scenario_name". Information to be retrieved is in the scenario tree order in the same way the CLI command "show scenario".

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.


Manage
ment PC
PureFlow WSX
For the first scenario, specify nothing as the scenario name.
"command" : "show scenario"
"scenario_name" : "" (empty string)
"search_type" : "next"
"scenario_name" : "/port1"
Information of the scenario
"/port1"
Gets information of the scenario next to the scenario specified by the retrieve scenario name.
"command" : "show scenario"
"scenario_name" : "/port1"
"search_type" : "next"
"scenario_name" : "/port1/North"
Information of the scenario
"/port1/North"
Repeat specifying a retrieved scenario name.
"command" : "show scenario"
"scenario_name" : "/port1/North"
"search_type" : "next"
"scenario_name" : "/port1/North/Area1"
Information of the scenario
"/port1/North/Area1"
Repeat this cycle until the error "Next scenario does not exist" occurs.
"error" : "Next scenario is not exist."

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" (Priority)	Filter precedence	[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" (Priority)	Filter precedence	[priority <filter_pri>]

API	Key	Value	Relevant CLI command and parameter
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>]
	"sip" or "sip list" (Optional)	Source IPv4 address or rule list name	[sip [list] {<src_IP_address> | <list_name>}]
	"dip" or "dip list" (Optional)	Destination IPv4 address or rule list name	[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	[sport [list] {<sport> | <list_name>}]
	"dport" or "dport list" (Optional)	Destination port number or rule list name	[dport [list] {<dport> | <list_name>}]
	"priority" (Priority)	Filter precedence	[priority <filter_pri>]

API	Key	Value	Relevant CLI command and parameter
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>]
	"sip" or "sip list" (Optional)	Source IPv6 address or rule list name	[sip [list] {<src_IP_address> | <list_name>}]
	"dip" or "dip list" (Optional)	Destination IPv6 address or rule list name	[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	[sport [list] {<sport> | <list_name>}]
	"dport" or "dport list" (Optional)	Destination port number or rule list name	[dport [list] {<dport> | <list_name>}]
	"priority" (Priority)	Filter precedence	[priority <filter_pri>]
Delete a filter (all specified)	"command" (Required)	"delete filter"	delete filter
	"scenario_name" (Required)	"all"	all
Delete a filter (specified scenario)	"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 filter 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)	How to get "exact": Gets information of the specified filter. "next": Gets information of the filter next to the specified filter. When omitted, "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": Gets information of the filter specified by "scenario_name" and "filter_name".
"next": Gets information of the filter next to the filter specified by "scenario_name" and "filter_name". Information to be retrieved is in alphabetical order in the same way as the CLI command "show filter". If the last filter of the scenario is specified, information of the first filter of the next scenario is retrieved.

To get information of a specific filter, specify a scenario name and a filter name and "exact". To get information of all filters of all scenarios in the same way as the CLI command "show filter 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 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>

API	Key	Value	Relevant CLI command and parameter
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
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>

API	Key	Value	Relevant CLI command and parameter
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, "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 as a CLI command.

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 address192.168.1.1-192.168.1.1
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
Save configuration	"command" (Required)	"save config"	save config
Save configuration information Get running status	"command" (Required)	"show save status"	None

API for saving configuration

The API for saving a configuration finishes without waiting for completion of saving. Saving the configuration runs in the background. If saving the configuration is specified by this API while another saving operation is running, the error message “configuration save is in progress” is shown. For time required to save a configuration, see Chapter 3 “Setting Basics”.

API for getting configuration saving status

There is no relevant CLI command. This API gets the running status of configuration saving.

“configuration save is in progress”: The configuration is being saved.
“configuration save is not in progress”: The configuration has been saved.

Appendix G WebAPI Sample Programs

Python is a widely used programming language for Web API. Python provides HTTP and JSON libraries, and suits the PureFlow WSX WebAPI.

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” 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 of PureFlowWSX WebAPI. HTTP
url       = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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 in back ground. Therefore, when several urlopen called, the previous session may not been terminated at PureFlow WSX on the next urlopen. If this operation is repeated, resource of session will be run out at PureFlow WSX, 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 kepping the connection for several settings.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL of PureFlowWSX 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 of PureFlowWSX WebAPI. HTTP
url       = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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 of PureFlowWSX WebAPI. HTTP
url      = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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 scenario information (specified scenario)

```
# -*- coding:  utf-8 -*-
import urllib
import urllib2
import json

# Define URL of PureFlowWSX WebAPI. HTTP
url      = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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
```

(2) Getting scenario information (all scenarios)

(2)-1 A sample program kepping the connection to get all scenarios.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL of PureFlowWSX 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)-2 A sample program open and close the connection for each scenario.

When following sample program is used, resource of session may be run out at PureFlow WSX and urlopen results error depending on the performance of the terminal. If urlopen results error, please use the previous sample program (2)-1.

```
# -*- coding:    utf-8 -*-
import urllib
import urllib2
import json

# Define URL of PureFlowWSX WebAPI. HTTP
url       = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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
```

(3) Getting filter information (specified filter)

```
# -*- coding:    utf-8 -*-
import urllib
import urllib2
import json

# Define URL of PureFlowWSX WebAPI. HTTP
url       = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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
```

(4) Getting filter information (all filters)

(4)-1 A sample program kepping the connection to get all filters.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL of PureFlowWSX 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()
```

(4)-2 A sample program open and close the connection for each filter.

When following sample program is used, resource of session may be run out at PureFlow WSX and urlopen results error depending on the performance of the terminal. If urlopen results error, please use the previous sample program (4)-1.

```
# -*- coding:    utf-8 -*-
import urllib
import urllib2
import json

# Define URL of PureFlowWSX WebAPI. HTTP
url       = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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
```

(5) Getting rule list information (specified rule list entry)

```
# -*- coding:    utf-8 -*-
import urllib
import urllib2
import json

# Define URL of PureFlowWSX WebAPI. HTTP
url       = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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
```

(6) Get rule list information (all rule lists)

(6)-1 A sample program kepping the connection to get all rule lists.

```
# -*- coding: utf-8 -*-
import urllib
import urllib2
import json
import httplib

# Define URL of PureFlowWSX 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()
```

(Continued)

(6)-2 A sample program open and close the connection for each rule list.

When following sample program is used, resource of session may be run out at PureFlow WSX and urlopen results error depending on the performance of the terminal. If urlopen results error, please use the previous sample program (6)-1.

```
# -*- coding:    utf-8 -*-
import urllib
import urllib2
import json

# Define URL of PureFlowWSX WebAPI. HTTP
url       = 'http://192.168.1.1/shapermng/json'
# Define URL of PureFlowWSX 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
```

(Blank page)