Ethernt Switches

MES2300-xx, MES3300-xx, MES3500I-10P, MES5312, MES5316A, MES5324A, MES5332A, MES5400-24, MES5400-48, MES5410-48, MES5500-32

MES Ethernet switches monitoring and configuration via SNMP, firmware version 6.6.4

Document Version	Issue Date	Revisions
Version 1.15	12.08.2024	Sections changed: 3.2 Operation with TFTP server 3.4 Firmware update 4.1 System resources
Version 1.14	24.05.2024	Sections changed: 6.1 Ethernet interface parameters
Version 1.13	18.04.2024	Synchronization with firmware version 6.6.3
Version 1.12	15.03.2024	Synchronization with firmware version 6.6.2.15
Version 1.11	29.02.2024	Synchronization with firmware version 6.6.2.9
Version 1.10	15.12.2023	Sections added: 6.2 Link Aggregation Group (LAG)
Version 1.9	09.10.2023	Sections changed: 4.1 System resources
Version 1.8	07.09.2023	Synchronization with firmware version 6.5.1.4
Version 1.7	18.06.2023	Sections changed: 2 SHORT DESIGNATIONS
Version 1.6	07.04.2023	Sections changed: 12 CONTROL FUNCTIONS
Version 1.5	10.03.2023	Synchronization with firmware version 6.5.0
Version 1.4	30.09.2022	Sections changed: 6.1 Ethernet interface parameters
Version 1.3	29.07.2022	Sections added: 20 VXLAN CONFIGURATION
Version 1.2	31.01.2022	Second issue
Version 1.1	04.02.2021	First issue
Firmware Version	**6.6.4**	

CONTENTS

DOCUMENT CONVENTIONS

Typographic element	Description
[ ]	Square brackets are used to indicate optional parameters in the command line; when entered, they provide additional options.
{}	In the command line, mandatory parameters are shown in curly braces.
«,» **«-»**	In the command description, these characters are used to specify ranges.
«|»	In the command description, this character means 'or'.
«/»	This sign separates possible and default values when specifying variable values.
Calibri Italic	Calibri Italic is used to indicate variables and parameters that should be replaced with an appropriate word or string.
Bold italic	Notes and warnings are shown in bold italic.
<Bold Italic>	Keyboard keys are shown in bold italic within angle brackets.
`Courier New`	Command examples are shown in Courier New Bold.

Notes and Warnings

Notes contain important information, tips, or recommendations on device operation and configuration.

Warnings inform the user about situations that may be harmful to the user, cause damage to the device, malfunction or data loss.

1 SNMP SERVER AND SNMP-TRAP SENDING CONFIGURATION

```
snmp-server server
snmp-server community public ro
snmp-server community private rw
snmp-server host 192.168.1.1 traps version 2c private
```

2 SHORT DESIGNATIONS

- **ifIndex —** port index.

May take the following values:

1. Access switches

Switch model	Indexes
MES2300-24 MES2300-24P MES2300B-24 MES2300B-24F MES2300D-24P	- indexes 49-72 — gigabitethernet 1/0/1-24; - indexes 105-108 — tengigabitethernet 1/0/1-4; - indexes 157-180 — gigabitethernet 2/0/1-24; - indexes 213-216 — tengigabitethernet 2/0/1-4; - indexes 265-288 — gigabitethernet 3/0/1-24; - indexes 321-324 — tengigabitethernet 3/0/1-4; - indexes 373-396 — gigabitethernet 4/0/1-24; - indexes 429-432 — tengigabitethernet 4/0/1-4; - indexes 481-504 — gigabitethernet 5/0/1-24; - indexes 537-540 — tengigabitethernet 5/0/1-4; - indexes 589-612 — gigabitethernet 6/0/1-24; - indexes 645-648 — tengigabitethernet 6/0/1-4; - indexes 697-720 — gigabitethernet 7/0/1-24; - indexes 753-756 — tengigabitethernet 7/0/1-4; - indexes 805-828 — gigabitethernet 8/0/1-24; - indexes 861-864 — tengigabitethernet 8/0/1-4; - indexes 1000-1031 — Port-Channel 1-32; - indexes 7000-7063 — loopback 1-64.

2. Aggregation switches

Switch model	Indexes
MES3300-08F MES3300-16F MES3300-24 MES3300-24F MES3300-48 MES3300-48F	- indexes 49-72 — gigabitethernet 1/0/1-24; - indexes 105-108 — tengigabitethernet 1/0/1-4; - indexes 157-180 — gigabitethernet 2/0/1-24; - indexes 213-216 — tengigabitethernet 2/0/1-4; - indexes 265-288 — gigabitethernet 3/0/1-24;

	- indexes 321-324 — tengigabitethernet 3/0/1-4; - indexes 373-396 — gigabitethernet 4/0/1-24; - indexes 429-432 — tengigabitethernet 4/0/1-4; - indexes 481-504 — gigabitethernet 5/0/1-24; - indexes 537-540 — tengigabitethernet 5/0/1-4; - indexes 589-612 — gigabitethernet 6/0/1-24; - indexes 645-648 — tengigabitethernet 6/0/1-4; - indexes 697-720 — gigabitethernet 7/0/1-24; - indexes 753-756 — tengigabitethernet 7/0/1-4; - indexes 805-828 — gigabitethernet 8/0/1-24; - indexes 861-864 — tengigabitethernet 8/0/1-4; - indexes 1000-1031 — Port-Channel 1-32; - indexes 7000-7063 — loopback 1-64.
MES5312 MES5316A MES5324A MES5332A	- indexes 1-32 — tengigabitethernet 1/0/1-32; - indexes 53-84 — tengigabitethernet 2/0/1-32; - indexes 105-136 — tengigabitethernet 3/0/1-32; - indexes 157-188 — tengigabitethernet 4/0/1-32; - indexes 209-240 — tengigabitethernet 5/0/1-32; - indexes 261-292 — tengigabitethernet 6/0/1-32; - indexes 313-344 — tengigabitethernet 7/0/1-32; - indexes 365-396 — tengigabitethernet 8/0/1-32; - indexes 1000-1127 — Port-Channel 1-128; - indexes 7000-7063 — loopback 1-64.

3. Industrial switches

Switch model	Indexes
MES2300DI-28	- indexes 49-76 — gigabitethernet 1/0/1-28; - indexes 157-184 — gigabitethernet 2/0/1-28; - indexes 265-292 — gigabitethernet 3/0/1-28; - indexes 373-400 — gigabitethernet 4/0/1-28; - indexes 481-508 — gigabitethernet 5/0/1-28; - indexes 589-616 — gigabitethernet 6/0/1-28; - indexes 697-724 — gigabitethernet 7/0/1-28; - indexes 805-832 — gigabitethernet 8/0/1-28; - indexes 1000-1031 — Port-Channel 1-32; - indexes 7000-7063 — loopback 1-64.
MES3500I-10P	- indexes 49-60 — gigabitethernet 1/0/1-12; - indexes 1000-1031 — Port-Channel 1-32; - indexes 7000-7063 — loopback 1-64.

4. Data center switches

Switch model	Indexes
MES5310-48 MES5400-24 MES5400-48 MES5410-48	- indexes 1-48 — tengigabitethernet 1/0/1-48; - indexes 49-72 — twentyfivegigabitethernet 1/0/1-24; - indexes 103-108 — hundredgigabitethernet 1/0/1-6; - indexes 109-156 — tengigabitethernet 2/0/1-48; - indexes 157-180 — twentyfivegigabitethernet 2/0/1-24; - indexes 211-216 — hundredgigabitethernet 2/0/1-6; - indexes 217-264 — tengigabitethernet 3/0/1-48; - indexes 265-288 — twentyfivegigabitethernet 3/0/1-24; - indexes 319-324 — hundredgigabitethernet 3/0/1-6; - indexes 325-372 — tengigabitethernet 4/0/1-48; - indexes 373-396 — twentyfivegigabitethernet 4/0/1-24; - indexes 427-432 — hundredgigabitethernet 4/0/1-6; - indexes 433-480 — tengigabitethernet 5/0/1-48; - indexes 481-504 — twentyfivegigabitethernet 5/0/1-24; - indexes 535-540 — hundredgigabitethernet 5/0/1-6; - indexes 541-588 — tengigabitethernet 6/0/1-48; - indexes 589-612 — twentyfivegigabitethernet 6/0/1-24; - indexes 643-648 — hundredgigabitethernet 6/0/1-6; - indexes 649-696 — tengigabitethernet 7/0/1-48; - indexes 697-720 — twentyfivegigabitethernet 7/0/1-24; - indexes 751-756 — hundredgigabitethernet 7/0/1-6; - indexes 757-804 — tengigabitethernet 8/0/1-48; - indexes 805-828 — twentyfivegigabitethernet 8/0/1-24; - indexes 859-864 — hundredgigabitethernet 8/0/1-6; - indexes 1000-1127 — Port-Channel 1-128; - indexes 7000-7063 — loopback 1-64.
MES5500-32	- indexes 1-32 - hundredgigabitethernet 1/0/1-32 - indexes 33-160 - twentyfivegigabitethernet 1/0/1-128 - indexes 161-162 - tengigabitethernet 1/0/1-2 - indexes 163-194 - hundredgigabitethernet 2/0/1-32 - indexes 195-322 - twentyfivegigabitethernet 2/0/1-128 - indexes 323-324 - tengigabitethernet 2/0/1-2 - indexes 325-356 - hundredgigabitethernet 3/0/1-32 - indexes 357-484 - twentyfivegigabitethernet 3/0/1-128 - indexes 485-486 - tengigabitethernet 3/0/1-2 - indexes 487-518 - hundredgigabitethernet 4/0/1-32 - indexes 519-646 - twentyfivegigabitethernet 4/0/1-128

	- indexes 647-648 - tengigabitethernet 4/0/1-2 - indexes 649-680 - hundredgigabitethernet 5/0/1-32 - indexes 681-808 - twentyfivegigabitethernet 5/0/1-128 - indexes 809-810 - tengigabitethernet 5/0/1-2 - indexes 811-842 - hundredgigabitethernet 6/0/1-32 - indexes 843-970 - twentyfivegigabitethernet 6/0/1-128 - indexes 971-972 - tengigabitethernet 6/0/1-2 - indexes 973-1004 - hundredgigabitethernet 7/0/1-32 - indexes 1005-1132 - twentyfivegigabitethernet 7/0/1-128 - indexes 1133-1134 - tengigabitethernet 7/0/1-2 - indexes 1135-1166 - hundredgigabitethernet 8/0/1-32 - indexes 1167-1294 - twentyfivegigabitethernet 8/0/1-128 - indexes 1295-1296 - tengigabitethernet 8/0/1-2 - indexes 3000-3127 - Port-Channel 1-128 - indexes 7000-7063 - loopback 1-64

- **index-of-rule** — rule index in ACL. Always multiple of 20. If the indexes are not divisible by 20 when the rules are created, the sequence numbers of the rules in the ACL will be divisible by 20 after the switch is rebooted;
- **value of field N** — in IP and MAC ACL any rule occupies from one to 3 fields depending on its structure;
- **IP address** — IP address for switch management;

  In the examples given in the document the following IP address is used for management: **192.168.1.30**;
- **ip address of tftp server** — TFTP server IP address;

  In the examples given in the document the following TFTP server IP address is used: **192.168.1.1**;
- **community** — community string (password) for the access via SNMP.

In the examples given in the document the following *community* are used:

private — reading/writing rights (rw);
public — reading only rights (ro).

3 FILE OPERATIONS

3.1 *Saving the configuration*

Saving the configuration to non-volatile memory

MIB: rlcopy.mib

Tables used: rlCopyEntry — 1.3.6.1.4.1.89.87.2.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.87.2.1.3.1 i {local(1)} \
 1.3.6.1.4.1.89.87.2.1.7.1 i {runningConfig(2)} \
 1.3.6.1.4.1.89.87.2.1.8.1 i {local(1)} \
 1.3.6.1.4.1.89.87.2.1.12.1 i {startupConfig (3)} \
 1.3.6.1.4.1.89.87.2.1.17.1 i {createAndGo (4)}
```

Example

CLI command:

```
copy running-config startup-config
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.87.2.1.3.1 i 1 \
1.3.6.1.4.1.89.87.2.1.7.1 i 2 \
1.3.6.1.4.1.89.87.2.1.8.1 i 1 \
1.3.6.1.4.1.89.87.2.1.12.1 i 3 \
1.3.6.1.4.1.89.87.2.1.17.1 i 4
```

Saving configuration to volatile memory from non-volatile one

MIB: rlcopy.mib

Tables used: rlCopyEntry — 1.3.6.1.4.1.89.87.2.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.87.2.1.3.1 i {local(1)} \
 1.3.6.1.4.1.89.87.2.1.7.1 i {startupConfig (3)} \
 1.3.6.1.4.1.89.87.2.1.8.1 i {local(1)} \
 1.3.6.1.4.1.89.87.2.1.12.1 i {runningConfig(2)} \
 1.3.6.1.4.1.89.87.2.1.17.1 i {createAndGo (4)}
```

Example

CLI command:

```
copy startup-config running-config
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.87.2.1.3.1 i 1 \
1.3.6.1.4.1.89.87.2.1.7.1 i 3 \
1.3.6.1.4.1.89.87.2.1.8.1 i 1 \
1.3.6.1.4.1.89.87.2.1.12.1 i 2 \
1.3.6.1.4.1.89.87.2.1.17.1 i 4
```

Removing the configuration from the non-volatile memory

MIB: RADLAN-rndMng

Tables used: rndAction — 1.3.6.1.4.89.1.2

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.1.2.0 i {eraseStartupCDB (20)}
```

Startup-config deletion example

CLI command:
```
delete startup-config
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.1.2.0 i 20
```

3.2 *Operation with TFTP server*

Copying the configuration from the volatile memory to TFTP server

MIB: RADLAN-COPY-MIB

Tables used: rlCopyEntry — 1.3.6.1.4.1.89.87.2.1

```
snmpset -v2c -c <community> -t 5 -r 3 <IP address> \
 1.3.6.1.4.1.89.87.2.1.3.1 i {local(1)} \
 1.3.6.1.4.1.89.87.2.1.7.1 i {runningConfig(2)} \
 1.3.6.1.4.1.89.87.2.1.8.1 i {tftp(3)} \
 1.3.6.1.4.1.89.87.2.1.9.1 a {ip address of tftp server} \
 1.3.6.1.4.1.89.87.2.1.11.1 s "MES-config.cfg" \
 1.3.6.1.4.1.89.87.2.1.17.1 i {createAndGo (4)} \
 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i {number of vrf (default 1)}
```

Command 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i {number of vrf (default 1)} is applied when VRF other than the default one is used.

Example of copying from running-config to TFTP server

CLI command:
```
copy running-config tftp://192.168.1.1/MES-config.cfg [vrf test]
```

SNMP command:
```
snmpset -v2c -c private -t 5 -r 3 192.168.1.30 \
1.3.6.1.4.1.89.87.2.1.3.1 i 1 \
1.3.6.1.4.1.89.87.2.1.7.1 i 2 \
1.3.6.1.4.1.89.87.2.1.8.1 i 3 \
1.3.6.1.4.1.89.87.2.1.9.1 a 192.168.1.1 \
1.3.6.1.4.1.89.87.2.1.11.1 s "conf.cfg" \
1.3.6.1.4.1.89.87.2.1.17.1 i 4 \
1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i 2
```

Command 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i 2 is applied when VRF other than the default one vrf test is used.

Copying the configuration to the non-volatile memory from TFTP server

MIB: rlcopy.mib

Tables used: rlCopyEntry — 1.3.6.1.4.1.89.87.2.1

```
snmpset -v2c -c <community> -t 5 -r 3 <IP address> \
 1.3.6.1.4.1.89.87.2.1.3.1 i {tftp(3)} \
 1.3.6.1.4.1.89.87.2.1.4.1 a {ip address of tftp server} \
 1.3.6.1.4.1.89.87.2.1.6.1 s "MES-config.cfg" \
 1.3.6.1.4.1.89.87.2.1.8.1 i {local(1)} \
 1.3.6.1.4.1.89.87.2.1.12.1 i {runningConfig(2)} \
```

```
1.3.6.1.4.1.89.87.2.1.17.1 i {createAndGo (4)} \
1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i {number of vrf (default 1)}
```

Command 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i {number of vrf (default 1)} is applied when VRF other than the default one is used.

Example of copying from a TFTP server to running-config

CLI command:

```
copy tftp://192.168.1.1/MES-config.cfg running-config [vrf test]
```

SNMP command:

```
snmpset -v2c -c private -t 5 -r 3 192.168.1.30 \
1.3.6.1.4.1.89.87.2.1.3.1 i 3 \
1.3.6.1.4.1.89.87.2.1.4.1 a 192.168.1.1 \
1.3.6.1.4.1.89.87.2.1.6.1 s "conf.cfg" \
1.3.6.1.4.1.89.87.2.1.8.1 i 1 \
1.3.6.1.4.1.89.87.2.1.12.1 i 2 \
1.3.6.1.4.1.89.87.2.1.17.1 i 4 \
1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i 2
```

Command 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i 2 is applied when VRF other than the default one vrf test is used.

Copying the configuration from the non-volatile memory to TFTP server

MIB: file rlcopy.mib

Tables used: rlCopyEntry — 1.3.6.1.4.1.89.87.2.1

```
snmpset -v2c -c <community> -t 5 -r 3 <IP address> \
 1.3.6.1.4.1.89.87.2.1.3.1 i {local(1)} \
 1.3.6.1.4.1.89.87.2.1.7.1 i {startupConfig (3)} \
 1.3.6.1.4.1.89.87.2.1.8.1 i {tftp(3)} \
 1.3.6.1.4.1.89.87.2.1.9.1 a {ip address of tftp server} \
 1.3.6.1.4.1.89.87.2.1.11.1 s "MES-config.cfg" \
 1.3.6.1.4.1.89.87.2.1.17.1 i {createAndGo (4)}\
 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i {number of vrf (default 1)}
```

Command 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i {number of vrf (default 1)} is applied when VRF other than the default one is used.

Example of copying from startup-config to TFTP server

CLI command:

```
boot config tftp://192.168.1.1/MES-config.cfg [vrf test]
```

SNMP command:

```
snmpset -v2c -c private -t 5 -r 3 192.168.1.30 \
1.3.6.1.4.1.89.87.2.1.3.1 i 1 \
1.3.6.1.4.1.89.87.2.1.7.1 i 2 \
1.3.6.1.4.1.89.87.2.1.8.1 i 3 \
1.3.6.1.4.1.89.87.2.1.9.1 a 192.168.1.1 \
1.3.6.1.4.1.89.87.2.1.11.1 s "conf.cfg" \
1.3.6.1.4.1.89.87.2.1.17.1 i 4 \
1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i 2
```

Command 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i 2 is applied when VRF other than the default one vrf test is used.

Copying the configuration to the non-volatile memory from TFTP server

MIB: RADLAN-COPY-MIB

Tables used: rlCopyEntry — 1.3.6.1.4.1.89.87.2.1

```
snmpset -v2c -c <community> -t 5 -r 3 <IP address> \
 1.3.6.1.4.1.89.87.2.1.3.1 i {tftp(3)} \
 1.3.6.1.4.1.89.87.2.1.4.1 a {ip address of tftp server} \
 1.3.6.1.4.1.89.87.2.1.6.1 s "MES-config.cfg" \
 1.3.6.1.4.1.89.87.2.1.8.1   i {local(1)} \
 1.3.6.1.4.1.89.87.2.1.12.1 i {startupConfig (3)} \
 1.3.6.1.4.1.89.87.2.1.17.1 i {createAndGo (4)}
```

Example of copying startup-config from TFTP server

CLI command:
```
boot config tftp://192.168.1.1/MES-config.cfg
```

SNMP command:
```
snmpset -v2c -c private -t 5 -r 3 192.168.1.30 \
1.3.6.1.4.1.89.87.2.1.3.1 i 3 \
1.3.6.1.4.1.89.87.2.1.4.1 a 192.168.1.1 \
1.3.6.1.4.1.89.87.2.1.6.1 s "conf.cfg" \
1.3.6.1.4.1.89.87.2.1.8.1 i 1  \
1.3.6.1.4.1.89.87.2.1.12.1 i 3 \
1.3.6.1.4.1.89.87.2.1.17.1 i 4
```

3.3 *Switch autoconfiguration*

Enabling DHCP-based autoconfiguration (enabled by default)

MIB: radlan-dhcpcl-mib.mib

Tables used: rlDhcpClOption67Enable — 1.3.6.1.4.1.89.76.9

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.76.9.0 i {enable(1), disable(2)}
```

Example

CLI command:
```
boot host auto-config
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.76.9.0 i 1
```

3.4 *Firmware update*

Switch firmware update

Performed in two steps:

1. Firmware image upload

MIB: RADLAN-COPY-MIB

Tables used: rlCopyEntry — 1.3.6.1.4.1.89.87.2.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.87.2.1.3.1 i {tftp (3)} \
1.3.6.1.4.1.89.87.2.1.4.1 a {ip add of tftp server} \
1.3.6.1.4.1.89.87.2.1.6.1 s "image name" \
1.3.6.1.4.1.89.87.2.1.8.1 i {local(1)} \
1.3.6.1.4.1.89.87.2.1.12.1 i {image(8)} \
1.3.6.1.4.1.89.87.2.1.17.1 i {createAndGo(4)}\
1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i {number of vrf (default 1)}
```

Command 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i {number of vrf (default 1)} when VRF other than the default one is used.

Example

CLI command:
```
boot system tftp://192.168.1.1/mes5300a-611-R2.ros [vrf test]
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.87.2.1.3.1 i 3 \
1.3.6.1.4.1.89.87.2.1.4.1 a 192.168.1.1 \
1.3.6.1.4.1.89.87.2.1.6.1 s "mes5300a-611-R2.ros" \
1.3.6.1.4.1.89.87.2.1.8.1 i 1 1.3.6.1.4.1.89.87.2.1.12.1 i 8 \
1.3.6.1.4.1.89.87.2.1.17.1 i 4 \
1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i 2
```

Command 1.3.6.1.4.1.35265.1.23.3.11.1.1.1 i 2 when VRF other than the default one vrf test is used.

2. Active switch image change

MIB: RADLAN-DEVICEPARAMS-MIB

Tables used: rndActiveSoftwareFileAfterReset — 1.3.6.1.4.1.89.2.13.1.1.3

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.2.13.1.1.3.1 i {image1 (1), image2 (2)}
```

Example

CLI command:
```
boot system inactive-image
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.2.13.1.1.3.1 i 1
```

The command is applied automatically after the firmware is downloaded from the server.

Switch reboot

MIB: rlmng.mib

Tables used: rlRebootDelay — 1.3.6.1.4.1.89.1.10

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.1.10.0 t {time delay before rebooting}
```

Example of a reboot delayed by 8 minutes

CLI command:
```
reload in 8
```
SNMP command:
```
snmpset -v2c -c private -r 0 192.168.1.30 \
1.3.6.1.4.1.89.1.10.0 t 48000
```

To reboot immediately, the value t=0 is required.

Viewing the firmware image

MIB: RADLAN-DEVICEPARAMS-MIB.mib

Tables used: rndActiveSoftwareFile — 1.3.6.1.4.1.89.2.13.1.1.2

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.2.13.1.1.2
```

Example

CLI command:
```
show bootvar
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.2.13.1.1.2
```

1) Possible options:

image1(1)
image2(2)

2) You can view the active software image after reboot in rndActiveSoftwareFileAfterReset — 1.3.6.1.4.1.89.2.13.1.1.3.

Viewing uploaded firmware images

MIB: RADLAN-DEVICEPARAMS-MIB.mib

Tables used: rndImageInfoTable — 1.3.6.1.4.1.89.2.16.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.2.16.1
```

Example

CLI command:
```
show bootvar
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.2.16.1
```

Viewing the current switch firmware version

MIB: RADLAN-DEVICEPARAMS-MIB.mib

Tables used: rndBrgVersion — 1.3.6.1.4.1.89.2.4

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.2.4
```

Example

CLI command:
```
show version
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.2.4
```

Viewing the current switch firmware version

MIB: RADLAN-DEVICEPARAMS-MIB.mib

Tables used: genGroupHWVersion — 1.3.6.1.4.1.89.2.11.1

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.2.11.1
```

Example

CLI command:
```
show system id
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.2.11.1
```

4 SYSTEM MANAGEMENT

4.1 *System resources*

View switch serial number

MIB: rlphysdescription.mib

Tables used: rlPhdUnitGenParamSerialNum — 1.3.6.1.4.1.89.53.14.1.5

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.53.14.1.5
```

Example

CLI command:
```
show system id
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.53.14.1.5
```

Viewing information on tcam load

MIB: RADLAN-QOS-CLI-MIB

Tables used: rlQosClassifierUtilizationPercent — 1.3.6.1.4.1.89.88.36.1.1.2

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.36.1.1.2
```

Example

CLI command:
```
show system tcam utilization
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.88.36.1.1.2
```

Viewing the maximum number of hosts

MIB: rltuning.mib

Tables used: rsMaxIpSFftEntries — 1.3.6.1.4.1.89.29.8.9.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.29.8.9.1
```

Example

CLI command:
```
show system router resources
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.29.8.9.1
```

Viewing the number of hosts used

MIB: rlfft.mib

Tables used: rlSysmngTcamAllocInUseEntries — 1.3.6.1.4.1.89.204.1.1.1.5

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.204.1.1.1.5.5.116.99.97.109.49.1
```

Example

CLI command:
```
show system router resources
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.204.1.1.1.5.5.116.99.97.109.49.1
```

Viewing the maximum number of routes

MIB: rltuning.mib

Tables used: rsMaxIpPrefixes — 1.3.6.1.4.1.89.29.8.21.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.29.8.21.1
```

Example

CLI command:
```
show system router resources
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.29.8.21.1
```

Viewing the number of routes used

MIB: rlip.mib

Tables used: rlIpTotalPrefixesNumber — 1.3.6.1.4.1.89.26.25

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.26.25
```

Example

CLI command:
```
show system router resources
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.26.25
```

Viewing the maximum number of IP interfaces

MIB: rltuning.mib

Tables used: rsMaxIpInterfaces — 1.3.6.1.4.1.89.29.8.25.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.29.8.25.1
```

Example

CLI command:
```
show system router resources
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.29.8.25.1
```

Viewing the number of IP interfaces used

MIB: rlip.mib

Tables used: rlIpAddressesNumber — 1.3.6.1.4.1.89.26.23

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.26.23
```

Example

CLI command:
```
show system router resources
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.26.23
```

Viewing the system MAC address of the switch

MIB: rlphysdescription.mib

Tables used: rlPhdStackMacAddr — 1.3.6.1.4.1.89.53.4.1.7

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.53.4.1.7
```

Example

CLI command:
```
show system
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.53.4.1.7
```

Viewing switch Uptime

The command is used for Uptime less than 497 days.

MIB: SNMPv2-MIB

Tables used: sysUpTime — 1.3.6.1.2.1.1.3

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.1.3
```

Example of viewing the switch Uptime

CLI command:
```
show system
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.1.3
```

Viewing the switch Uptime counter in seconds

MIB: eltSystemGlobal.mib

Tables used: eltSysUpTimeInSec - .1.3.6.1.4.1.35265.1.23.1.10.3.1.1

```
snmpget -v2c -c <community> <IP address> \
.1.3.6.1.4.1.35265.1.23.1.10.3.1.1.0
```

Viewing the switch Uptime counter in seconds

SNMP command:
```
snmpget -v2c -c private 192.168.1.30 \
.1.3.6.1.4.1.35265.1.23.1.10.3.1.1.0
```

Viewing the switch Uptime counter in seconds for a unit on the stack

MIB: ELTEX-PHYSICAL-DESCRIPTION-MIB.mib

Tables used: eltPhdUnitEnvParamTable - .1.3.6.1.4.1.35265.1.23.53.7

```
snmpget -v2c -c <community> <IP address> \
.1.3.6.1.4.1.35265.1.23.53.7.1.1.{unit}
```

Viewing the switch Uptime counter in seconds for a unit on the stack

SNMP command:
```
snmpget -v2c -c private 192.168.1.30 \
.1.3.6.1.4.1.35265.1.23.53.7.1.1.1
```

Viewing Uptime port

MIB: SNMPv2-MIB, IF-MIB

Tables used:
sysUpTime — 1.3.6.1.2.1.1.3
ifLastChange — 1.3.6.1.2.1.2.2.1.9

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.1.3
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.2.2.1.9.{ifindex}
```

Viewing Uptime TenGigabitethernet 1/0/23 port

CLI command:
```
show interface status TenGigabitethernet 1/0/23
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.1.3
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.9.23
```

The output of the first command must be removed from the output of the second command. The resulting value will be the uptime of the port.

Enabling CPU traffic monitoring service

MIB: rlsct.mib

Tables used: rlSctCpuRateEnabled — 1.3.6.1.4.1.89.203.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.203.1.0 i {true(1), false(2)}
```

Example

CLI command:
```
service cpu-input-rate
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.203.1.0 i 1
```

Viewing the counters and the number of packets processed by CPU per second (by traffic type)

MIB: rlsct.mib

Tables used: eltCpuRateStatisticsTable — 1.3.6.1.4.1.35265.1.23.1.773.1.2.1

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.35265.1.23.1.773.1.2.1.1.{rate in pps(2), packets count(3)}
```

Example of viewing the number of packets processed by CPU per second

CLI command:
```
show cpu input-rate detailed
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.773.1.2.1.1.2
```

Assigning indexes to traffic types:

stack(1)
http(2)
telnet(3)
ssh(4)
snmp(5)
ip(6)
arp(7)
arpInspec(8)
stp(9)
ieee(10)
routeUnknown(11)
ipHopByHop(12)
mtuExceeded(13)
ipv4Multicast(14)
ipv6Multicast(15)
dhcpSnooping(16)
igmpSnooping(17)
mldSnooping(18)
ttlExceeded(19)
ipv4IllegalAddress(20)
ipv4HeaderError(21)
ipDaMismatch(22)
sflow(23)
logDenyAces(24)
dhcpv6Snooping(25)
vrrp(26)
logPermitAces(27)
ipv6HeaderError (28)

Changing CPU limits

MIB: eltSwitchRateLimiterMIB.mib

Tables used: eltCPURateLimiterTable — 1.3.6.1.4.1.35265.1.23.1.773.1.1.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.1.773.1.1.1.1.2.{index} i {limiter value}
```

Example of setting a 512 pps limit for SNMP traffic to CPU

CLI command:
```
service cpu-rate-limits snmp 512
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.773.1.1.1.1.2.4 i 512
```

Index list:

eltCPURLTypeHttp(1)
eltCPURLTypeTelnet(2)
eltCPURLTypeSsh(3)
eltCPURLTypeSnmp(4)

eltCPURLTypeIp(5)
eltCPURLTypeLinkLocal(6)
eltCPURLTypeArpRouter(7)
eltCPURLTypeArpInspec(9)
eltCPURLTypeStpBpdu(10)
eltCPURLTypeOtherBpdu(11)
eltCPURLTypeIpRouting(12)
eltCPURLTypeIpOptions(13)
eltCPURLTypeDhcpSnoop(14)
eltCPURLTypeIgmpSnoop(16)
eltCPURLTypeMldSnoop(17)
eltCPURLTypeSflow(18)
eltCPURLTypeLogDenyAces(19)
eltCPURLTypeIpErrors(20)
eltCPURLTypeOther(22)

CPU load monitoring

MIB: rlmng.mib

Tables used:
rlCpuUtilDuringLastSecond — 1.3.6.1.4.1.89.1.7
rlCpuUtilDuringLastMinute — 1.3.6.1.4.1.89.1.8
rlCpuUtilDuringLastMinute — 1.3.6.1.4.1.89.1.8

- Download in the last five seconds: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.1.7;
- Load in 1 minute: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.1.8;
- Load in 5 minutes: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.1.9.

Example of viewing CPU utilization for the last 5 seconds

CLI command:
```
show cpu utilization
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.1.7
```

Enable CPU utilization monitoring by tasks

MIB: RADLAN-rndMng

Tables used: rlCpuTasksUtilEnable — 1.3.6.1.4.1.89.1.6

```
snmpset -v2c -c <community> <IP address>
1.3.6.1.4.1.89.1.6.0 i {true(1), false(2)}
```

Example

CLI command:
```
service tasks-utilization
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.1.6.0 i 1
```

CPU utilization monitoring by tasks

MIB: ELTEX-MES-MNG-MIB

Tables used:
eltCpuTasksUtilStatisticsUtilizationDuringLast5Seconds — 1.3.6.1.4.1.35265.1.23.1.9.1.2.1.1.3
eltCpuTasksUtilStatisticsUtilizationDuringLastMinute — 1.3.6.1.4.1.35265.1.23.1.9.1.2.1.1.4
eltCpuTasksUtilStatisticsUtilizationDuringLast5Minutes — 1.3.6.1.4.1.35265.1.23.1.9.1.2.1.1.5

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.1.9.1.2.1.1.3.{5sec(3), 1min(4), 5min(5)}.{task index}
```

Example of viewing CPU utilization by tasks for the last 5 seconds

CLI command:
```
show tasks utilization
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.9.1.2.1.1.3
```

Assigning indexes to tasks:

LTMR(0)	**NTST(50)**	**IPRD(100)**
ROOT(1)	**CNLD(51)**	**PNGA(101)**
IT33(2)	**HOST(52)**	**UDPR(102)**
IV11(3)	**TBI_(53)**	**VRRP(103)**
URGN(4)	**BRMN(54)**	**TRCE(104)**
TMNG(5)	**COPY(55)**	**SSLP(105)**
IOTG(6)	**TRNS(56)**	**WBSO(106)**
IOUR(7)	**MROR(57)**	**WBSR(107)**
IOTM(8)	**DFST(58)**	**GOAH(108)**
SSHU(9)	**SFTR(59)**	**ECHO(109)**
XMOD(10)	**SFMG(60)**	**TNSR(110)**
MSCm(11)	**HCPT(61)**	**TNSL(111)**
STSA(12)	**EVAU(62)**	**SSHP(112)**
STSB(13)	**EVFB(63)**	**PTPT(113)**
STSC(14)	**EVRT(64)**	**NBBT(114)**
STSD(15)	**EPOE(65)**	**SQIN(115)**
STSE(16)	**DSPT(66)**	**MUXT(116)**
CPUT(17)	**B_RS(67)**	**DMNG(117)**
EVAP(18)	**TRIG(68)**	**DSYN(118)**
HCLT(19)	**MACT(69)**	**HSEU(119)**
EVLC(20)	**SW2M(70)**	**DTSA(120)**
SELC(21)	**3SWQ(71)**	**SS2M(121)**
SEAU(22)	**POLI(72)**	**DSND(122)**
ESTC(23)	**OBSR(73)**	**STMB(123)**
SSTC(24)	**NTPL(74)**	**AAAT(124)**
BOXS(25)	**L2HU(75)**	**AATT(125)**
BSNC(26)	**L2PS(76)**	**SCPT(126)**
BOXM(27)	**SFSM(77)**	**DH6C(127)**
TRMT(28)	**NSCT(78)**	**RCLA(128)**
D_SP(29)	**NSFP(79)**	**RCLB(129)**
D_LM(30)	**NVCT(80)**	**RCDS(130)**
PLCT(31)	**NACT(81)**	**GRN_(131)**
PLCR(32)	**NSTM(82)**	**IPMT(132)**

exRX(33)	**NINP(83)**	**SNTP(133)**
3SWF(34)	**L2UT(84)**	**DHCP(134)**
MSRP(35)	**BRGS(85)**	**DHCp(135)**
HSES(36)	**FHSS(86)**	**RELY(136)**
HSCS(37)	**FHSF(87)**	**MSSS(137)**
MRDP(38)	**FFTT(88)**	**WBAM(138)**
MLDP(39)	**IPAT(89)**	**WNTT(139)**
SETX(40)	**IP6M(90)**	**RADS(140)**
EVTX(41)	**IP6L(91)**	**SNAS(141)**
SERX(42)	**IP6C(92)**	**SNAE(142)**
EVRX(43)	**IP6R(93)**	**SNAD(143)**
HLTX(44)	**RPTS(94)**	**MNGT(144)**
LBDR(45)	**ARPG(95)**	**UTST(145)**
DDFG(46)	**IPG_(96)**	**SOCK(146)**
SYLG(47)	**DNSC(97)**	**TCPP(147)**
CDB_(48)	**ICMP(98)**	**UNQt(148)**
SNMP(49)	**TFTP(99)**	

Viewing the total amount of RAM

MIB: ELTEX-PROCESS-MIB.mib

Tables used: eltexProcessMemoryEntry - 1.3.6.1.4.1.35265.41.1.2.1.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.41.1.2.1.1.{For MES5312, MES5316A, MES5324A, MES5332A — (3),
for MES5400-24, MES5400-48, MES5500-32 — (5)}.0
```

Example

CLI command:
```
show cpu utilization
```

SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.35265.41.1.2.1.1.3.0
```

Viewing the free amount of RAM

MIB: ELTEX-PROCESS-MIB.mib

Tables used: eltexProcessMemoryEntry - 1.3.6.1.4.1.35265.41.1.2.1.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.41.1.2.1.1.{For MES5312, MES5316A, MES5324A, MES5332A — (7),
for MES5400-24, MES5400-48, MES5500-32 — (9)}.0
```

Example

CLI command:
```
show cpu utilization
```

SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.35265.41.1.2.1.1.7.0
```

Enabling jumbo-frames support

MIB: radlan-jumboframes-mib.mib

Tables used: rlJumboFrames — 1.3.6.1.4.1.89.91

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.91.2.0 i {enabled(1), disabled(2)}
```

Example

CLI command:
```
port jumbo-frame
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.91.2.0 i 1
```

4.2 *System parameters*

Power supply units state control

MIB: rlphysdescription.mib

Tables used: rlPhdUnitEnvParamTable — 1.3.6.1.4.1.89.53.15

- The primary power supply unit: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.2;
- The redundant power supply unit: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.3.

Example of viewing the primary power supply unit state

CLI command:
```
show system
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.53.15.1.2
```

1) The primary power supply unit has the following states:

normal (1)
warning (2)
critical (3)
shutdown (4)
notPresent (5)
notFunctioning (6)

2) The redundant power supply unit has the following states:

normal (1)
warning (2)
critical (3)
shutdown (4)
notPresent (5)
notFunctioning (6)

Fans state control

MIB: rlphysdescription.mib

Tables used: rlPhdUnitEnvParamTable — 1.3.6.1.4.1.89.53.15

- Fan 1: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.4
- Fan 2: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.5
- Fan 3: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.6
- Fan 4: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.7

Example of viewing the status of MES5332A fan 3

CLI command:
```
show system
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.53.15.1.6
```

The following states are possible:

normal (1)
notFunctioning (5)

Monitoring of temperature sensor readings

MIB: RADLAN-MIB

Tables used: rlEnv — 1.3.6.1.4.1.89.83.2.1.1.1.4

Temperature sensor 1: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.83.2.1.1.1.4

Example of viewing the sensor temperature

CLI command:
```
show system sensors
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.83.2.1.1.1.4
```

Monitoring of temperature sensors state:

MIB: rlphysdescription.mib

Tables used: rlPhdUnitEnvParamTable — 1.3.6.1.4.1.89.53.15

Temperature sensor 1: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.11

Example

CLI command:
```
show system sensors
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.53.15.1.11
```

4.3 *Stack parameters*

Stack parameters monitoring

MIB: rlphysdescription.mib

Tables used: rlPhdStackTable — 1.3.6.1.4.1.89.53.4

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.53.4
```

Example of viewing stack parameters

CLI command:
```
show stack
```

SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.89.53.4
```

Stack ports monitoring

MIB: rlphysdescription.mib

Tables used: rlCascadeTable — 1.3.6.1.4.1.89.53.23

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.53.23
```

Example of viewing stack ports status

CLI command:
```
show stack links
```

SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.89.53.23
```

4.4 *Device management*

Setting/changing the hostname on the device

MIB: SNMPv2-MIB

Tables used: sysName — 1.3.6.1.2.1.1.5

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.2.1.1.5.0 s "{hostname}"
```

Example of "mes5332A" hostname assignment

CLI command:
```
hostname mes2324
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.1.5.0 s "mes5332A"
```

Enabling/disabling management acl

MIB: RADLAN-MNGINF-MIB

Tables used:
rlMngInfEnable — 1.3.6.1.4.1.89.89.2
rlMngInfActiveListName — 1.3.6.1.4.1.89.89.3

```
snmpset -v2c -c <community> <IP address>
1.3.6.1.4.1.89.89.2.0 i {true(1), false(2)}\
1.3.6.1.4.1.89.89.3.0 s {name}do ping
```

Example of "eltex" management acl enabling

CLI command:
```
management access-class eltex
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.89.2.0 i 1 \
1.3.6.1.4.1.89.89.3.0 s eltex
```

Using the ping utility

MIB: rlapplication.mib

Tables used: rsPingInetTable — 1.3.6.1.4.1.89.35.4.2

```
snmpset -v2c -c <community> <IP address>\

1.3.6.1.4.1.89.35.4.1.1.2.{IP address>} i {Packet count}\
1.3.6.1.4.1.89.35.4.1.1.3.{IP address>} i {Packet Size}\
1.3.6.1.4.1.89.35.4.1.1.4.{IP address>} i {Packet Timeout}\
1.3.6.1.4.1.89.35.4.1.1.5.{IP address>} i {Ping Delay}\
1.3.6.1.4.1.89.35.4.1.1.6.{IP address>} i {Send SNMP Trap(2)}\
1.3.6.1.4.1.89.35.4.1.1.14.{IP   address>}   i   {createAndGo(4),   destroy(6),
active(1)}
```

Example of a 192.168.1.1 node ping

CLI command:
```
ping 192.168.1.1 count 10 size 250 timeout 1000
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.35.4.1.1.2.192.168.1.1 i 10 \
1.3.6.1.4.1.89.35.4.1.1.3.192.168.1.1 i 250 \
1.3.6.1.4.1.89.35.4.1.1.4.192.168.1.1 i 1000 \
1.3.6.1.4.1.89.35.4.1.1.5.192.168.1.1 i 0 \
1.3.6.1.4.1.89.35.4.1.1.6.192.168.1.1 i 2 \
1.3.6.1.4.1.89.35.4.1.1.14.192.168.1.1 i 4
```

When 4 (createAndGo) is set to the rsPingEntryStatus field, a ping operation is created and enabled.
To re-ping a remoted host, set the value 1 (active) in the rsPingEntryStatus field.
After the operation is completed, delete all the entries by setting the value 6 (destroy) in the rsPingEntryStatus field. Otherwise, it will be impossible to ping another host via CLI and SNMP.

Example of removal:
snmpset -v2c -c private 192.168.1.30
1.3.6.1.4.1.89.35.4.1.1.2.192.168.1.1 i 10
1.3.6.1.4.1.89.35.4.1.1.3.192.168.1.1 i 250
1.3.6.1.4.1.89.35.4.1.1.4.192.168.1.1 i 1000
1.3.6.1.4.1.89.35.4.1.1.5.192.168.1.1 i 0
1.3.6.1.4.1.89.35.4.1.1.6.192.168.1.1 i 2
1.3.6.1.4.1.89.35.4.1.1.14.192.168.1.1 i 6

Ping utility monitoring

MIB: rlapplication.mib

Tables used: rsPingEntry — 1.3.6.1.4.1.89.35.4.1.1

```
snmpwalk -v2c -c <community> <IP address>\

1.3.6.1.4.1.89.35.4.1.1.{Number of packets sent(7), Number of packets
received(8), Minimum response time(9), Average response time(10), Maximum
response time(11)}
```

Example of viewing the number of packets received

CLI command:
```
ping 192.168.1.1
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.35.4.1.1.8
```

When the value 6 (destroy) is set in the rsPingEntryStatus field, monitoring will be forbidden until a new operation is created.

System log configuration

MIB: DRAFT-IETF-SYSLOG-DEVICE-MIB

Tables used: snmpSyslogCollectorEntry – 1.3.6.1.4.1.89.82.1.2.4.1

```
snmpset -v2c -c <community> -t 10 -r 5 <IP address> \
1.3.6.1.4.1.89.82.1.2.4.1.2.1 s "{name}" \
1.3.6.1.4.1.89.82.1.2.4.1.3.1 i  {ipv4(1), ipv6(2)} \
1.3.6.1.4.1.89.82.1.2.4.1.4.1 x {ip add in HEX} \
1.3.6.1.4.1.89.82.1.2.4.1.5.1 u {udp port number} \
1.3.6.1.4.1.89.82.1.2.4.1.6.1 i {syslog facility(16-24)} \
1.3.6.1.4.1.89.82.1.2.4.1.7.1 i {severity level} \
1.3.6.1.4.1.89.82.1.2.4.1.9.1 i {createAndGo(4), destroy(6)}
```

Example of adding a server for logging

CLI command:

```
logging host 192.168.1.1 description 11111
```

SNMP command:

```
snmpset -v2c -c private -t 10 -r 5 192.168.1.30 \
1.3.6.1.4.1.89.82.1.2.4.1.2.1 s "11111" \
1.3.6.1.4.1.89.82.1.2.4.1.3.1 i 1 \
1.3.6.1.4.1.89.82.1.2.4.1.4.1 x C0A80101 \
1.3.6.1.4.1.89.82.1.2.4.1.5.1 u 514 \
1.3.6.1.4.1.89.82.1.2.4.1.6.1 i 23 \
1.3.6.1.4.1.89.82.1.2.4.1.7.1 i 6 \
1.3.6.1.4.1.89.82.1.2.4.1.9.1 i 4
```

Severity level is specified as follows:

emergency(0),
alert(1),
critical(2),
error(3),
warning(4),
notice(5),
info(6),
debug(7)

Facility:

local0(16),
local1(17),
local2(18),
local3(19),
local4(20),
local5(21),
local6(22),
local7(23),
no-map(24)

5 SYSTEM TIME CONFIGURATION

SNTP server address configuration

MIB: rlsntp.mib

Tables used: rlSntpConfigServerInetTable — 1.3.6.1.4.1.89.92.2.2.17

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.92.2.2.17.1.3.1.4.{ip address in DEC. IP address bytes are
separated by points} i {true(1), false(2). Set poll value} \
 1.3.6.1.4.1.89.92.2.2.17.1.9.1.4.{ip address in DEC. IP address bytes are
separated by points} u 0 \
 1.3.6.1.4.1.89.92.2.2.17.1.10.1.4.{ip address in DEC. IP address bytes are
separated by points} i {createAndGo(4), destroy(6)}
```

Example of specifying SNTP server with IP address 91.226.136.136

CLI command:

```
sntp server 91.226.136.136 poll
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.92.2.2.17.1.3.1.4.91.226.136.136 i 1 \
1.3.6.1.4.1.89.92.2.2.17.1.9.1.4.91.226.136.136 u 0 \
1.3.6.1.4.1.89.92.2.2.17.1.10.1.4.91.226.136.136 i 4
```

Setting the polling time for SNTP client

MIB: rlsntp.mib

Tables used: rlSntpNtpConfig — 1.3.6.1.4.1.89.92.2.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.92.2.1.4.0 i {range 60-86400}
```

Example of setting the polling time of 60 seconds

CLI command:

```
sntp client poll timer 60
```

SNMP command:

```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.92.2.1.4.0 i 60
```

To return to default settings, set the time of 1024 seconds.

Setting the operation of unicast SNTP clients

MIB: rlsntp.mib

Tables used: rlSntpConfig — 1.3.6.1.4.1.89.92.2.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.92.2.2.5.0 i {true(1), false(2)}
```

Example of a unicast SNTP server polling

CLI command:
```
sntp unicast client poll
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.92.2.2.5.0 i 1
```

Adding a time zone

MIB: rlsntp.mib

Tables used: rlTimeSyncMethodMode — 1.3.6.1.4.1.89.92.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.92.1.6.0 s "{TimeZone}" \
1.3.6.1.4.1.89.92.1.7.0 s "{NameZone}"
```

Example of adding a time zone on a device

CLI command:
```
clock timezone test +7
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.92.1.6.0 s "+7:00" \
1.3.6.1.4.1.89.92.1.7.0 s "test"
```

6 INTERFACE CONFIGURATION

6.1 *Ethernet interface parameters*

Viewing Description port

MIB: IF-MIB or eltMng.mib

Tables used: ifAlias — 1.3.6.1.2.1.31.1.1.1.18 or iflongDescr — 1.3.6.1.4.1.35265.1.23.1.1.31.1.1.1.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.31.1.1.1.18.{ifIndex}

snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.1.1.31.1.1.1.1.{ifIndex}
```

Example of viewing Description on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interfaces description TenGigabitEthernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.31.1.1.1.18.23
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.1.31.1.1.1.1.23
```

Viewing Description vlan

MIB: Q-BRIDGE-MIB

Tables used: dot1qVlanStaticTable — 1.3.6.1.2.1.17.7.1.4.3

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.17.7.1.4.3.1.1.{vlan id}
```

Example of viewing Description vlan 100

CLI command:
```
show interfaces description vlan 100
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.17.7.1.4.3.1.1.1
```

Viewing speed on the interface

MIB: IF-MIB

Tables used: ifHighSpeed — 1.3.6.1.2.1.31.1.1.1.15

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.31.1.1.1.15.{ififndex}
```

Example of disabling negotiation on the TenGigabitethernet 1/0/23

CLI command:
```
show interface status TenGigabitethernet 1/0/23
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.31.1.1.1.15.23
```

Enabling/disabling speed autonegotiation on an interface

MIB: rlinterfaces.mib

Tables used: swIfSpeedDuplexAutoNegotiation — 1.3.6.1.4.1.89.43.1.1.16

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.43.1.1.16.{ifIndex} i {negotiation(1), no negotiation(2)}
```

Example of disabling negotiation on the TenGigabitethernet 1/0/23

CLI command:
```
interface TenGigabitethernet 1/0/23
no negotiation
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.16.23 i 2
```

Setting speed autonegotiation modes on an interface

MIB: swinterfaces.mib

Tables used: swIfAdminSpeedDuplexAutoNegotiationLocalCapabilities —1.3.6.1.4.1.89.43.1.1.40

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.43.1.1.40.{ifIndex} x "{negotiation mode(HEX-string)}"
```

Example of setting up auto-negotiation at 1000f and 10000f speeds on the TenGigabitethernet 1/0/23 interface

CLI command:
```
interface TenGigabitethernet 1/0/23
negotiation 1000f 10000f
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.40.23 x 14
```

1) In the binary system 1000f and 10000f are written as 00110000000. In the hexadecimal system they are written as 180.

2) Bit description

Default(0),
Unknown(1),
TenHalf(2),
TenFull(3),
FastHalf(4),

FastFull(5),
GigaHalf(6),
GigaFull(7),
TenGigaFull(8),
FiveGigaFull(9),
TwoPointFiveFull(10).

Bit order

10,9,8,7,6,5,4,3,2,1,0

Viewing a port duplex mode

MIB: EtherLike-MIB

Tables used: dot3StatsDuplexStatus — 1.3.6.1.2.1.10.7.2.1.19

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.10.7.2.1.19.{ifindex}
```

Example of viewing the duplex mode of the TenGigabitEthernet port 1/0/23

CLI command:
```
show interfaces status TenGigabitEthernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.2.1.19.23
```

Description of the output values

unknown (1)
halfDuplex (2)
fullDuplex (3)

Changing a duplex mode on the interface

MIB: RADLAN-rlInterfaces

Tables used: swIfDuplexAdminMode — 1.3.6.1.4.1.89.43.1.1.3

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.43.1.1.3.{ifIndex} i {none(1),half(2),full (3)}
```

Example of changing the duplex mode of the TenGigabitethernet port 1/0/23

CLI command:
```
interface TenGigabitethernet 1/0/23
 duplex half
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.3.23 i 2
```

Viewing an interface transmission medium

MIB: EtherLike-MIB

Tables used: swIfTransceiverType — 1.3.6.1.4.1.89.43.1.1.7

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.43.1.1.7.{ifindex}
```

Example of viewing the TenGigabitEthernet 1/0/23 transmission medium

CLI command:
```
show interfaces status TenGigabitEthernet 1/0/1
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.7.23
```

Description of the output values

Copper (1)
fiberOptics (2)
ComboCopper (3)
comboFiberOptics (4)

Flow control

MIB: RADLAN-rlInterfaces

Tables used: swIfFlowControlMode — 1.3.6.1.4.1.89.43.1.1.14

```
snmpset -v2c -c <community>  <IP address> \
1.3.6.1.4.1.89.43.1.1.14.{ifindex} i {on(1),off(2),auto (3)}
```

Example of enabling flow control on the TenGigabitethernet 1/0/23 interface

CLI command:
```
interface TenGigabitethernet 1/0/23
flowcontrol on
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.14.23 i 1
```

Viewing an administrative state of the port

MIB: IF-MIB

Tables used: ifAdminStatus — 1.3.6.1.2.1.2.2.1.7

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.2.2.1.7.{ifIndex}
```

Example of viewing the status of the TenGigabitethernet port 1/0/23

CLI command:
```
show interfaces status TenGigabitEthernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.7.23
```

Possible options:

up(1)
down(2)
testing(3)

Enabling/disabling a configured interface

MIB: IF-MIB

Tables used: ifAdminStatus — 1.3.6.1.2.1.2.2.1.7

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.2.1.2.2.1.7.{ifIndex} i {up(1),down(2)}
```

Example

CLI command:
```
interface TenGigabitethernet 1/0/23
shutdown
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.2.2.1.7.23 i 2
```

Enable/disable splitting on a configurable interface

MIB: eltMesHardwareMib

Tables used: eltHardwareInterfaceEntry — 1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1.2.{ifIndex} i {4*25g(3),1*100g(1)}
```

Example of enabling the HundredGigabitEthernet1/0/2 port splitting for MES5500-32

CLI command:
```
interface HundredGigabitEthernet1/0/2
hardware profile portmode 4x25g
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1.2.1 i 3
```

Viewing the actual status of the port splitting

MIB: eltMesHardwareMib

Tables used: eltHardwareInterfaceEntry — 1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1.3.{ifIndex}
```

Example of viewing the port splitting status of the HundredGigabitEthernet1/0/2 MES5500-32

CLI command:
```
show hardware profile portmode
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1.3.2
```

Possible options:

1*100g(1)
4*25g(3)

Viewing the port splitting status after reboot

MIB: eltMesHardwareMib

Tables used: eltHardwareInterfaceEntry — 1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1.2.{ifIndex}
```

Example of viewing the port splitting status HundredGigabitEthernet1/0/2 after reboot for MES5500-32

CLI command:
```
show hardware profile portmode
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.14.1.1.1.3.1.2.2
```

Possible options:

1*100g(1)
4*25g(3)

Viewing the operative state of the port

MIB: IF-MIB

Tables used: ifOperStatus — 1.3.6.1.2.1.2.2.1.8

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.2.2.1.8.{ifIndex}
```

Example of viewing the status of the TenGigabitethernet port 1/0/23

CLI command:
```
show interfaces status TenGigabitEthernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.8.23
```

Possible options:

up(1)
down(2)

Determining a port connection type

MIB: rlinterfaces.mib

Tables used: swIfTransceiverType — 1.3.6.1.4.1.89.43.1.1.7

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.43.1.1.7.{ifindex}
```

Example of determining the type of the TenGigabitethernet port 1/0/23

CLI command:
```
show interfaces status
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.7.23
```

Possible options:

regular (1)
fiberOptics (2)
comboRegular (3)
comboFiberOptics (4)

Viewing the counter of unicast packets on the interface

MIB: IF-MIB

Tables used: ifInUcastPkts — 1.3.6.1.2.1.2.2.1.11

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.2.1.2.2.1.11.{ifIndex}
```

Example of viewing an incoming unicast packets counter on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interface counters TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.11.23
```

Viewing the counter of multicast packets on the interface

MIB: IF-MIB

Tables used: ifInMulticastPkts — 1.3.6.1.2.1.31.1.1.1.2

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.2.1.31.1.1.1.2.{ifindex}
```

Example of viewing an incoming multicast packets counter on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interface counters TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.31.1.1.1.2.23
```

Viewing the counter of broadcast packets on the interface

MIB: IF-MIB

Tables used: ifInBroadcastPkts — 1.3.6.1.2.1.31.1.1.1.3

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.2.1.31.1.1.1.3.{ifindex}
```

Example of viewing an incoming broadcast packets counter on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interface counters TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.31.1.1.1.3.23
```

Viewing the octet counter on the interface

MIB: IF-MIB

Tables used:
ifInOctets — 1.3.6.1.2.1.2.2.1.10
ifHCInOctets — 1.3.6.1.2.1.31.1.1.1.6
ifOutOctets — 1.3.6.1.2.1.2.2.1.16
ifHCOutOctets — 1.3.6.1.2.1.31.1.1.1.10

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.2.1.2.2.1.10.{ifindex}
```

Example of viewing the counter of received octets on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interface counters TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.10.23
```

Octet is the number of bytes.

1 octet = 1 byte

Viewing FCS Errors counter on an interface

MIB: EtherLike-MIB

Tables used: dot3StatsFCSErrors — 1.3.6.1.2.1.10.7.2.1.3

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.2.1.10.7.2.1.3.{ifindex}
```

Example of viewing the FCS Errors counter on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interface counters TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.2.1.3.23
```

Viewing the Internal MAC Rx Errors counter on an interface

MIB: EtherLike-MIB

Tables used: dot3StatsInternalMacReceiveErrors — 1.3.6.1.2.1.10.7.2.1.16

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.2.1.10.7.2.1.16.{ifindex}
```

Example of viewing the Internal MAC Rx Errors counter on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interface counters TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.2.1.16.23
```

Example of viewing the Transmitted Pause Frames counter on an interface

MIB: EtherLike-MIB

Tables used: dot3OutPauseFrames — 1.3.6.1.2.1.10.7.10.1.4

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.2.1.10.7.10.1.4.{ifindex}
```

Example of viewing the Transmitted Pause Frames counter on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interface counters TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.10.1.4.23
```

Viewing the Received Pause Frames counter on an interface

MIB: EtherLike-MIB

Tables used: dot3InPauseFrames — 1.3.6.1.2.1.10.7.10.1.3

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.2.1.10.7.10.1.3.{ifindex}
```

Example of viewing the Received Pause Frames counter on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show interface counters TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.10.1.3.23
```

Resetting interface counters

MIB: rlInterfaces.mib

Tables used: rlIfClearPortMibCounters — 1.3.6.1.4.1.89.54.4

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.54.4.0 x {bit mask}
```

Example of cleaning the interface counter

CLI command:
```
clear counters
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.54.4.0 x
FFFFFFFF00000FFFFFFFF00000FFFFFFFF00000FFFFFFFF00000FFFFFFFF00000FFFFFFFF00000F
FFFFFFF00000FFFFFFFF00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000001FFFFFFFE000000
```

1) A bit mask is set to the stack counters reset value for all ports of all units of the stack:

**snmpset -v2c -c private 192.168.1.30 **
1.3.6.1.4.1.89.54.4.0 x 000000000000FFFFFF00000000F000000000000FFFFFF00000000F
000000000000FFFFFF00000000F000000000000FFFFFF00000000F000000000000FFFFFF0
0000000F000000000000FFFFFF00000000F000000000000FFFFFF00000000F00000000000
0FFFFFF00000000F0000000000000000000000000000001FFFE0000000000

2) To view the value of a bit mask, use the following command:

**snmpwalk -v2c -c public 192.168.1.30 **
1.3.6.1.4.1.89.54.9.0

Monitoring of switch ports load

MIB: eltMes.mib

Tables used: eltSwIfUtilizationEntry — 1.3.6.1.4.1.35265.1.23.43.2.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.43.2.1.{parametr}
```

Example

CLI command:
```
show interfaces utilization
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.43.2.1.1
```

The list of possible parameters

eltSwIfUtilizationIfIndex(1)
eltSwIfUtilizationAverageTime(2)
eltSwIfUtilizationCurrentInPkts(3)
eltSwIfUtilizationCurrentInRate(4)
eltSwIfUtilizationCurrentOutPkts(5)
eltSwIfUtilizationCurrentOutRate(6)
eltSwIfUtilizationAverageInPkts(7)
eltSwIfUtilizationAverageInRate(8)
eltSwIfUtilizationAverageOutPkts(9)
eltSwIfUtilizationAverageOutRate(10)

Enabling/disabling the unidirectional port transmission mode

MIB: ELTEX-MES-eltInterfaces

Tables used: eltSwIfTable — 1.3.6.1.4.1.35265.1.23.43.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.43.1.1.15.{index} i {disable(0), send-only(1)
```

Example of enabling unidirectional port transmission mode

CLI command:
```
interface TenGigabitEthernet1/0/1
unidirectional send-only
exit
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.43.1.1.15.1 i 1
```

6.2 *Link Aggregation Group (LAG)*

Enabling/disabling the interface as part of an aggregation group

MIB: IEEE8023-LAG-MIB

Tables used:
dot3adAggPortTable — 1.2.840.10006.300.43.1.2.1

```
snmpset -v2c -c <community> <IP address> \
1.2.840.10006.300.43.1.2.1.1.20.{ifIndex} x {auto and long timeout(A2), auto and
short timeout(E2), on(22)}\
1.2.840.10006.300.43.1.2.1.1.4.{ifIndex} i {ifIndex}
```

Example of enabling channel-group on TenGigabitethernet 1/0/1

CLI command:
```
interface TenGigabitethernet 1/0/1
channel-group 1 mode auto
```
SNMP command:
```
sudo snmpset -v2c -c private 192.168.1.30 \
1.2.840.10006.300.43.1.2.1.1.20.1 x "A2" \
1.2.840.10006.300.43.1.2.1.1.4.1 i 10000
```

Example of disabling channel-group on the TenGigabitethernet 1/0/1

CLI command:
```
interface TenGigabitethernet 1/0/1
no channel-group
```
SNMP command:
```
sudo snmpset -v2c -c private 192.168.1.30 \
1.2.840.10006.300.43.1.2.1.1.20.1 s '"'\
1.2.840.10006.300.43.1.2.1.1.4.1 i 0
```

6.3 *VLAN Configuration*

Adding VLAN to vlan database

MIB: rlvlan.mib

Tables used:
rldot1qVlanStaticList1to1024 — 1.3.6.1.4.1.89.48.69.1.2
rldot1qVlanStaticList1025to2048 — 1.3.6.1.4.1.89.48.69.1.3
rldot1qVlanStaticList2049to3072 — 1.3.6.1.4.1.89.48.69.1.4
rldot1qVlanStaticList3073to4094 — 1.3.6.1.4.1.89.48.69.1.5

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.69.1.2 x {bit mask}
```

Example of creating vlan 994 in vlan database

CLI command:
```
vlan database
vlan 994
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.69.1.2.0 x
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000040000000
```

1) When calculating a bit mask for 1025-2048, subtract 1024 from the required vlan and then perform the mask calculation. Same for vlan 2049–3072: subtract 2048 before calculation. For 3073–4094, subtract 3072. A bit mask should contain at least 10 characters.

2) An example of bit mask calculation is given in the "APPENDIX A. Bit mask calculation method" section.

Adding a VLAN to a port

MIB: rlvlan.mib

Tables used: rldot1qPortVlanStaticTable — 1.3.6.1.4.1.89.48.68

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.68.1.{1-8}.{ifIndex} x {vlan as a bit mask}
```

Example of adding vlan 622 and 3100 to the TenGigabitEthernet 1/0/23 interface in trunk mode

CLI command:

```
interface TenGigabitethernet 1/0/23
 switchport mode trunk
 switchport trunk allowed vlan add 622,3100
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.1.23 x
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000004
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.4.23 x 0000001000
```

Example of adding vlan 622 to the TenGigabitEthernet 1/0/23 interface as a native vlan

CLI command:

```
interface TenGigabitethernet 1/0/23
 switchport mode trunk
 switchport trunk native vlan 622
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.1.23 x
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000004
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.5.23 x
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000004
```

Example of adding vlan 622 to the TenGigabitEthernet 1/0/23 interface in access mode

CLI command:

```
interface TenGigabitethernet 1/0/23
 switchport access vlan 622
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.1.21 x
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000004 \
1.3.6.1.4.1.89.48.68.1.5.21 x
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000004
```

1. Table list:

rldot1qPortVlanStaticEgressList1to1024 — 1.3.6.1.4.1.89.48.68.1.1.{ifindex}
rldot1qPortVlanStaticEgressList1025to2048 — 1.3.6.1.4.1.89.48.68.1.2.{ifindex}
rldot1qPortVlanStaticEgressList2049to3072 — 1.3.6.1.4.1.89.48.68.1.3.{ifindex}
rldot1qPortVlanStaticEgressList3073to4094 — 1.3.6.1.4.1.89.48.68.1.4.{ifindex}

rldot1qPortVlanStaticUntaggedEgressList1to1024 — 1.3.6.1.4.1.89.48.68.1.5.{ifindex}
rldot1qPortVlanStaticUntaggedEgressList1025to2048 1.3.6.1.4.1.89.48.68.1.6.{ifindex}
rldot1qPortVlanStaticUntaggedEgressList2049to3072 1.3.6.1.4.1.89.48.68.1.7.{ifindex}
rldot1qPortVlanStaticUntaggedEgressList3073to4094 1.3.6.1.4.1.89.48.68.1.8.{ifindex}

2. An example of creating a bit mask is given in section «APPENDIX A. Bit mask calculation method».

3. A bit mask should contain at least 10 characters.

Ban default VLAN on a port

MIB: eltVlan.mib

Tables used: eltVlanDefaultForbiddenPorts — 1.3.6.1.4.1.35265.1.23.5.5.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.5.5.1.0 x {port as a bit mask}
```

Example of default vlan ban on the TenGigabitEthernet 1/0/23 port

CLI command:
```
interface TenGigabitethernet 1/0/23
switchport forbidden default-vlan
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.5.5.1.0 x 0000020000
```

1. An example of creating a bit mask is given in section «APPENDIX A. Bit mask calculation method».

2. A bit mask should contain at least 10 characters.

Viewing VLAN name

MIB: rlvlan.mib

Tables used: rldot1qVlanStaticName — 1.3.6.1.4.1.89.48.70.1.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.70.1.1.{vlan}
```

Example of viewing vlan 994 name

CLI command:
```
show vlan tag 994
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.48.70.1.1.994
```

Viewing port membership in VLAN

MIB: rlvlan.mib

Tables used: rldot1qPortVlanStaticTable — 1.3.6.1.4.1.89.48.68

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.68.1.{1-4}.{ifindex}
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.68.1.{5-8}.{ifindex}
```

Example of viewing a VLAN on the TenGigabitethernet 1/0/23

CLI command:

```
show interfaces switchport TenGigabitethernet 1/0/23
```

SNMP command:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.1.23
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.5.23
```

1. The example shows 2 snmpwalk commands. If the port is Tagged, values in the second command output are zero, and a VLAN number corresponds to the first command output values. If the port is Untagged, the second command output contains values other than zero, and a VLAN number corresponds to these values.

2. Table list:

rldot1qPortVlanStaticEgressList1to1024 — 1.3.6.1.4.1.89.48.68.1.1.{ifindex}
rldot1qPortVlanStaticEgressList1025to2048 — 1.3.6.1.4.1.89.48.68.1.2.{ifindex}
rldot1qPortVlanStaticEgressList2049to3072 — 1.3.6.1.4.1.89.48.68.1.3.{ifindex}
rldot1qPortVlanStaticEgressList3073to4094 — 1.3.6.1.4.1.89.48.68.1.4.{ifindex}
rldot1qPortVlanStaticUntaggedEgressList1to1024 — 1.3.6.1.4.1.89.48.68.1.5.{ifindex}
rldot1qPortVlanStaticUntaggedEgressList1025to2048 — 1.3.6.1.4.1.89.48.68.1.6.{ifindex}
rldot1qPortVlanStaticUntaggedEgressList2049to3072 — 1.3.6.1.4.1.89.48.68.1.7.{ifindex}
rldot1qPortVlanStaticUntaggedEgressList3073to4094 — 1.3.6.1.4.1.89.48.68.1.8.{ifindex}

3. The values obtained as a result of the query are a bit mask, the method of calculation of which is given in the section «APPENDIX A. Bit mask calculation method».

Port mode configuration

MIB: rlvlan.mib

Tables used: vlanPortModeEntry — 1.3.6.1.4.1.89.48.22.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.48.22.1.1.{ifIndex}   i   {general(1),   access(2),   trunk(3),
customer(7)}
```

Example of configuring the TenGigabitEthernet 1/0/23 interface in trunk mode

CLI command:
```
interface TenGigabitethernet 1/0/23
switchport mode trunk
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.22.1.1.21 i 2
```

Viewing a port mode

MIB: rlvlan.mib

Tables used: vlanPortModeState — 1.3.6.1.4.1.89.48.22.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.22.1.1.{ifindex}
```

Example of viewing mode on the TenGigabitethernet 1/0/23

CLI command:
```
show interfaces switchport TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.48.22.1.1.23
```

Possible options:

general(1)
access(2)
trunk (3)
customer (7)

Assigning pvid to an interface

MIB: Q-BRIDGE-MIB.mib

Tables used: dot1qPortVlanTable — 1.3.6.1.2.1.17.7.1.4.5

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.2.1.17.7.1.4.5.1.1.{ifindex} u {1-4094}
```

Example of pvid 15 assignment for the TenGigabitEthernet 1/0/23

CLI command:
```
interface TenGigabitethernet 1/0/23
switchport general pvid 15
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.17.7.1.4.5.1.1.23 u 15
```

Configuring map mac

MIB: rlvlan.mib

Tables used: vlanMacBaseVlanGroupTable — 1.3.6.1.4.1.89.48.45

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.45.1.3.{MAC address in DEC}.{mask} i {map-group number} \
1.3.6.1.4.1.89.48.45.1.4.{MAC address in DEC}.{mask} i {createAndGo(4),
destroy(6)}
```

Example

CLI command:

```
vlan database
map mac a8:f9:4b:33:29:c0 32 macs-group 1
```

SNMP command:

```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.48.45.1.3.168.249.75.51.41.192.32 i 1 \
1.3.6.1.4.1.89.48.45.1.4.168.249.75.51.41.192.32 i 4
```

Setting a MAC-address-binding-based VLAN classification rule for an interface

MIB: rlvlan.mib

Tables used: vlanMacBaseVlanPortTable — 1.3.6.1.4.1.89.48.46.1.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.46.1.2.58.1 u {vlan} 1.3.6.1.4.1.89.48.46.1.3.58.1 i
{createAndGo(4), destroy(6)}
```

Example of enabling the VLAN classification rule for the TenGigabitEthernet 1/0/23 interface

CLI command:

```
interface TenGigabitethernet 1/0/23
switchport general map macs-group 1 vlan 20
```

SNMP command:

```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.48.46.1.2.23.1 u 1 \
1.3.6.1.4.1.89.48.46.1.3.23.1 i 4
```

6.4 *Errdisable state configuration and monitoring*

Viewing settings for automatic interface activation

MIB: rlinterfaces_recovery.mib

Tables used: rlErrdisableRecoveryEnable — 1.3.6.1.4.1.89.128.2.1.2

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.128.2.1.2
```

Example of viewing settings for automatic interface activation

CLI command:
```
show errdisable recovery
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.128.2.1.2
```

Viewing the reason of port blocking

MIB: rlErrdisableRecoveryIfReason

Tables used: rlErrdisableRecoveryIfReason — 1.3.6.1.4.1.89.128.3.1.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.128.3.1.1
```

Example

CLI command:
```
show errdisable interfaces
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.128.3.1.1
```

Possible options:

loopback-detection (1)
port-security (2)
dot1x-src-address (3)
acl-deny (4)
stp-bpdu-guard (5)
stp-loopback-guard (6)
unidirectional-link (7)
dhcp-rate-limit (8)
l2pt-guard (9)
storm-control (10)

Configuring automatic interface activation

MIB: rlinterfaces_recovery.mib

Tables used: rlErrdisableRecoveryEnable — 1.3.6.1.4.1.89.128.2.1.2

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.128.2.1.2. {index of reason} i {true(1), false(2)}
```

Example of enabling automatic interface activation in case of loopback detection

CLI command:
```
errdisable recovery cause loopback-detection
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.128.2.1.2.1 i 1
```

Possible index of reason values, depending on the configuration type:

loopback detection — (1)
port-security — (2)
dot1x-src-address — (3)
acl-deny — (4)
stp-bpdu-guard — (5)
stp-loopback-guard (6)
unidirectional-link — (8)
storm-control — (9)
l2pt-guard — (11)

Configuring an interval for exit from the errdisable state

MIB: rlinterfaces_recovery.mib

Tables used: rlErrdisableRecoveryInterval — 1.3.6.1.4.1.89.128.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.128.1.0 i {interval 30-86400}
```

Example of setting the 30 second interval for exit from the errdisable state

CLI command:
```
errdisable recovery interval 30
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.128.1.0 i 30
```

6.5 *Configuring voice vlan*

Adding voice vlan

MIB: RADLAN-vlanVoice-MIB

Tables used: vlanVoiceAdminVid — 1.3.6.1.4.1.89.48.54.8

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.54.8.0 i {vlan id}
```

Example of adding voice vlan id 10

CLI command:
```
voice vlan id 10
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.54.8.0 i 10
```

Enabling voice vlan on an interface

MIB: RADLAN-vlanVoice-MIB

Tables used: vlanVoiceOUIBasedPortTable — 1.3.6.1.4.1.89.48.54.12.5

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.54.12.5.1.1.{ifIndex} i 1 \
1.3.6.1.4.1.89.48.54.12.5.1.2.{ifIndex} u {voice vlan id}
```

Example

CLI command:
```
interface TenGigabitethernet 1/0/23
voice vlan enable
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.54.12.5.1.1.23 i 1 \
1.3.6.1.4.1.89.48.54.12.5.1.2.23 u 10
```

Editing the OUI table

MIB: rlvlanVoice.mib

Tables used: vlanVoiceOUIBasedTable — 1.3.6.1.4.1.89.48.54.12.4

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.54.12.4.1.3.{OUI in DEC. Bytes are separated by points} i
{createAndGo(4), destroy(6)}
```

Example

CLI command:
```
voice vlan oui-table add 002618
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.48.54.12.4.1.3.0.38.24 i 4
```

6.6 *Configuring LLDP*

Global LLDP enabling/disabling

MIB: rlLldp.mib

Tables used: rlLldpEnabled — 1.3.6.1.4.1.89.110.1.1.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.110.1.1.1.0 i {true (1), false (2)}
```

Example of LLDP disabling

CLI command:
```
no Lldp run
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.110.1.1.1.0 i 2
```

Configuring lldp-med policy with the voice vlan number for tagged voice vlan traffic

MIB: rllldb.mib

Tables used: rlLldpXMedLocMediaPolicyContainerTable — 1.3.6.1.4.1.89.110.1.2.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.110.1.2.1.1.2.1 i {voice(1), voice-signaling(2), guest-voice(3),
guest-voice-signaling(4), softphone-voice(5), video-conferencing(6), streaming-
video(7), video-signaling(8)} \
1.3.6.1.4.1.89.110.1.2.1.1.3.1 i {vlan} \
1.3.6.1.4.1.89.110.1.2.1.1.4.1 i {priority} \
1.3.6.1.4.1.89.110.1.2.1.1.7.1 {true(1), false(2)} \
1 1.3.6.1.4.1.89.110.1.2.1.1.9.1 i {createAndGo(4), destroy(6)}
```

Example of configuring lldp-med policy with VLAN 10, priority 4

CLI command:

```
lldp med network-policy 1 voice vlan 10 vlan-type tagged up 4
```

SNMP command:

```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.110.1.2.1.1.2.1 i 1 \
1.3.6.1.4.1.89.110.1.2.1.1.3.1 i 10 \
1.3.6.1.4.1.89.110.1.2.1.1.4.1 i 4 \
1.3.6.1.4.1.89.110.1.2.1.1.7.1 i 1 \
1.3.6.1.4.1.89.110.1.2.1.1.9.1 i 4
```

Configuring lldp-med policy for voice vlan tagged traffic

MIB: rllldb.mib

Tables used: rlLldpXMedNetPolVoiceUpdateMode — 1.3.6.1.4.1.89.110.1.7

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.110.1.7.0 i {manual(0), auto(1)}
```

Example of configuring lldp-med policy in auto mode

CLI command:

```
no lldp med network-policy voice auto
```

SNMP command:

```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.110.1.7.0 i 1
```

7 IPV4 ADDRESSING CONFIGURATION

Creating an IP address on the interface vlan

MIB: rlip.mib

Tables used: rsIpAddrEntry — 1.3.6.1.4.1.89.26.1.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.26.1.1.2.{ip address(DEC)} i {ifIndex} \
 1.3.6.1.4.1.89.26.1.1.3.{ip address(DEC)} a {netmask}
```

Example of configuring the IP address 192.168.10.30/24 on vlan 30

CLI command:

```
interface vlan 30
ip address 192.168.10.30 /24
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.26.1.1.2.192.168.10.30 i 100029 \
1.3.6.1.4.1.89.26.1.1.3.192.168.10.30 a 255.255.255.0
```

Deleting an IP address from the interface vlan

MIB: rlip.mib

Tables used: rsIpAddrEntry — 1.3.6.1.4.1.89.26.1.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.26.1.1.2.{ip address(DEC)} i {ifIndex} \
 1.3.6.1.4.1.89.26.1.1.3.{ip address(DEC)} a {netmask} \
1.3.6.1.4.1.89.26.1.1.6.{ip address(DEC)} i 2
```

Example of deleting the IP address 192.168.10.30 from vlan 30

CLI command:

```
interface vlan 30
 no ip address 192.168.10.30
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.26.1.1.2.192.168.10.30 i 100029 \
1.3.6.1.4.1.89.26.1.1.3.192.168.10.30 a 255.255.255.0 \
1.3.6.1.4.1.89.26.1.1.6.192.168.10.30 i 2
```

Obtaining IP address via DHCP on the interface vlan

MIB: radlan-dhcpcl-mib.mib

Tables used: rlDhcpClActionStatus — 1.3.6.1.4.1.89.76.3.1.2

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.76.3.1.2.{ifIndex} i {createAndGo(4), destroy(6)}
```

Example

CLI command:
```
interface vlan 30
 ip address dhcp
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \1.3.6.1.4.1.89.76.3.1.2.100029 i 4
```

Adding/deleting a default gateway

MIB: rlip.mib

Tables used: rlInetStaticRouteEntry — 1.3.6.1.4.1.89.26.28.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.26.28.1.4.0.0.0.0.0.1.4.{IP address}.0 i {metric(4)} \
1.3.6.1.4.1.89.26.28.1.4.0.0.0.0.0.1.4.{IP address}.0 i {remote(4)} \
1.3.6.1.4.1.89.26.28.1.4.0.0.0.0.0.1.4.{IP address}.0 i {createAndGo (4),
destroy(6)}
```

Example of adding ip default gateway 192.168.1.10

CLI command:
```
ip default-gateway 192.168.1.10
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.26.28.1.7.1.4.0.0.0.0.0.1.4.192.168.1.10.0 u 4 \
1.3.6.1.4.1.89.26.28.1.8.1.4.0.0.0.0.0.1.4.192.168.1.10.0 i 4 \
1.3.6.1.4.1.89.26.28.1.10.1.4.0.0.0.0.0.1.4.192.168.1.10.0 i 4
```

8 IPV6 ADDRESSING CONFIGURATION

Enabling/disabling IPv6 addressing on the interface vlan

MIB: ip-mib.mib

Tables used: ipv6InterfaceEnableStatus — 1.3.6.1.2.1.4.30.1.5

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.2.1.4.30.1.5.{ifindex} i {enable(1), disable(2)}
```

Example of enabling IPv6 addressing on vlan 2

CLI command:

```
interface vlan 2
ipv6 enable
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.4.30.1.5.100001 i 1
```

Creating/deleting an IPv6 address from the interface vlan

MIB: rlip.mib

Tables used: rlIpAddressEntry — 1.3.6.1.4.1.89.26.36.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.26.36.1.5.{the number of bytes in an address}.{every byte in a
decimal form via separator}.2.0 i {ifindex} \
1.3.6.1.4.1.89.26.36.1.13.{the number of bytes in an address}.{every byte in a
decimal form via separator}.2.0 u {mask in a decimal form}\
1.3.6.1.4.1.89.26.36.1.11.{the number of bytes in an address}.{every byte in a
decimal form via separator}.2.0 i {createAndGo (4), destroy(6)}
```

Example of adding the address 2001::1/64 on vlan 2

CLI command:

```
interface vlan 2
ipv6 address 2001::1/64
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.26.36.1.5.2.16.32.1.0.0.0.0.0.0.0.0.0.0.0.0.1.2.0 i 100001 \
1.3.6.1.4.1.89.26.36.1.13.2.16.32.1.0.0.0.0.0.0.0.0.0.0.0.0.1.2.0 u 64 \
1.3.6.1.4.1.89.26.36.1.11.2.16.32.1.0.0.0.0.0.0.0.0.0.0.0.0.1.2.0 i 4
```

9 GREEN ETHERNET CONFIGURATION

Global disabling of green-ethernet short-reach

MIB: rlgreeneth.mib

Tables used: rlGreenEthShortReachEnable — 1.3.6.1.4.1.89.134.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.134.2.0 i {true (1), false (2)}
```

Example of disabling green-ethernet short-reach

CLI command:
```
no green-ethernet short-reach
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.134.2.0 i 2
```

Global disabling of green-ethernet energy-detect

MIB: rlgreeneth.mib

Tables used: rlGreenEthEnergyDetectEnable — 1.3.6.1.4.1.89.134.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.134.1.0 i {true (1), false (2)}
```

Example of disabling green-ethernet energy-detect

CLI command:
```
no green-ethernet energy-detect
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.134.1.0 i 2
```

Viewing green-ethernet parameters

MIB: rlgreeneth.mib

Tables used: rlGreenEthCumulativePowerSaveMeter — 1.3.6.1.4.1.89.134.5

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.134.5
```

Example of viewing green-ethernet parameters

CLI command:
```
show green-ethernet
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.134.5
```

10 RING PROTOCOLS CONFIGURATION

10.1 *ERPS protocol*

Determining the west port number

MIB: ELTEX-BRIDGE-ERPS-V2-MIB.mib

Tables used: eltexErpsMgmtRAPSWestPort — 1.3.6.1.4.1.35265.35.1.1.3.1.1.2

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.35.1.1.3.1.1.2
```

Example

CLI command:
```
show erps
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.35.1.1.3.1.1.2
```

Viewing the status of the west port

MIB: ELTEX-BRIDGE-ERPS-V2-MIB.mib

Tables used: eltexErpsMgmtRAPSWestPortState —1.3.6.1.4.1.35265.35.1.1.3.1.1.3

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.35.1.1.3.1.1.3
```

Example

CLI command:
```
show erps vlan 10
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.35.1.1.3.1.1.3
```

Possible port states:

1. Forwarding (1)
2. Blocking (2)
3. Signal-fail (3)
4. Manual-switch (4)
5. Forced-switch (5)

Determining the east port number

MIB: ELTEX-BRIDGE-ERPS-V2-MIB.mib

Tables used: eltexErpsMgmtRAPSEastPort — 1.3.6.1.4.1.35265.35.1.1.3.1.1.4

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.35265.35.1.1.3.1.1.4
```

Example

CLI command:
```
show erps
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.35.1.1.3.1.1.4
```

Viewing the east port state

MIB: ELTEX-BRIDGE-ERPS-V2-MIB.mib

Tables used: eltexErpsMgmtRAPSEastPortState — 1.3.6.1.4.1.35265.35.1.1.3.1.1.5

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.35265.35.1.1.3.1.1.5
```

Example

CLI command:
```
show erps vlan 10
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.35.1.1.3.1.1.5
```

Possible port states:

1. Forwarding (1)
2. Blocking (2)
3. Signal-fail (3)
4. Manual-switch (4)
5. Forced-switch (5)

Viewing a ring state

MIB: ELTEX-BRIDGE-ERPS-V2-MIB.mib

Tables used: eltexErpsMgmtRAPSRingState — 1.3.6.1.4.1.35265.35.1.1.3.1.1.12

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.35265.35.1.1.3.1.1.12
```

Example

CLI command:
```
show erps vlan 10
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.35.1.1.3.1.1.12
```

Possible states of the erps ring:

1. Init (1)
2. Idle(2)
3. Protection (3)
4. Manual-switch (4)
5. Forced-switch (5)
6. Pending (6)

10.2 *Spanning Tree protocol configuration*

Enabling/disabling spanning-tree

MIB: radlan-brgmacswitch.mib

Tables used: rldot1dStp — 1.3.6.1.4.1.89.57.2.3

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.57.2.3.0 i {enabled(1), disabled(2)}
```

Example of spanning-tree disabling

CLI command:
```
no spanning-tree
```
SNMP command:
```
snmpset -v2c -c  private  192.168.1.30 \
1.3.6.1.4.1.89.57.2.3.0 i 2
```

Enabling/disabling spanning-tree on a configured interface

MIB: BRIDGE-MIB

Tables used: dot1dStpPortTable — 1.3.6.1.2.1.17.2.15.1.4

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.2.1.17.2.15.1.4.{ifIndex} i {enabled(1), disabled(2)}
```

Example of disabling the spanning-tree on the TenGigabitEthernet interface 1/0/23

CLI command:
```
interface TenGigabitethernet 1/0/23
spanning-tree disable
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.2.1.17.2.15.1.4.23 i 2
```

Enabling/disabling BPDU packet processing by an interface with STP protocol disabled

MIB: radlan-bridgemibobjects-mib.mib

Tables used: rldot1dStpPortTable — 1.3.6.1.4.1.89.57.2.13.1.4

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.57.2.13.1.4.{ifIndex} i {filtering(1), flooding(2)}
```

Example of enabling BPDU filtering on the TenGigabitethernet 1/0/23 interface

CLI command:
```
interface tengigabitethernet 1/0/23
spanning-tree bpdu filterin
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.57.2.13.1.4.23 i 1
```

Configuring spanning-tree operation mode

MIB: draft-ietf-bridge-rstpmib.mib

Tables used: dot1dStpVersion — 1.3.6.1.2.1.17.2.16

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.2.1.17.2.16.0 i {stp(0), rstp(2), mstp(3)}
```

Example of Spanning-tree operation mode setting

CLI command:
```
spanning-tree mode rstp
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.2.1.17.2.16.0 i 2
```

Viewing port role in STP

MIB: radlan-bridgemibobjects-mib.mib

Tables used: rldot1dStpPortRole — 1.3.6.1.4.1.89.57.2.13.1.7

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.57.2.13.1.7.{ifindex}
```

Example of viewing the TenGigabitethernet 1/0/23 role in STP

CLI command:
```
show spanning-tree TenGigabitethernet 1/0/23
```

SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.89.57.2.13.1.7.23
```

Possible port states:

1. Disabled (1)
2. Alternate (2)
3. Backup(3)
4. Root(4)
5. Designated(5)

Viewing port state in MSTP

MIB: radlan-bridgemibobjects-mib.mib

Tables used: rldot1sMstpInstancePortState — 1.3.6.1.4.1.89.57.6.2.1.4

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.57.6.2.1.4.1.{ifindex}
```

Example of viewing the state of the TenGigabitethernet 1/0/23 in mstp

CLI command:
```
show spanning-tree TenGigabitethernet0/23
```

SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.89.57.6.2.1.4.1.23
```

Possible port states:

1. Disabled (1)
2. Blocking (2)
3. Listening (3)
4. Forwarding(5)

The number of topology changes

MIB: BRIDGE-MIB

Tables used: dot1dStpTopChanges — 1.3.6.1.2.1.17.2.4.0

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.17.2.4.0
```

Example of viewing the topology change

CLI command:
```
show spanning-tree
```

SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.2.1.17.2.4.0
```

Viewing the time from last topology change

MIB: MIB: BRIDGE-MIB

Tables used: dot1dStpTimeSinceTopologyChange — 1.3.6.1.2.1.17.2.3.0

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.17.2.3.0
```

Example of viewing from the last rebuild

CLI command:
```
show spanning-tree
```

SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.2.1.17.2.3.0
```

Viewing the interface from which the last topology change was accepted

MIB: eltBridgeExtMIB.mib

Tables used: eltdot1dStpLastTopologyChangePort — 1.3.6.1.4.1.35265.1.23.1.401.0.5.2

```
snmpwalk -v2c -c <community> <IP address> \1.3.6.1.4.1.35265.1.23.1.401.0.5.2
```

Example of viewing the interface from which the last topology change was accepted

CLI command:

```
show spanning-tree
```

SNMP command:

```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.401.0.5.2
```

11 MULTICAST ADDRESSING

11.1 *Multicast addressing rules*

Prohibition of adding the port dynamically to a multicast group

MIB: rlbrgmulticast.mib

Tables used: rlBrgStaticInetMulticastEntry — 1.3.6.1.4.1.89.116.5.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.116.5.1.6.{vlan id}.1.4.{ip address(DEC)}.1.4.0.0.0.0 x
0000000000000000 \
 1.3.6.1.4.1.89.116.5.1.7.{vlan id}.1.4.{ip address(DEC)}.1.4.0.0.0.0 x
{Interface bit mask} \
1.3.6.1.4.1.89.116.5.1.8.{vlan id}.1.4.{ip address(DEC)}.1.4.0.0.0.0 i
{createAndGo(4), destroy (6)}
```

Example of prohibiting 239.200.200.17 on the TenGigabitEthernet 1/0/23 in vlan 622

CLI command:

```
interface vlan 622
bridge multicast forbidden ip-address 239.200.200.17 add TenGigabitEthernet
1/0/23
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.116.5.1.6.622.1.4.239.200.200.17.1.4.0.0.0.0 x 0000000000000000
\
1.3.6.1.4.1.89.116.5.1.7.622.1.4.239.200.200.17.1.4.0.0.0.0 x 0000020000 \
1.3.6.1.4.1.89.116.5.1.8.622.1.4.239.200.200.17.1.4.0.0.0.0 i 4
```

1) The total number of digits in OID 1.3.6.1.4.1.89.116.5.1.6 and OID 1.3.6.1.4.1.89.116.5.1.7 must be the same and even.

2) The method of calculating a bit mask can be found in the section "APPENDIX A. Bit mask calculation method".

Prohibition of unregistered Multicast traffic passing

MIB: rlbrgmulticast.mib

Tables used: rlMacMulticastUnregFilterEnable — 1.3.6.1.4.1.89.55.4.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.55.4.1.0 x "{bit mask for interfaces}"
```

Example of prohibition of unregistered Multicast traffic passing for the TenGigabitEthernet 1/0/20-21 ports

CLI command:

```
interface range TenGigabitEthernet 1/0/20-21
bridge multicast unregistered filtering
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.55.4.1.0 x "0000180000000000"
```

1) To delete a setting, replace the corresponding fields by 0.

2) The method of calculating a bit mask can be found in the section "APPENDIX A. Bit mask calculation method".

Multicast traffic filtering

MIB: rlbrgmulticast.mib

Tables used: rlMacMulticastEnable — 1.3.6.1.4.1.89.55.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.55.1.0 i {true(1), false(2)}
```

Example of enabling multicast address filtering

CLI command:
```
bridge multicast filtering
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.55.1.0 i 1
```

Global enabling of igmp snooping

MIB: rlbrgmulticast.mib

Tables used: rlIgmpSnoopEnable — 1.3.6.1.4.1.89.55.2.2

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.55.2.2.0 i {true(1), false(2)}
```

Example

CLI command:
```
ip igmp snooping
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.55.2.2.0 i 1
```

Enabling igmp snooping in vlan

MIB: rlbrgmulticast.mib

Tables used: rlIgmpMldSnoopVlanEnable — 1.3.6.1.4.1.89.55.5.5.1.3

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.55.5.5.1.3.1.{vlan id} i {true(1), false(2)}
```

Example of igmp snooping enabling in vlan 30

CLI command:
```
ip igmp snooping vlan 30
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.55.5.5.1.3.1.30 i 1
```

Viewing igmp snooping table

MIB: rlbrgmulticast.mib

Tables used: rlIgmpMldSnoopMembershipTable — 1.3.6.1.4.1.89.55.5.4

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.55.5.4
```

Example

CLI command:
```
show ip igmp snooping groups
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.55.5.4
```

Multicast-tv vlan (MVR) configuration

MIB: rlvlan.mib

Tables used: vlanMulticastTvEntry — 1.3.6.1.4.1.89.48.44.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.44.1.1.{ifIndex} u {vlan-id} \
1.3.6.1.4.1.89.48.44.1.2.50 i {createAndGo(4), destroy (6)}
```

Example of configuring multicast-tv vlan 622 on the TenGigabitEthernet 1/0/23 interface

CLI command:
```
interface tengigabitethernet 1/0/23
switchport access multicast-tv vlan 622
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.44.1.1.23 u 622 \
1.3.6.1.4.1.89.48.44.1.2.23 i 4
```

Setting of multicast-tv vlan <customer/access/trunk/general> operation mode depends on the port setting mode, i.e. the switchport mode command customer/access/trunk/general.

11.2 *Multicast traffic restriction functions*

Multicast snooping profile creation

MIB: eltIpMulticast.mib

Tables used: eltMesIpMulticast — 1.3.6.1.4.1.35265.1.23.46.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.46.1.1.2.{Index of profile} s {profile name} \
1.3.6.1.4.1.35265. 1.23.46.1.1.3.{Index of profile} i {deny(1), permit(2)} \
1.3.6.1.4.1.35265. 1.23.46.1.1.4.{Index of profile} i {createAndGo(4),
destroy(6)}
```

Example of creating a profile with the name IPTV (assuming the profile will have the serial number 3)

CLI command:
```
multicast snooping profile IPTV
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.46.1.1.2.3 s IPTV \
1.3.6.1.4.1.35265.1.23.46.1.1.3.3 i 1 \
1.3.6.1.4.1.35265.1.23.46.1.1.4.3 i 4
```

Specification of Multicast address ranges in multicast snooping profile

MIB: eltIpMulticast.mib

Tables used: eltMesIpMulticast — 1.3.6.1.4.1.35265. 1.23.46.3

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.46.3.1.3.{index of rule}.{Index of profile} i
{ip(1),ipv6(2)} \
1.3.6.1.4.1.35265. 1.23.46.3.1.4.{index of rule}.{Index of profile} x {ip
address of the beginning of the range in hexadecimal form} \
1.3.6.1.4.1.35265. 1.23.46.3.1.5.{index of rule}.{Index of profile} x {ip
address of the end of the range in hexadecimal form} \
1.3.6.1.4.1.35265. 1.23.46.3.1.6.{index of rule}.{Index of profile} i
{createAndGo(4), destroy(6)}
```

Example of a restriction of multicast groups 233.7.70.1-233.7.70.10 for a profile with the name IPTV (assume that the profile has a serial number 3. There are 2 rules in the first profile and one in the second)

CLI command:
```
multicast snooping profile IPTV
 match ip 233.7.70.1 233.7.70.10
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.46.3.1.3.4.3 i 1 \
1.3.6.1.4.1.35265.1.23.46.3.1.4.4.3 x E9074601 \
1.3.6.1.4.1.35265.1.23.46.3.1.5.4.3 x E907460A \
1.3.6.1.4.1.35265.1.23.46.3.1.6.4.3 i 4
```

index of rule **is calculated as the sum of all rules in all profiles.**

Assigning multicast snooping profile to a port

MIB: eltIpMulticast.mib

Tables used: eltMesIpMulticast — 1.3.6.1.4.1.35265. 1.23.46.7.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.46.7.1.1.{ifIndex}.{Index of profile} i {ifIndex} \
1.3.6.1.4.1.35265. 1.23.46.7.1.2.{ifIndex}.{Index of profile} i {Index of
profile} \
1.3.6.1.4.1.35265. 1.23.46.7.1.3.{ifIndex}.{Index of profile} i
{createAndGo(4), destroy(6)}
```

Example of adding a test profile (with profile index 3) to the TenGigabitEthernet 1/0/23 interface

CLI command:

```
interface TenGigabitEthernet 1/0/23
 multicast snooping add test
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.46.7.1.1.23.3 i 23 \
1.3.6.1.4.1.35265.1.23.46.7.1.2.23.3 i 3 \
1.3.6.1.4.1.35265.1.23.46.7.1.3.23.3 i 4
```

Setting a limit on the number of Multicast groups on the port

MIB: eltIpMulticast.mib

Tables used: eltMesIpMulticast — 1.3.6.1.4.1.35265.1.23.46.6.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.46.6.1.2.{ifIndex} i {MAX number}
```

Example of setting a limit for three Multicast groups on the TenGigabitethernet 1/0/23 interface

CLI command:

```
interface TenGigabitethernet 1/0/23
 multicast snooping max-groups 3
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.46.6.1.2.23 i 3
```

12 CONTROL FUNCTIONS

12.1 *AAA mechanism*

Adding a new user

MIB: rlaaa.mib

Tables used: rlAAALocalUserTable — 1.3.6.1.4.1.89.79.17

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.17.1.1.{number of letters}.{Login in DEC, each letter of the
login is separated from the next one by a point} s {login} \
1.3.6.1.4.1.89.79.17.1.2.{number of letters}.{Login in DEC, each letter of the
login is separated from the next one by a point} s "#{encoding password}" \
1.3.6.1.4.1.89.79.17.1.3.{number of letters}.{Login in DEC, each letter of the
login is separated from the next one by a point} i {privelege level(1-15)} \
1.3.6.1.4.1.89.79.17.1.4.{number of letters}.{Login in DEC, each letter of the
login is separated from the next one by a point} i {create and go(4)}
```

Example of adding a techsup user with password 'password' and privilege level 15

CLI command:

```
username techsup password password privilege 15
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.79.17.1.1.7.116.101.99.104.115.117.112 s techsup \
1.3.6.1.4.1.89.79.17.1.2.7.116.101.99.104.115.117.112 s
"#5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8" \
1.3.6.1.4.1.89.79.17.1.3.7.116.101.99.104.115.117.112 i 15
\1.3.6.1.4.1.89.79.17.1.4.7.116.101.99.104.115.117.112 i 4
```

1. The login is converted from ASCII to HEX using a table, which can be found at https://ru.wikipedia.org/wiki/ASCII.

2. The password is set only in encrypted form, must be written in inverted commas, and # is added before the password.

Configuring authorization methods for login user

MIB: rlaaa.mib

Tables used: rlAAAMethodListEntry — 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.15.1.{authindex}.15.{“login_c_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.15.{“login_n_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.15.{“login_c_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.15.{“login_n_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
```

```
1.3.6.1.4.1.89.79.15.1.10.15.{“login_c_default” in DEC, each login letter is
separated from the next one by a dot} i {disable (0), enable(1)} \
1.3.6.1.4.1.89.79.15.1.10.15.{“login_n_default” in DEC, each login letter is
separated from the next one by a dot} i {disable (0), enable(1)}
```

Example

CLI command:

```
aaa authentication login authorization default radius local
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.79.15.1.2.15.108.111.103.105.110.95.99.95.100.101.102.97.117.108
.116 i 4 \
1.3.6.1.4.1.89.79.15.1.2.15.108.111.103.105.110.95.110.95.100.101.102.97.117.10
8.116 i 4 \
1.3.6.1.4.1.89.79.15.1.3.15.108.111.103.105.110.95.99.95.100.101.102.97.117.108
.116 i 3 \
1.3.6.1.4.1.89.79.15.1.3.15.108.111.103.105.110.95.110.95.100.101.102.97.117.10
8.116 i 3 \
1.3.6.1.4.1.89.79.15.1.10.15.108.111.103.105.110.95.99.95.100.101.102.97.117.10
8.116 i 1 \
1.3.6.1.4.1.89.79.15.1.10.15.108.111.103.105.110.95.110.95.100.101.102.97.117.1
08.116 i 1
```

authindex — the index of the authorization method. The available values are from 2 to 7. The method with the lowest number is used first.

The field 1.3.6.1.4.1.89.79.15.1.10.15 allows authorization for the login user.

108.111.103.105.110.95.99.95.100.101.102.97.117.108.116 is converted from the ASCII table (login_c_default decrypted).

108.111.103.105.110.95.110.95.100.101.102.97.117.108.116 is converted from the ASCII table (login_n_default).

Removing the authorization method settings for the login user

MIB: rlaaa.mib

Tables used: rlAAAMethodListEntry — 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.15.1.{authindex}.15.{“login_c_default” in DEC, each login
letter is separated from the next one by a dot} i {deny(0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.15.{“login_n_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.15.{“login_c_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.15.{“login_n_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.10.15.{“login_c_default” in DEC, each login letter is
separated from the next one by a dot} i {disable (0), enable(1)} \
1.3.6.1.4.1.89.79.15.1.10.15.{“login_n_default” in DEC, each login letter is
separated from the next one by a dot} i {disable (0), enable(1)}
```

Example of removing authorization methods for login user

CLI command:

```
no aaa authentication login default
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.79.15.1.2.15.108.111.103.105.110.95.99.95.100.101.102.97.117.108
.116 i 3 \
1.3.6.1.4.1.89.79.15.1.2.15.108.111.103.105.110.95.110.95.100.101.102.97.117.10
8.116 i 3 \
1.3.6.1.4.1.89.79.15.1.3.15.108.111.103.105.110.95.99.95.100.101.102.97.117.108
.116 i 0 \
1.3.6.1.4.1.89.79.15.1.3.15.108.111.103.105.110.95.110.95.100.101.102.97.117.10
8.116 i 0 \
1.3.6.1.4.1.89.79.15.1.10.15.108.111.103.105.110.95.99.95.100.101.102.97.117.10
8.116 i 0 \
1.3.6.1.4.1.89.79.15.1.10.15.108.111.103.105.110.95.110.95.100.101.102.97.117.1
08.116 i
```

authindex — the index of the authorization method. The available values are from 2 to 7. The method with the lowest number is used first.

The field 1.3.6.1.4.1.89.79.15.1.10.15 allows authorization for the login user.

108.111.103.105.110.95.99.95.100.101.102.97.117.108.116 is converted from the ASCII table (login_c_default decrypted).

108.111.103.105.110.95.110.95.100.101.102.97.117.108.116 is converted from the ASCII table (login_n_default).

Configuring authorization methods for enable user

MIB: rlaaa.mib

Tables used: rlAAAMethodListEntry — 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.15.1.{authindex}.16.{“login_c_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.16.{“login_n_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.16.{“login_c_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.16.{“login_n_default” in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.10.16.{“login_c_default” in DEC, each login letter is
separated from the next one by a dot} i {disable (0), enable(1)} \
1.3.6.1.4.1.89.79.15.1.10.16.{“login_n_default” in DEC, each login letter is
separated from the next one by a dot} i {disable (0), enable(1)}
```

Example of configuring authorization methods for enable user

CLI command:

```
aaa authentication enable authorization default radius enable
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.1
08.116 i 4 \
1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.110.95.100.101.102.97.117.
108.116 i 4 \
1.3.6.1.4.1.89.79.15.1.3.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.1
08.116 i 2 \
1.3.6.1.4.1.89.79.15.1.3.16.101.110.97.98.108.101.95.110.95.100.101.102.97.117.
108.116 i 2 \
1.3.6.1.4.1.89.79.15.1.10.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.
108.116 i 1 \
1.3.6.1.4.1.89.79.15.1.10.16.101.110.97.98.108.101.95.110.95.100.101.102.97.117
.108.116 i 1
```

authindex — the index of the authorization method. The available values are from 2 to 7. The method with the lowest number is used first.

The field 1.3.6.1.4.1.89.79.15.1.10.16 allows authorization for the enable user.

101.110.97.98.108.101.95.99.95.100.101.102.97.117.108.116 is converted from the ASCII table (enable_c_default).

101.110.97.98.108.101.95.110.95.100.101.102.97.117.108.116 is translated from the ASCII table (enable_n_default).

Removing the authorization method settings for the enable user

MIB: rlaaa.mib

Tables used: rlAAAMethodListEntry — 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.15.1.{authindex}.16.{"login_c_default" in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.16.{"login_n_default" in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.16.{"login_c_default" in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.{authindex}.16.{"login_n_default" in DEC, each login
letter is separated from the next one by a dot} i {deny (0),
line(1),enable(2),local(3)radius(4),tacacs(5),none(6)} \
1.3.6.1.4.1.89.79.15.1.10.16.{"login_c_default" in DEC, each login letter is
separated from the next one by a dot} i {disable (0), enable(1)} \
```

Example of removing authorization methods for enable user

CLI command:

```
no aaa authentication enable default
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
>1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.
108.116 i 2 \
>1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.110.95.100.101.102.97.117
.108.116 i 2 \
>1.3.6.1.4.1.89.79.15.1.3.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.
108.116 i 0 \
>1.3.6.1.4.1.89.79.15.1.3.16.101.110.97.98.108.101.95.110.95.100.101.102.97.117
.108.116 i 0 \
>1.3.6.1.4.1.89.79.15.1.10.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117
.108.116 i 0 \
>1.3.6.1.4.1.89.79.15.1.10.16.101.110.97.98.108.101.95.110.95.100.101.102.97.11
7.108.116 i 0
```

authindex — the index of the authorization method. The available values are from 2 to 7. The method with the lowest number is used first.

The field 1.3.6.1.4.1.89.79.15.1.10.16 allows authorization for the enable user.

101.110.97.98.108.101.95.99.95.100.101.102.97.117.108.116 is converted from the ASCII table (enable_c_default).

101.110.97.98.108.101.95.110.95.100.101.102.97.117.108.116 is translated from the ASCII table (enable_n_default).

12.2 *Access configuration*

Enabling the TELNET server

MIB: radlan-telnet-mib.mib

Tables used: rlTelnetEnable — 1.3.6.1.4.1.89.58.7

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.58.7.0 i {on(1), off(2)}
```

Example of TELNET server enabling

CLI command:

```
ip telnet server
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.58.7.0 i 1
```

Enabling the SSH server

MIB: rlssh.mib

Tables used: rlSshServerEnable — 1.3.6.1.4.1.89.78.2.102

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.78.2.102.0 i {on(1), off(2)}
```

Example of enabling the SSH server

CLI command:
```
ip ssh server
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.78.2.102.0 i 1
```

Viewing active sessions

MIB: rlAAA.mib

Tables used: rlAAAUserInetName — 1.3.6.1.4.1.89.79.57.1.5

```
snmpwalk -v2c -c <community> <IP address>  \
1.3.6.1.4.1.89.79.57.1.5
```

Example of viewing active sessions

CLI command:
```
show users
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.79.57.1.5
```

13 PORT MIRRORING

Port Mirroring configuration

MIB: rlspan.mib

Tables used:
rlSpanDestinationTable — 1.3.6.1.4.1.89.219.2
rlSpanSourceTable — 1.3.6.1.4.1.89.219.3

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.219.2.1.2.{session id} i {ifindex dst port} \
1.3.6.1.4.1.89.219.2.1.3.{session id} i {span(1), rspan-start(2), rspan-
final(3)} \
1.3.6.1.4.1.89.219.2.1.4.{session id} i {monitor-only(1), network(2)} \
1.3.6.1.4.1.89.219.2.1.5.{session id} i {vlan id} \
1.3.6.1.4.1.89.219.2.1.6.{session id} i {createAndGo(4), destroy(6)}

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.219.3.1.4.{session id}.1.{ifindex src port} i {rx(1), tx(2),
both(3)} \
1.3.6.1.4.1.89.219.3.1.5.{session id}.1.{ifindex src port} i {createAndGo(4),
destroy(6)}
```

Example of traffic mirroring from the TenGigabitEthernet 1/0/16 interface to the TenGigabitEthernet 1/0/17 interface

CLI command:

```
monitor session 7 destination interface TenGigabitEthernet 1/0/17
monitor session 7 source interface TenGigabitEthernet 1/0/16
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.219.2.1.2.7 i 17 \
1.3.6.1.4.1.89.219.2.1.3.7 i 1 \
1.3.6.1.4.1.89.219.2.1.4.7 i 1 \
1.3.6.1.4.1.89.219.2.1.5.7 i 1 \
1.3.6.1.4.1.89.219.2.1.6.7 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.219.3.1.4.7.1.16 i 3 \
1.3.6.1.4.1.89.219.3.1.5.7.1.16 i 4
```

Enabling vlan mirroring

MIB: rlspan.mib

Tables used:
rlSpanDestinationTable — 1.3.6.1.4.1.89.219.2
rlSpanSourceTable — 1.3.6.1.4.1.89.219.3

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.2.1.16.22.1.3.1.1.4.{ifindex vlan}.{ifindex dst port} i
{copyRxOnly(1)} \
 1.3.6.1.2.1.16.22.1.3.1.1.5.{ifindex vlan}.{ifindex dst port} i
{createAndGo(4), destroy(6)}
```

Example of configuring vlan 622 mirroring on the TenGigabitEthernet 1/0/17 interface

CLI command:

```
monitor session 7 destination interface TenGigabitEthernet 1/0/17
monitor session 7 source interface vlan 622
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.219.2.1.2.1 i 17 \
1.3.6.1.4.1.89.219.2.1.3.1 i 1 \
1.3.6.1.4.1.89.219.2.1.4.1 i 1 \
1.3.6.1.4.1.89.219.2.1.5.1 i 1 \
1.3.6.1.4.1.89.219.2.1.6.1 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.219.3.1.4.1.2.100621 i 1 \
1.3.6.1.4.1.89.219.3.1.5.1.2.100621 i 4
```

14 PHYSICAL LAYER DIAGNOSTIC FUNCTIONS

14.1 *Optical transceiver diagnostics*

DDM readings

MIB: rlphy.mib

Tables used: rlPhyTestGetResult — 1.3.6.1.4.1.89.90.1.2.1.3

```
snmpwalk  -v2c  -c  <community>  <IP address> \
 1.3.6.1.4.1.89.90.1.2.1.3.{port index}.{parameter type}
```

Example of requesting DDM readings from the TenGigabitethernet 1/0/23 interface (for all parameters)

CLI command:
```
show fiber-ports optical-transceiver interface TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.89.90.1.2.1.3.23
```

Parameter type can take the following values:

rlPhyTestTableTransceiverTemp (5) — SFP transceiver temperature;
rlPhyTestTableTransceiverSupply (6) — power supply voltage in µV;
rlPhyTestTableTxBias (7) — bias current in µA;
rlPhyTestTableTxOutput (8) — transmission power level in mDbm;
rlPhyTestTableRxOpticalPower (9) — reception power level in mDbm.

Viewing the serial number of the SFP transceiver

MIB: eltMes.mib

Tables used: eltMesPhdTransceiver — 1.3.6.1.4.1.35265.1.23.53

```
snmpwalk  -v2c  -c  <community>  <IP address> \
 1.3.6.1.4.1.35265.1.23.53.1.1.1.6.{port index}
```

Example of viewing the SFP serial number from the TenGigabitEthernet 1/0/23 interface (for all parameters)

CLI command:
```
show fiber-ports optical-transceiver interface TenGigabitethernet 1/0/23
```
SNMP command:
```
snmpwalk -v2c -c  public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.53.1.1.1.6.23
```

15 SECURITY FUNCTIONS

15.1 *Port security functions*

Limiting the number of MAC addresses learned on Ethernet ports

MIB: rlInterfaces.mib

Tables used: swIfTable — 1.3.6.1.4.1.89.43.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.43.1.1.38.{ifIndex} i {max mac addresses}
```

Example of a limit of 20 MAC addresses on the TenGigabitEthernet port 1/0/23

CLI command:

```
interface TenGigabitethernet 1/0/23
 port security max 20
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.38.23 i 20
```

Enabling port security

MIB: rlInterfaces.mib

Tables used:swIfPortLockIfRangeTable — 1.3.6.1.4.1.89.43.6

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.43.6.1.3.1 i {locked(1), unlocked(2)} \
 1.3.6.1.4.1.89.43.6.1.4.1 i {discard(1), forwardNormal(2), discardDisable(3),
action on a package that is not covered by port security regulations} \
 1.3.6.1.4.1.89.43.6.1.5.1 i {true(1), false(2). For trap sending} \
 1.3.6.1.4.1.89.43.6.1.6.1 i {trap sending frequency (s)} \
 1.3.6.1.4.1.89.43.6.1.2.1 x {ifindex as a bit mask}
```

Example of configuring port security for the TenGigabitEthernet 1/0/21-23 interfaces

CLI command:

```
interface range TenGigabitEthernet 1/0/21-23
 port security discard trap 30
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.6.1.3.1 i 1 \
1.3.6.1.4.1.89.43.6.1.4.1 i 1 \
1.3.6.1.4.1.89.43.6.1.5.1 i 1 \
1.3.6.1.4.1.89.43.6.1.6.1 i 30 \
1.3.6.1.4.1.89.43.6.1.2.1 x "00000E0000"
```

Example of bit mask calculation is given in section "APPENDIX A. Bit mask calculation method".

Setting port security operation mode

MIB: rlInterfaces.mib

Tables used: swIfTable — 1.3.6.1.4.1.89.43.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.43.1.1.37.{ifIndex} i {disabled(1), dynamic(2), secure-
permanent(3), secure-delete-on-reset(4)}
```

Example of setting a limit on the number of MAC addresses learned on the TenGigabitEthernet 1/0/23 port

CLI command:
```
interface TenGigabitethernet 1/0/23
 port security mode max-addresses
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.37.23 i 2
```

Viewing port security status

MIB: rlInterfaces.mib

Tables used: swIfLockAdminStatus — 1.3.6.1.4.1.89.43.1.1.8

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.43.1.1.8
```

Example of viewing port security state

CLI command:
```
show ports security
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.8
```

Viewing port security type

MIB: rlInterfaces.mib

Tables used: swIfAdminLockAction — 1.3.6.1.4.1.89.43.1.1.20

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.43.1.1.20
```

Example of viewing port security type

CLI command:
```
show ports security
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.20
```

Viewing the maximum specified number of MAC addresses learned on Ethernet ports

MIB: rlInterfaces.mib

Tables used: swIfLockMaxMacAddresses — 1.3.6.1.4.1.89.43.1.1.38

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.43.1.1.38
```

Example of viewing the maximum specified number of MAC addresses learned on Ethernet ports

CLI command:
```
show ports security
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.38
```

Switching a port to isolation mode and within a group of ports

MIB: rlprotectedport.mib

Tables used: rlProtectedPortsTable — 1.3.6.1.4.1.89.132.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.132.1.1.1.{Ifindex} i {not-protected(1), protected(2)}
```

Example of isolation settings on the TenGigabitEthernet 1/0/21 and the TenGigabitEthernet 1/0/23 ports

CLI command:
```
interface range TenGigabitEthernet 1/0/23
switchport protected-port
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.132.1.1.1.21 i 2 \
1.3.6.1.4.1.89.132.1.1.1.23 i 2
```

Create a static bind in MAC table

MIB: Q-BRIDGE-MIB

Tables used: dot1qStaticUnicastTable — 1.3.6.1.2.1.17.7.1.3.1

```
snmpset -v2c -c <community> -t 20 -r 0 <IP address> \
 1.3.6.1.2.1.17.7.1.3.1.1.4.{vlan id}.{mac address(DEC). MAC address bytes are
separated by points}.{ifIndex} i {other(1), invalid(2), permanent(3),
deleteOnReset(4), deleteOnTimeout(5)}
```

Example of binding MAC address 00:22:68:7d:0f:3f in vlan 622 to the TenGigabitethernet 1/0/23 interface in secure mode (permanent mode is used by default)

CLI command:
```
mac address-table static 00:22:68:7d:0f:3f vlan 622 interface tenGigabitethernet
1/0/23 secure
```

SNMP command:
```
snmpset -v2c -c private -t 20 -r 0 192.168.1.30 \
1.3.6.1.2.1.17.7.1.3.1.1.4.622.0.34.104.125.15.63.23 i 1
```

View MAC table

MIB: Q-BRIDGE-MIB

Tables used: dot1qTpFdbTable — 1.3.6.1.2.1.17.7.1.2.2

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.2.1.17.7.1.2.2
```

Example of viewing a MAC table

CLI command:
```
show mac address-table
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.17.7.1.2.2
```

Creating a static bind in ARP table

MIB: RFC1213-MIB

Tables used: ipNetToMediaTable — 1.3.6.1.2.1.4.22

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.2.1.4.22.1.2.{vlan id}.{IP address} x {„MAC address"}  \
1.3.6.1.2.1.4.22.1.3.{vlan id}.{IP address}  a {IP address} \
1.3.6.1.2.1.4.22.1.4.{vlan id}.{IP address}  i 4
```

Example of binding ip 192.168.1.21 and MAC aa:bb:cc:dd:ee:ff to vlan 1

CLI command:
```
arp 192.168.1.21 aa:bb:cc:dd:ee:ff vlan 1
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.4.22.1.2.100000.192.168.1.21 x "aabbccddeeff" \
1.3.6.1.2.1.4.22.1.3.100000.192.168.1.21 a 192.168.1.21 \
1.3.6.1.2.1.4.22.1.4.100000.192.168.1.21 i 4
```

1. To remove the binding, assign the value 2 in the field 1.3.6.1.2.1.4.22.1.4.
2. The IP address of the device and the IP address of the created static record in the ARP table must be in the same subnet.

Viewing ARP table

MIB: RFC1213-MIB.mib, Q-BRIDGE-MIB.mib

Tables used:
pNetToMediaPhysAddress — 1.3.6.1.2.1.4.22.1.2
dot1qTpFdbEntry — 1.3.6.1.2.1.17.7.1.2.2.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.4.22.1.2.{(2) ip address, (3)MAC address}

snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.17.7.1.2.2.1
```

Example of viewing ARP table

CLI command:
```
show arp
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.4.22.1.2
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.17.7.1.2.2.1
```

1. The pNetToMediaPhysAddress table value shows VLAN IP and MAC addresses.
2. The dot1qTpFdbEntry table value shows the status and the identification number of the port from which the device is available.

15.2 *DHCP control and option 82*

Enabling/disabling the DHCP server function on the switch

MIB: rldhcp.mib

Tables used: rlDhcpRelayInterfaceListTable — 1.3.6.1.4.1.89.38.29

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.38.30.0 i {true(1), false(2)}
```

Example of enabling DHCP server on the switch

CLI command:
```
ip dhcp server
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.38.30.0 i 1
```

Viewing the dhcp snooping table entries

MIB: rlBridgeSecurity.mib

Tables used: rlIpDhcpSnoopEntry — 1.3.6.1.4.1.89.112.1.11.1

```
snmpwalk -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.112.1.11.1
```

Example of viewing the dhcp snooping table

CLI command:
```
Show ip dhcp snooping binding
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.112.1.11.1
```

Enabling/disabling DHCP/DHCPv6 snooping globally

MIB: rlbridge-security.mib

Tables used: rlIpDhcpSnoopEnable — 1.3.6.1.4.1.89.112.1.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.112.1.2.0 i {enable(1), disable(2)}
```

Example of enabling global dhcp snooping

CLI command:
```
ip dhcp snooping
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.1.2.0 i 1
```

Enabling/disabling dhcp snooping in vlan

MIB: rlbridge-security.mib

Tables used:: rlIpDhcpSnoopEnableVlanTable — 1.3.6.1.4.1.89.112.1.12

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.112.1.12.1.2.{vlan id} i {createAndGo(4), destroy(6)}
```

Example of adding dhcp snooping to vlan 622

CLI command:
```
ip dhcp snooping vlan 622
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.1.12.1.2.622 i 4
```

Configuring IP DHCP information option

MIB: rlbridgesecurity.mib

Tables used: rlIpDhcpOpt82InsertionEnable — 1.3.6.1.4.1.89.112.1.8

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.112.1.8.0 i {enable(1), disable(2)}
```

Example

CLI command:
```
ip dhcp information option
```
SNMP command:
```
snmpset -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.112.1.8.0 i 1
```

Configuring DHCP trusted port

MIB: rlbridge-security.mib

Tables used: rlIpDhcpSnoopTrustedPortTable — 1.3.6.1.4.1.89.112.1.13

```
snmpset -v2c -c <community> <IP address>
 1.3.6.1.4.1.89.112.1.13.1.2.{ifIndex} i {createAndGo(4), destroy(6)}
```

Example of configuring the TenGigabitEthernet 1/0/23 trusted interface

CLI command:
```
interface TenGigabitethernet 1/0/23
ip dhcp snooping trust
```

SNMP command:
```
snmpset -v2c -c private  192.168.1.30 \
1.3.6.1.4.1.89.112.1.13.1.2.23 i 4
```

Configuring DHCP relay on VLAN

MIB: rldhcp.mib

Tables used:
rlDhcpRelayInterfaceListVlanId1To1024 — 1.3.6.1.4.1.89.38.29.1.3
rlDhcpRelayInterfaceListVlanId1025To2048 — 1.3.6.1.4.1.89.38.29.1.4
rlDhcpRelayInterfaceListVlanId2049To3072 — 1.3.6.1.4.1.89.38.29.1.5
rlDhcpRelayInterfaceListVlanId3073To4094 — 1.3.6.1.4.1.89.38.29.1.6

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.38.29.1.3.1 x {bit mask}
```

Example of configuring IP DHCP relay enable on vlan 1

CLI command:
```
Interface vlan 1
ip dhcp relay enable
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.38.29.1.3.1 x 800000000000
```

Example of configuring IP DHCP relay enable on vlan 1026

CLI command:
```
Interface vlan 1026
ip dhcp relay enable
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.38.29.1.4.1 x 400000000000
```

An example of calculating a bit mask can be found in section "APPENDIX A. Bit mask calculation method".

15.3 *IP-source Guard*

Enabling/disabling ip source guard globally

MIB: rlbridge-security.mib

Tables used: rlIpSourceGuardEnable — 1.3.6.1.4.1.89.112.2.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.112.2.2.0 i {enable(1), disable(2)}
```

Example of enabling ip source guard globally

CLI command:
```
ip source-guard
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.2.2.0 i 1
```

Creating ip source guard static bind

MIB: rlbridge-security.mib

Tables used: rlIpDhcpSnoopStaticTable — 1.3.6.1.4.1.89.112.1.10

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.112.1.10.1.3.{vlan id}.{MAC in DEC. Each MAC address byte is
separated from a previous one by a point} a {ip address (DEC)} \
 1.3.6.1.4.1.89.112.1.10.1.4.{vlan id}.{MAC in DEC. Each MAC address byte is
separated from a previous one by a point} i {ifIndex} \
 1.3.6.1.4.1.89.112.1.10.1.5.{vlan id}.{MAC in DEC. Each MAC address byte is
separated from a previous one by a point} i {createAndGo(4), destroy(6)}
```

Example of binding MAC address 00:11:22:33:44:55 to IP 192.168.1.34, vlan 622, TenGigabitEthernet 1/0/23 interface

CLI command:
```
ip source-guard binding 00:11:22:33:44:55 622 192.168.1.34 TenGigabitEthernet
1/0/23
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.1.10.1.3.622.0.17.34.51.68.85 a 192.168.1.34 \
1.3.6.1.4.1.89.112.1.10.1.4.622.0.17.34.51.68.85 i 23 \
1.3.6.1.4.1.89.112.1.10.1.5.622.0.17.34.51.68.85 i 4
```

Enabling/disabling ip source guard on the port

MIB: rlbridge-security.mib

Tables used: rlIpSourceGuardPortTable — 1.3.6.1.4.1.89.112.2.5

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.112.2.5.1.2.<ifIndex> i {createAndGo(4), destroy(6)}
```

Example of enabling ip source guard on the TenGigabitEthernet 1/0/23 interface

CLI command:
```
interface TenGigabitethernet 1/0/23
 ip source-guard
```

SNMP command:
```
snmpset -v2c -c private  192.168.1.30 \
1.3.6.1.4.1.89.112.2.5.1.2.23 i 4
```

15.4 *ARP Inspection*

Enabling/disabling ARP Inspection globally

MIB: rlbridge-security.mib

Tables used: rlIpArpInspectEnable — 1.3.6.1.4.1.89.112.3.2

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.112.3.2.0 i {enable(1), disable (2)}
```

Example of enabling arp inspection globally

CLI command:
```
ip arp inspection
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.3.2.0 i 1
```

Enabling/disabling ARP Inspection in VLAN

MIB: rlbridge-security.mib

Tables used: rlIpArpInspectEnableVlanTable — 1.3.6.1.4.1.89.112.3.6

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.112.3.6.1.3.{vlan id} i {createAndGo(4), destroy(6)}
```

Example of enabling arp inspection in vlan 622

CLI command:
```
ip arp inspection vlan 622
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.3.6.1.3.622 i 4
```

Configuring ARP Inspection trusted port

MIB: rlbridge-security.mib

Tables used: rlIpArpInspectTrustedPortRowStatus — 1.3.6.1.4.1.89.112.3.7.1.2

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.112.3.7.1.2.{ifIndex} i {createAndGo(4), destroy(6)}
```

Example of configuring the TenGigabitEthernet 1/0/23 trusted interface

CLI command:
```
interface TenGigabitethernet 1/0/23
 ip arp inspection trust
```
SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.3.7.1.2.23 i 4
```

Binding ip arp inspection to vlan

MIB: rlbridge-security.mib

Tables used: rlIpArpInspectAssignedListName — 1.3.6.1.4.1.89.112.3.6.1.2

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.112.3.6.1.2.{vlan id} s {list name}
```

Example of binding the test list to vlan 622

CLI command:
```
ip arp inspection list assign 100 test
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.3.6.1.2.622 s test
```

15.5 *Port-based client authentication (802.1x)*

Enabling 802.1x authentication on the switch

MIB: dot1xPaeSystem.mib

Tables used: dot1xPaeSystemAuthControl — 1.0.8802.1.1.1.1.1.1

```
snmpset -v2c -c <community> <IP address> \
1.0.8802.1.1.1.1.1.1.0 i {enabled(1), disabled(2)}
```

Example of enabling 802.1x

CLI Command:
```
dot1x system-auth-control
```

SNMP command:
```
snmpset -v2c -c  private  192.168.1.30 \
1.0.8802.1.1.1.1.1.1.0 i 1
```

Enabling periodic re-authentication of the client

MIB: draft-ietf-bridge-8021x.mib

Tables used: dot1xAuthReAuthEnabled — 1.0.8802.1.1.1.1.2.1.1.13

```
snmpset -v2c -c <community> <IP address> \
1.0.8802.1.1.1.1.2.1.1.13.{ifIndex} i {true(1), false(2)}
```

Example of enabling periodic client re-authentication on the TenGigabitEthernet 1/0/23 interface

CLI command:
```
interface tengigabitethernet 1/0/23
dot1x reauthentication
```

SNMP command:
```
snmpset -v2c -c  private  192.168.1.30 \
1.0.8802.1.1.1.1.2.1.1.13.23 i 1
```

Period between re-authentications configuration

MIB: draft-ietf-bridge-8021x.mib

Tables used: dot1xAuthConfigTable — 1.0.8802.1.1.1.1.2.1.1.12

```
snmpset -v2c -c <community> <IP address> \
1.0.8802.1.1.1.1.2.1.1.12.{ifIndex} u {size 300-4294967295}
```

Example of setting a period of 300 seconds between repeated checks on the TenGigabitEthernet 1/0/23 interface

CLI command:
```
interface tengigabitethernet 1/0/23
dot1x timeout reauth-period 300
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.0.8802.1.1.1.1.2.1.1.12.23 u 300
```

Configuring 802.1x authentication modes on the interface

MIB: draft-ietf-bridge-8021x.mib

Tables used: dot1xAuthConfigTable — 1.0.8802.1.1.1.1.2.1.1.6

```
snmpset -v2c -c <community> <IP address> \
1.0.8802.1.1.1.1.2.1.1.6.{ifIndex}  i  {force-Unauthorized(1),  auto(2),  force-
Authorized(3)}
```

Example of 802.1x authentication configuring in auto mode on the TenGigabitEthernet 1/0/23 interface

CLI command:
```
interface tengigabitethernet 1/0/23
dot1x port-control auto
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.0.8802.1.1.1.1.2.1.1.6.23 i 2
```

Enabling authentication based on users' MAC addresses

MIB: radlan-dot1x-mib.mib

Tables used: rldot1xAuthenticationPortTable — 1.3.6.1.4.1.89.95.10.1.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.95.10.1.1.{ifIndex}   i   {destroy(1),   mac-and-802.1x(2),   mac-
only(3)}
```

Example of enabling MAC-based authentication on the TenGigabitEthernet 1/0/23 interface

CLI command:
```
interface tengigabitethernet 1/0/23
dot1x authentication mac
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.95.10.1.1.23 i 3
```

Allowing the presence of one or more clients on an authorized 802.1x port

MIB: rlinterfaces.mib

Tables used: swIfTable — 1.3.6.1.4.1.89.43.1.1.30

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.43.1.1.30.{ifIndex} i {single(1), none(2), multi-sessions(3)}
```

Example of multiple client permissions on the TenGigabitEthernet 1/0/23 interface

CLI command:
```
interface TenGigabitethernet 1/0/23
dot1x host-mode multi-sessions
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.30.23 i 3
```

Enabling one or two authentication, authorization and accounting (AAA) methods for use on IEEE 802.1x interfaces

MIB: rlAAA.mib

Tables used: rlAAAEapMethodListTable — 1.3.6.1.4.1.89.97.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.97.1.1.1.7.{“default” in DEC, each letter is separated from the
next one by a point} s {authentication list}
\1.3.6.1.4.1.89.97.1.1.2.7.{“default” in DEC, each letter is separated from the
next one by a point} i {Deny(0), radius(1), none(2)} \
1.3.6.1.4.1.89.97.1.1.3.7.{“default” in DEC, each letter is separated from the
next one by a point} i {Deny(0), radius(1), none(2)} \
1.3.6.1.4.1.89.97.1.1.7.7.{“default” in DEC, each letter is separated from the
next one by a point} i 1
```

Example of enabling RADIUS server list for user authentication

CLI Command:
```
aaa authentication dot1x default radius none
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.97.1.1.1.7.100.101.102.97.117.108.116 s default \
1.3.6.1.4.1.89.97.1.1.2.7.100.101.102.97.117.108.116 i 1 \
1.3.6.1.4.1.89.97.1.1.3.7.100.101.102.97.117.108.116 i 2 \
1.3.6.1.4.1.89.97.1.1.7.7.100.101.102.97.117.108.116 i 1
```

1) To return default settings, change the values to Deny(0).

2) Default is converted from ASCII to HEX using a table, which can be found at https://ru.wikipedia.org/wiki/ASCII.

Adding a specified server to a list of used RADIUS servers

MIB: rlAAA.mib

Tables used: rlRadiusServerInetTable — 1.3.6.1.4.1.89.80.8

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.80.8.1.2.1.4.{ip address (DEC)}.{default UDP port 1812}.{default
UDP port 1813} x "{ip adress(HEX)}" \
1.3.6.1.4.1.89.80.8.1.1.1.4.{ip address (DEC)}.{default UDP port 1812}.{default
UDP port 1813} i {ipv4(1), ipv6(2), ipv4z(3)} \
1.3.6.1.4.1.89.80.8.1.3.1.4.{ip address(DEC)}.{default UDP port 1812}.{default
UDP port 1813} i {default UDP port 1812} \
1.3.6.1.4.1.89.80.8.1.4.1.4.{ip address(DEC)}.{default UDP port 1812}.{default
UDP port 1813} i {default UDP port 1813} \
1.3.6.1.4.1.89.80.8.1.9.1.4.{ip address (DEC)}.{default UDP port 1812}.{default
UDP port 1813} s "#{encoding key}" \
1.3.6.1.4.1.89.80.8.1.13.1.4.{ip address (DEC)}.{default UDP port 1812}.{default
UDP port 1813} i {createAndGo(4), destroy(6)}
```

Example

CLI Command:

```
radius-server host 192.168.1.10 encrypted key da90833f59be
```

SNMP command:

```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.89.80.8.1.2.1.4.192.168.1.10.1812.1813 x "c0a8010a" \
1.3.6.1.4.1.89.80.8.1.1.1.4.192.168.1.10.1812.1813 i 1 \
1.3.6.1.4.1.89.80.8.1.3.1.4.192.168.1.10.1812.1813 i 1812 \
1.3.6.1.4.1.89.80.8.1.4.1.4.192.168.1.10.1812.1813 i 1813 \
1.3.6.1.4.1.89.80.8.1.9.1.4.192.168.1.10.1812.1813 s "#da90833f59be" \
1.3.6.1.4.1.89.80.8.1.13.1.4.192.168.1.10.1812.1813 i 4
```

15.6 *Loopback detection mechanism*

Global enabling of loopback-detection

MIB: rllbd.mib

Tables used: rlLbdEnable — 1.3.6.1.4.1.89.127.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.127.1.0 i { true(1), false(2) }
```

Example of global enabling of loopback-detection

CLI Command:

```
loopback-detection enable
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.127.1.0 i 1
```

Changing the loopback-detection interval

MIB: rllbd.mib

Tables used: rlLbdDetectionInterval — 1.3.6.1.4.1.89.127.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.127.2.0 I { seconds 1-60 }
```

Example of changing loopback frames for 23 seconds

CLI command:
```
loopback-detection interval 23
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.127.2.0 i 23
```

Changing loopback-detection operation mode

MIB: rllbd.mib

Tables used: rlLbdMode — 1.3.6.1.4.1.89.127.3

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.127.3.0  i  {source-mac-addr(1),base-mac-addr(2),  multicast-mac-
addr(3),broadcast-mac-addr (4) }
```

Example of changing loopback operation mode to source-mac-addr

CLI command:
```
loopback-detection mode src-mac-addr
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.127.3.0 i 1
```

Enabling/disabling loopback-detection on interfaces

MIB: rllbd.mib

Tables used: rlLbdPortAdminStatus — 1.3.6.1.4.1.89.127.4.1.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.127.4.1.1.{ifindex} i { enable(1), disable(2)}
```

Example of enabling loopback detection on the TenGigabitethernet 1/0/23 interface

CLI command:
```
interface TenGigabitethernet 1/0/23
loopback-detection enable
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.127.4.1.1.23 i 1
```

Viewing loopback-detection operation status on an interface

MIB: rllbd.mib

Tables used: rlLbdPortOperStatus — 1.3.6.1.4.1.89.127.4.1.2

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.127.4.1.2.{ifindex}
```

Example of viewing loopback-detection status on the TenGigabitethernet 1/0/23 interface

CLI command:
```
show loopback-detection TenGigabitethernet 1/0/23
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.127.4.1.2.23
```

When using the SNMP command:

1 — inactive state,
2 — active state,
3 — loopdetected.

Viewing blocked VLANs in the vlan-based mode

MIB: rllbd.mib

Tables used: eltMesLdb — 1.3.6.1.4.1.35265.1.23.127

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.127.4.1.3.{ifindex}.{vlan}
```

Example of viewing the status of vlan 2 on the TenGigabitethernet port 1/0/23

CLI command:
```
show loopback-detection TenGigabitethernet 1/0/2
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.127.4.1.3.23.622
```

Possible states:

1 — active,
2 — blocked.

15.7 *Broadcast storm control (storm-control)*

Configuring storm-control on an interface

MIB: RADLAN-MIB

Tables used: rlStormCtrl **—** 1.3.6.1.4.1.89.77

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.77.12.1.2.{ifindex}.{broadcast(1),multicastRegistred(2),multicas
tUnregistred(3), multicastAll(4), unknownUnicast(5)} u {rate} \
1.3.6.1.4.1.89.77.12.1.3.{ifindex}.{broadcast(1),multicastRegistred
(2),multicastUnregistred(3),multicastAll(4),unknownUnicast(5)} I
kiloBitsPerSecond(1),precentaged(2)} \
1.3.6.1.4.1.89.77.12.1.4.{ifindex}.{broadcast(1),multicastRegistred
(2),multicastUnregistred(3), multicastAll(4), unknownUnicast(5)}  i
{none(1),trap(2),shutdown(3),trapAndShutdown(4)}
```

Example of enabling storm-control for broadcast traffic on the TenGigabitethernet 1/0/23 interface

CLI command:

```
interface TenGigabitethernet 1/0/23
storm-control broadcast kbps 10000 trap shutdown
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.12.1.3.23.1 i 1 \
1.3.6.1.4.1.89.77.12.1.2.23.1 u 1000 \
1.3.6.1.4.1.89.77.12.1.4.23.1 i
```

Example of disabling storm-control for broadcast traffic on the TenGigabitethernet 1/0/23 interface

CLI command:

```
interface TenGigabitethernet 1/0/23
 no storm-control broadcast
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.12.1.2.23.1 u 0
```

Enabling/disabling storm-control for unknown unicast-traffic

MIB: radlan-stormctrl.mib

Tables used: rlStormCtrlRateLimCfgTable — 1.3.6.1.4.1.89.77.12

```
snmpset -v2c -c <community> <IP address> \
iso.3.6.1.4.1.89.77.12.1.2.{ifIndex}.5 u {Kbps,disable (0)}
```

Example of enabling control of unknown unicast traffic up to 50 kbps

CLI command:

```
interface TenGigabitethernet 1/0/23
storm-control unicast Kbps 50
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.12.1.2.23.5 u 50
```

16 IP AND MAC ACL CONFIGURATION

Creating a mac access-list

MIB: qosclimib.mib

Tables used: rlQosAclTable — 1.3.6.1.4.1.89.88.7

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.7.1.2.{index-of-acl} s "{name-of-acl}" \
1.3.6.1.4.1.89.88.7.1.3.{index-of-acl} i {type-of-acl: mac(1), ip (2)} \
1.3.6.1.4.1.89.88.7.1.4.{index-of-acl} i {createAndGo(4), destroy(6)}
```

Example of creating MAC ACL with index 207

CLI command:
```
mac access-list extended 7-mac
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.7.1.2.207 s "7-mac" \
1.3.6.1.4.1.89.88.7.1.3.207 i 1 \
1.3.6.1.4.1.89.88.7.1.4.207 i 4
```

Creating an ip access-list (ACL)

MIB: qosclimib.mib

Tables used: rlQosAclTable — 1.3.6.1.4.1.89.88.7

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.7.1.2.{index-of-acl} s "{name-of-acl}" \
1.3.6.1.4.1.89.88.7.1.3.{index-of-acl} i {type-of-acl: mac(1), ip (2)} \
1.3.6.1.4.1.89.88.7.1.4.{index-of-acl} i {createAndGo(4), destroy(6)}
```

Example of creating IP ACL with index 107

CLI command:
```
ip access-list extended 7-ip
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.7.1.2.107 s "7-ip"  \
1.3.6.1.4.1.89.88.7.1.3.107 i 2 \
1.3.6.1.4.1.89.88.7.1.4.107 i 4
```

Example of filling ACL with rules is described in detail in section "Appendix B: Example of creating a standard IP ACL**".**

Binding IP or MAC ACL to a port

MIB: qosclimib.mib

Tables used:
rlQosIfAclIn — 1.3.6.1.4.1.89.88.13.1.14
rlQosIfPolicyMapStatus — 1.3.6.1.4.1.89.88.13.1.13

```
 snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.13.1.14.{ifIndex}.2 i {Index-of-acl} \
1.3.6.1.4.1.89.88.13.1.13.{ifIndex}.2 i 1
```

Example of assigning a rule with index 107 (name ACL 7-ip) to the TenGigabitEthernet 1/0/23 port

CLI command:
```
interface TenGigabitethernet 1/0/23
 service-acl input 7-ip
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.13.1.14.23.2 i 107 \
1.3.6.1.4.1.89.88.13.1.13.23.2 i 1
```

To remove ACL from the port, the ACL index should be replaced by 0.
snmpset -c -v2c private 192.168.1.301.3.6.1.4.1.89.88.13.1.14.50.2 i 0 1.3.6.1.4.1.89.88.13.1.13.50.2 i 1

Binding IP and MAC ACL to a port

MIB: qosclimib.mib

Tables used:
rlQosIfAclIn — 1.3.6.1.4.1.89.88.13.1.14
rlQosIfIpv6AclIn — 1.3.6.1.4.1.89.88.13.1.201.3.6.1.4.1.89.88.13.1.20
rlQosIfPolicyMapStatus — 1.3.6.1.4.1.89.88.13.1.13

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.13.1.14.{Ifindex}.2 i {Index-of-mac-acl} \
 1.3.6.1.4.1.89.88.13.1.20.{Ifindex}.2 i {Index-of-ip-acl} \
 1.3.6.1.4.1.89.88.13.1.13.{ifIndex}.2 i 1
```

Example of assigning a rule with an index 107 and 207 (name ACL 7-ip for IP ACL and 7-mac for MAC ACL) to the TenGigabitEthernet 1/0/23 port (Ifindex 23)

CLI command:
```
interface TenGigabitethernet 1/0/23
 service-acl input 7-mac 7-ip
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.13.1.14.23.2 i 207 \
1.3.6.1.4.1.89.88.13.1.20.23.2 i 107 \
1.3.6.1.4.1.89.88.13.1.13.23.2 i 1
```

To remove ACL from the port, the IP index and MAC ACL should be replaced by 0.

**snmpset -v2c -c private 192.168.1.30 **
**1.3.6.1.4.1.89.88.13.1.14.23.2 i 0 **
**1.3.6.1.4.1.89.88.13.1.20.23.2 i 0 **
1.3.6.1.4.1.89.88.13.1.13.23.2 i 1

Creating a policy-map and binding an ACL to it

MIB: qosclimib.mib

Tables used:
rlQosClassMapTable — 1.3.6.1.4.1.89.88.9
rlQosPolicyMapTable — 1.3.6.1.4.1.89.88.11
rlQosPolicyClassPriorityRefTable — 1.3.6.1.4.1.89.88.39

Scheme: the creation of a policy-map is done in several queries

1. Create a class and assign properties to it

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.9.1.2.{index-of-class} s "{name-of-class-map}" \
1.3.6.1.4.1.89.88.9.1.3.{index-of-class} i {matchAll (1)} \
1.3.6.1.4.1.89.88.9.1.7.{index-of-class} i {index-of-acl} \
1.3.6.1.4.1.89.88.9.1.9.{index-of-class} i {Mark vlan disable (1), enable(2)} \
1.3.6.1.4.1.89.88.9.1.13.{index-of-class} i {create and go(4),destroy(6)}
```

2. Create a policy-map and enable it

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.11.1.2.{index-of-policy-map} s {name-of-policy-map} \
1.3.6.1.4.1.89.88.11.1.3.{index-of-policy-map} i {createAndGo(4), destroy(6)}
```

3. Binding class-map to policy-map

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.39.1.2.1.{index-of-class} i {index-of-class} \
1.3.6.1.4.1.89.88.39.1.3.1.{index-of-class} i {index-of-policy-map}
```

4. Creating a speed limit for class-map

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.10.1.2.{Number-of-class-in-policy} s {Policer-cm-20} \
1.3.6.1.4.1.89.88.10.1.3.{Number-of-class-in-policy} i {single(1),
aggregate(2)} \
1.3.6.1.4.1.89.88.10.1.4.{Number-of-class-in-policy} i {rate} \
1.3.6.1.4.1.89.88.10.1.5.{Number-of-class-in-policy} i {burst} \
1.3.6.1.4.1.89.88.10.1.6.{Number-of-class-in-policy} i {none(1), drop(2),
remark(3)} \
1.3.6.1.4.1.89.88.10.1.8.{Number-of-class-in-policy} i {createAndGo(4),
destroy(6)}
```

5. Binding the speed limit to class-map

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.9.1.6.{index-of-class} i {Number-of-class-in-policy}
```

6. Set the label to DSCP, cos traffic or specify the output queue
1.3.6.1.4.1.89.88.233

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.9.1.4.{index-of-class} i {setDSCP(3), setQueue(4), setCos(5)}
\
1.3.6.1.4.1.89.88.9.1.5.{index-of-class} i {Mark value of DSCP/queue/cos(DEC)}
```

Example: IP ACL with index-of-acl = 107 is binding to a class-map named test and labeling DSCP = 36(DEC), cos = 4 and queue = 8 for traffic covered by IP ACL. Class test is binding to a policy-map named test1.

CLI command:

```
qos advanced
 ip access-list extended 7-ip
 permit ip any any
exit

class-map test
 match access-group 7-ip
exit
 policy-map test1
 class test
 set dscp 36
 set queue 8
 set cos 4
 police 97000 524288 exceed-action drop
exit
exit
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.9.1.2.20 s "test" \
1.3.6.1.4.1.89.88.9.1.3.20 i 1 \
1.3.6.1.4.1.89.88.9.1.7.20 i 107 \
1.3.6.1.4.1.89.88.9.1.9.20 i 1 \
1.3.6.1.4.1.89.88.9.1.13.20 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.11.1.2.1 s "test1" \
1.3.6.1.4.1.89.88.11.1.3.1 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.39.1.2.1.20 i 20 \
1.3.6.1.4.1.89.88.39.1.3.1.20 i 1

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.10.1.2.1 s "Policer-cm-20" \
1.3.6.1.4.1.89.88.10.1.3.1 i 1 \
1.3.6.1.4.1.89.88.10.1.4.1 u 97000 \
1.3.6.1.4.1.89.88.10.1.5.1 u 524288 \
1.3.6.1.4.1.89.88.10.1.6.1 i 2 \
1.3.6.1.4.1.89.88.10.1.8.1 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.9.1.6.20 i 1

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.9.1.4.20 i 3 \
1.3.6.1.4.1.89.88.9.1.5.20 i 36

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.9.1.4.20 i 4 \
1.3.6.1.4.1.89.88.9.1.5.20 i 8

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.9.1.4.20 i 5 \
1.3.6.1.4.1.89.88.9.1.5.20 i 4
```

Assigning Policy-map to a port

MIB: qosclimib.mib

Tables used: rlQosIfPolicyMapPointerIn — 1.3.6.1.4.1.89.88.13.1.3

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.13.1.3.{Ifindex}.2 i {Index-of-policy-map}
```

Example of assigning a policy-map with index 1 to the TenGigabitEthernet 1/0/23 port

CLI command:

```
interface TenGigabitethernet 1/0/23
 service-policy input test1
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.13.1.3.23.2 i 1
```

17 CONFIGURING PROTECTION AGAINST DOS ATTACKS

Enabling security-suite

MIB: rlSecuritySuiteMib

Tables used: rlSecuritySuiteGlobalEnable — 1.3.6.1.4.1.89.120.1

```
snmpset -v2c -c <community> <IP address> 1.3.6.1.4.1.89.120.1.0 i {enable-
global-rules-only (1), enable-all-rules-types (2), disable (3)}
```

Example of enabling security-suite command class for all rules

CLI command:
```
security-suite enable
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.120.1.0 i 2
```

Configuring security suite operation mode

MIB: rlSecuritySuiteMib

Tables used: rlSecuritySuiteSynProtectionMode — 1.3.6.1.4.1.89.120.10

```
snmpset -v2c -c <community> <IP address> 1.3.6.1.4.1.89.120.10.0 i {disabled
(1), report (2), block (3)}
```

Example of enabling report operation mode

CLI command:
```
security-suite syn protection mode report
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.120.10.0 i 2
```

Deny protection against TCP packets with simultaneously set SYN and FIN

MIB: rlSecuritySuiteMib

Tables used: rlSecuritySuiteDenySynFinTcp — 1.3.6.1.4.1.89.120.9

```
snmpset -v2c -c <community> <IP address> 1.3.6.1.4.1.89.120.9.0 i {(deny (1),
permit (2)}
```

Example of enabling report operation mode

CLI command:
```
security-suite deny syn-fin
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.120.9.0 i 2
```

18 QUALITY OF SERVICE (QOS)

18.1 *QoS configuration*

Limiting uplink bandwidth on Ethernet ports

MIB: qosclimib.mib

Tables used: rlQosIfPolicyEntry — 1.3.6.1.4.1.89.88.13.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.13.1.6.{port ifindex}.2 i {disable(1),enable
(1)} \
 1.3.6.1.4.1.89.88.13.1.7.{port ifindex}.2 i {traffic-shape} \
 1.3.6.1.4.1.89.88.13.1.8.{port ifindex}.2 i {Burst size in bytes}
```

Example of limiting the outgoing speed on the port to a value of 20 Mbps

CLI command:

```
interface TenGigabitethernet 1/0/23
traffic-shape 20480 500000
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.13.1.6.23.2 i 2 \
1.3.6.1.4.1.89.88.13.1.7.23.2 i 20480 \
1.3.6.1.4.1.89.88.13.1.8.23.2 i 500000
```

Limiting downlink bandwidth on Ethernet ports

MIB: RADLAN-STORMCTRL-MIB

Tables used: rlStormCtrlRateLimCfgTable — 1.3.6.1.4.1.89.77.12

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.77.12.1.2.{ifIndex}.6 u {limit} \
1.3.6.1.4.1.89.77.12.1.5.{ifIndex}.6 u {Burst size in bytes}
```

Example of limiting the incoming speed on the TenGigabitEthernet 1/0/23 interface to a value of 10 Mbps

CLI command:

```
interface TenGigabitEthernet 1/0/23
rate-limit 10240 burst 500000
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.12.1.2.23.6 u 10240 \
1.3.6.1.4.1.89.77.12.1.5.23.6 u 500000
```

To disable rate-limit on an interface, the following must be done (on the example of the TenGigabitethernet 1/0/23 interface):

snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.77.12.1.2.23.6 u 0 1.3.6.1.4.1.89.77.12.1.5.23.6 u 128000

Creating a qos tail-drop profile and expanding queue descriptors

MIB: eltQosTailDropMIB.mib

Tables used: eltQosTailDropProfileQueueTable — 1.3.6.1.4.1.35265.1.23.12.1.1.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.12.1.1.1.1.4.{Profile number (1-4)}.{queue number(1-8)}
i {size (0-11480)}
```

Example

CLI command:
```
qos tail-drop profile 2
queue 1 limit 900
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.12.1.1.1.1.4.2.1 i 900
```

To return to default settings, set the value to 12.

Setting the size of the packet separable pool for the port

MIB: eltQosTailDropMIB.mib

Tables used: eltQosTailDropProfileTable — 1.3.6.1.4.1.35265.1.23.12.1.1.4

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.12.1.1.4.1.2{profile number (1-4)} i {size (0-11480)}
```

Example

CLI command:
```
qos tail-drop profile 2
port-limit 900
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.12.1.1.4.1.2.2 i 900
```

Assigning a created profile to an interface

MIB: eltQosTailDropMIB.mib

Tables used: eltQosTailDropIfConfigTable — 1.3.6.1.4.1.35265.1.23.12.1.1.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.12.1.1.2.1.1.{IfIndex} i {profile number (1-4)}
```

Example

CLI command:
```
interface TenGigabitethernet 1/0/23
qos tail-drop profile 2
```

SNMP command:
```
snmpset -v2c -c  private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.12.1.1.2.1.1.23 i 2
```

View display of global limits, descriptors, buffers

MIB: ELTEX-MES-QOS-TAIL-DROP-MIB

Tables used: eltQosTailDropConfigTable — 1.3.6.1.4.1.35265.1.23.12.1.1.3

```
snmpwalk -v2c -c <community> <ip address> \
1.3.6.1.4.1.35265.1.23.12.1.1.3
```

Example

CLI command:
```
show qos tail-drop
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.12.1.1.3
```

View output table of current allocated qos resources (limits, descriptors, buffers)

MIB: ELTEX-MES-QOS-TAIL-DROP-MIB

Tables used: eltQosTailDropStatusTable — 1.3.6.1.4.1.35265.1.23.12.1.2.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.12.1.2.1
```

Example

CLI command:
```
show qos tail-drop
```
SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.12.1.2.1
```

Viewing Tail Drop counters per queue

MIB: RADLAN-COPY-MIB

Tables used: eltMesCountersMIB — 1.3.6.1.4.1.35265.1.23.1.8

```
snmpwalk -v2c -c <community>  <IP address> \
1.3.6.1.4.1.35265.1.23.1.8.1.2.1.1.1.7.{ifIndex}.{1-8}.0
```

Example of viewing counters on the first queue

CLI command:
```
show interface TenGigabitethernet 1/0/23
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.8.1.2.1.1.1.7.23.1.0
```

18.2 *QoS statistics*

Enabling/disabling QoS statistics

MIB: qosclimib.mib

Tables used: eltCountersQosStatisticsEnable — 1.3.6.1.4.1.35265.1.23.1.8.1.1.1.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.1.8.1.1.1.1.0 i {enable(1), disable(2)}
```

Example of setting up QoS statistics

CLI command:
```
qos statistics interface
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.8.1.1.1.1.0 i 1
```

Viewing QoS statistics counters

MIB: qosclimib.mib

Tables used: rlInterfaceQueueStatisticsTxPackets — 1.3.6.1.4.1.89.233.2.1.4

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.233.2.1.{Counter number}.{ifIndex}.{Queue number}
```

Example of taking readings of the TxPackets counter on 4 queues of the TenGigabitEthernet interface 1/0/23

CLI command:
```
show qos statistics interface
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.233.2.1.4.23.4
```

Possible counter numbers:

1. **All counters ( )**
2. **Queue counter(2)**
3. **txpackets counter(4)**
4. **TxBytes counter(5)**
5. **droppedpackets counter(6)**
6. **DroppedBytes counter(7)**

Example of clearing QoS statistics counters

MIB: qosclimib.mib

Tables used: rlInterfaceQueueStatisticsClear — 1.3.6.1.4.1.89.233.1.0

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.233.1.0 i 1
```

Example

CLI command:
```
clear qos statistics
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.233.1.0 i 1
```

19 ROUTING

19.1 *Static routing*

Viewing routing table

MIB: IP-FORWARD-MIB

Tables used: ipCidrRouteTable — 1.3.6.1.2.1.4.24.4

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.2.1.4.24.4
```

Example

CLI command:
```
show ip route
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.4.24.4
```

Viewing static routes

MIB: rlip.mib

Tables used: rlIpStaticRouteTable — 1.3.6.1.4.1.89.26.17.1

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.26.17.1
```

Example

CLI command:
```
show running-config routing
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.26.17.1
```

19.2 *Dynamic routing*

Viewing OSPF neighbourhood

MIB: rlip.mib

Tables used: rlOspfNbrTable — 1.3.6.1.4.1.89.210.11

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.210.11
```

Example

CLI command:
```
show ip ospf neighbor
```

SNMP command:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.210.11
```

20 VXLAN CONFIGURATION

Creating VXLAN instance

MIB: ELTEX-EVPN-MIB

Tables used: eltexEvpnVxlanTable — 1.3.6.1.4.1.35265.56.1.1.1, eltexEvpnVxlanFirstFreeIndex - 1.3.6.1.4.1.35265.56.1.1.3

```
snmpget -v2c -c <community> <IP address> 1.3.6.1.4.1.35265.56.1.1.3
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.56.1.1.1.1.3.{index} i { adminStatusUp(1),
adminStatusDown(2) } \
1.3.6.1.4.1.35265.56.1.1.1.1.4.{index} i { vni } \
1.3.6.1.4.1.35265.56.1.1.1.1.5.{index} i { vlan }  \
1.3.6.1.4.1.35265.56.1.1.1.1.6.{index} s { vxlan_name } \
1.3.6.1.4.1.35265.56.1.1.1.1.2.{index} i 4
```

Example

CLI command:
```
vxlan VX105
vni 10105
vlan 105
exit
```

SNMP command:
```
snmpget -v2c -c private 192.168.1.30 1.3.6.1.4.1.35265.56.1.1.3
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.56.1.1.1.1.3.4 i 1 \
1.3.6.1.4.1.35265.56.1.1.1.1.4.4 i 10105 \
1.3.6.1.4.1.35265.56.1.1.1.1.5.4 i 105 \
1.3.6.1.4.1.35265.56.1.1.1.1.6.4 s "VX105" \
1.3.6.1.4.1.35265.56.1.1.1.1.2.4 i 4
```

First, get the number of the first free index, and then use it to create a VXLAN instance.

Deleting VXLAN instance

MIB: ELTEX-EVPN-MIB

Tables used: eltexEvpnVxlanTable — 1.3.6.1.4.1.35265.56.1.1.1

```
snmpset -v2c -c <community> <IP address>
1.3.6.1.4.1.35265.56.1.1.1.1.2.{index} i 6
```

Example of deleting VXLAN instance

CLI command:
```
no vxlan VX105
```

SNMP command:
```
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.35265.56.1.1.1.1.2.4 i 6
```

Viewing VXLAN instances

MIB: ELTEX-EVPN-MIB

Tables used: eltexEvpnVxlanTable — 1.3.6.1.4.1.35265.56.1.1.1

```
snmpwalk -v2c -c <community> <IP address>
1.3.6.1.4.1.35265.56.1.1.1.1
```

Example of viewing VXLAN instances

SNMP command:
```
snmpwalk -v2c -c private 192.168.1.30 1.3.6.1.4.1.35265.56.1.1.1.1
```

APPENDIX A. BIT MASK CALCULATION METHOD

Bit masks consist of 128 bytes (256 hexadecimal digits in total).
Each digit represents four VLANS/ports. The required field shall be determined by the VLAN/port number.

Example 1

Write a bitmask for the TenGigabitEthernet 2/0/16-17 interfaces:

- for 1G interfaces, ifIndex starts with 1;
- for te2/0/16 port, ifIndex is 68, for te2/0/17 — 69.

Determination of the digit number:

68/4=17 69/4=17.25 (Each digit is responsible for 4 ifIndex. When dividing ifindex by 4 to determine the digit number for an entry, the resulting value is rounded up).

If we need te2/0/16-17 (ifindex 68,69) ports, then they should be written in th 17 and 18 fields.

In binary sequence, the field 17 will be written as 0001 (last 1 — index 68). When translating to HEX, we get 1.

In binary sequence, the field 18 will be written as 1000 (first 1 — index 69). When translating to HEX, we get 8.

the bit mask will be 16 zeros, 1, 8: 000000000000000018.

Example 2

Write a bit mask for vlan 622, 3100.

- 622/4=155.5 (Each 0 is responsible for 4 vlans. When dividing vlan by 4 to determine the field number for the entry, rounding always goes up).
  If vlan 622 is needed, it should be put in the field 156.
  In binary sequence, the field 156 will be written as 10100 (second 1 — vlan 622). When translating to HEX, we get 4.
  The bit mask will be 155 zeros and 4:
  000000000000000000000000000000000000000000000000000000000000000000000000
  000000000000000000000000000000000000000000000000000000000000000000000000
  000000000004

- 3100/4=775
  It must be taken into account that tables are taken to indicate a VLAN number:
  rldot1qPortVlanStaticEgressList1to1024
  rldot1qPortVlanStaticEgressList1025to2048
  rldot1qPortVlanStaticEgressList2049to3072
  rldot1qPortVlanStaticEgressList3073to4094
  As the vlan falls into Table 4, it is 775-256*3=7.
  Vlan 3100 will be written to the field 7 of the table.
  In binary sequence, the field 7 will be written as 0001 (last 1 — vlan 3100). When translating to HEX, we get 1.
  The bit mask will be 6 zeros and 1: 0000001.

APPENDIX B: EXAMPLE OF CREATING A STANDARD IP ACL

This appendix describes an example of filling an IP ACL with index-of-acl = 107 with the following rules:

```
ip access-list extended 7-ip
 deny udp any bootps any bootpc ace-priority 20
 permit igmp any any ace-priority 40
 deny ip any 224.0.0.0 15.255.255.255 ace-priority 60
 permit ip 37.193.119.7 0.0.0.0 any ace-priority 80
 permit ip 10.130.8.3 0.0.0.0 any ace-priority 100
 permit ip 192.168.0.0 0.0.0.15 any ace-priority 120
 permit ip 37.193.119.7 0.0.0.0 any ace-priority 140
exit
```

Creating a deny udp any bootps any bootpc rule

MIB: qosclimib.mib

Tables used:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Scheme: the rule is created in two requests

1. The parameters of the rule are set.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 1} i {protocol(1)}  \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 1} x {protocol index (HEX)} \
 1.3.6.1.4.1.89.88.5.1.3.{value of field 1} i {Value in port table for protocol
= 0. Constant for this rule} \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 2} i {udp-port-src(6)} \
 1.3.6.1.4.1.89.88.5.1.3.{value of field 2} i {Number of source port (DEC)} \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 2} x {source ip(HEX)} \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 3} i { udp-port-dst(6)} \
 1.3.6.1.4.1.89.88.5.1.3.{value of field 3} i {Number of dst port (DEC)} \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 3} x {dst ip(HEX)}
```

2. Binding a rule by index-of-rule to an ACL by index-of-acl as deny.

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.31.1.3.{index-of-acl}.{index-of-rule} i {deny(2)} \
 1.3.6.1.4.1.89.88.31.1.4.{index-of-acl}.{index-of-rule} i {udp(3)} \
 1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {value of field 1} \
 1.3.6.1.4.1.89.88.31.1.7.{index-of-acl}.{index-of-rule} i {value of field 3} \
 1.3.6.1.4.1.89.88.31.1.9.{index-of-acl}.{index-of-rule} i {value of field 2}
```

Example of adding a deny udp any bootpc rule to IP ACL 7-ip (since the rule is assumed to be the first one, then index-of-rule=20)

CLI command:

```
ip access-list extended 7-ip
 deny udp any bootps any bootpc ace-priority 20
exit
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.1 i 1 \
```

```
1.3.6.1.4.1.89.88.5.1.4.1 x "0x11 FF" \
1.3.6.1.4.1.89.88.5.1.3.1 i 0 \
1.3.6.1.4.1.89.88.5.1.2.2 i 6 \
1.3.6.1.4.1.89.88.5.1.3.2 i 67 \
1.3.6.1.4.1.89.88.5.1.4.2 x "0x00 00" \
1.3.6.1.4.1.89.88.5.1.2.3 i 7 \
1.3.6.1.4.1.89.88.5.1.3.3 i 68 \
1.3.6.1.4.1.89.88.5.1.4.3 x "0x00 00"

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.1.20 i 2 \
1.3.6.1.4.1.89.88.31.1.4.1.20 i 3 \
1.3.6.1.4.1.89.88.31.1.5.1.20 i 1 \
1.3.6.1.4.1.89.88.31.1.7.1.20 i 2 \
1.3.6.1.4.1.89.88.31.1.9.1.20 i 3
```

Creating a permit igmp any any rule

MIB: qosclimib.mib

Tables used:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Scheme: a rule is created in two requests.

1. The parameters of the rule are set.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 4} i {protocol(1)} \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 4} x {protocol index (HEX)}
```

2. Binding a rule by index-of-rule to an ACL by index-of-acl as permit.
snmpset -v2c -c <community> <IP address> ****

```
 1.3.6.1.4.1.89.88.31.1.3.{index-of-acl}.{index-of-rule} i {permit (1)}  \
 1.3.6.1.4.1.89.88.31.1.4.{index-of-acl}.{index-of-rule} i {igmp (8)} \
 1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {value of field 4}
```

Example of adding a permit igmp any rule to IP ACL 7-ip (since the rule is assumed to be the second one, the index-of-rule=40)

CLI command:

```
ip access-list extended 7-ip
 permit igmp any any ace-priority 40
exit
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.4 i 1 \
1.3.6.1.4.1.89.88.5.1.4.4 x "0x02 FF"

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.1.40 i 1 \
1.3.6.1.4.1.89.88.31.1.4.1.40 i 8 \
1.3.6.1.4.1.89.88.31.1.5.1.40 i 4
```

Creating a deny ip any any any 224.0.0.0 15.255.255.255 rule

MIB: qosclimib.mib

Tables used:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Scheme: a rule is created in two requests.

1. The parameters of the rule are set.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 5} i {ip-dest(3)} \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 5} x {dst ip +wildcard mask (HEX)}
```

2. Binding a rule by index-of-rule to an ACL by index-of-acl as deny.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.31.1.3.{index-of-acl}.{index-of-rule} i {deny (2)} \
  1.3.6.1.4.1.89.88.31.1.4.{index-of-acl}.{index-of-rule} i {ip (1)} \
  1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {value of field 5}
```

Example of adding a deny ip any any any 224.0.0.0 15.255.255.255 rule to IP ACL 7-ip (since the rule is assumed to be the third one, then index-of-rule=60)

CLI command:

```
ip access-list extended 7-ip
deny ip any any any 224.0.0.0 15.255.255.255 ace-priority 60
exit
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.5 i 3 \
1.3.6.1.4.1.89.88.5.1.4.5 x "0xE0 00 00 00 0F FF FF FF"

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.1.60 i 2 \
1.3.6.1.4.1.89.88.31.1.4.1.60 i 1 \
1.3.6.1.4.1.89.88.31.1.5.1.60 i 5
```

Creating a permit ip any any 37.193.119.7 0.0.0.0 any rule

MIB: qosclimib.mib

Tables used: rlQosTupleTable — 1.3.6.1.4.1.89.88.5, rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Scheme: a rule is created in two requests.

1. The parameters of the rule are set.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 6} i {ip-source(2)} \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 6} x {source ip +wildcard mask (HEX)}
```

2. Binding a rule by index-of-rule to an ACL by index-of-acl as permit.

```
snmpset -v2c -c <community> <IP address> \
```

```
  1.3.6.1.4.1.89.88.31.1.3.{index-of-acl}.{index-of-rule} i {permit (1)} \
  1.3.6.1.4.1.89.88.31.1.4.{index-of-acl}.{index-of-rule} i {ip (1)} \
  1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {value of field 6}
```

Example of adding a permit ip 37.193.119.7 0.0.0.0 any to IP ACL 7-ip (since the rule is assumed to be the fourth one, the index-of-rule=80)

CLI command:

```
ip access-list extended 7-ip
permit ip 37.193.119.7 0.0.0.0 any ace-priority 80
exit
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.6 i 2 \
1.3.6.1.4.1.89.88.5.1.4.6 x "0x25 C1 77 07 00 00 00 00"

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.1.80 i 1 \
1.3.6.1.4.1.89.88.31.1.4.1.80 i 1 \
1.3.6.1.4.1.89.88.31.1.6.1.80 i 6
```

Creating a permit ip 10.130.8.3 0.0.0.0 any rule

MIB: qosclimib.mib

Tables used:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Scheme: a rule is created in two requests.

1. The parameters of the rule are set.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 7} i {ip-source(2)} \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 7} x {source ip +wildcard mask (HEX)}
```

2. Binding a rule by index-of-rule to an ACL by index-of-acl as permit

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.31.1.3.{index-of-acl}.{index-of-rule} i {permit (1)}  \
 1.3.6.1.4.1.89.88.31.1.4.{index-of-acl}.{index-of-rule} i {ip (1)} \
 1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {value of field 7}
```

Example of adding a permit ip 10.130.8.3 0.0.0.0 any to IP ACL 7-ip (since the rule is assumed to be the fifth one, the index-of-rule=100)

CLI command:

```
ip access-list extended 7-ip
 permit ip 10.130.8.3 0.0.0.0 any ace-priority 100
exit
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.7 i 2 \
1.3.6.1.4.1.89.88.5.1.4.7 x "0x0A 82 08 03 00 00 00 00"
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.1.100 i 1 \
```

```
1.3.6.1.4.1.89.88.31.1.4.1.100 i 1 \
1.3.6.1.4.1.89.88.31.1.6.1.100 i 7
```

Creating a permit ip any any 192.168.0.0 0.0.0.15 any rule

MIB: qosclimib.mib

Tables used:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Scheme: a rule is created in two requests.

1. The parameters of the rule are set.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 8} i {ip-source(2)} \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 8} x {source ip +wildcard mask (HEX)}
```

2. Binding a rule by index-of-rule to an ACL by index-of-acl as permit.

```
snmpset -v2c -c <community> <IP address> \
  1.3.6.1.4.1.89.88.31.1.3.{index-of-acl}.{index-of-rule} i {permit (1)}  \
  1.3.6.1.4.1.89.88.31.1.4.{index-of-acl}.{index-of-rule} i {ip (1)} \
  1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {value of field 8}
```

Example of adding a permit ip 192.168.0.0 0.0.0.15 any to IP ACL 7-ip (since the rule is assumed to be the sixth one, the index-of-rule=120)

CLI command:

```
ip access-list extended 7-ip
 permit ip 192.168.0.0 0.0.0.15 any ace-priority 120
exit
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.8 i 2 \
1.3.6.1.4.1.89.88.5.1.4.8 x "0xC0 A8 00 00 00 00 00 0F"

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.1.120 i 1 \
1.3.6.1.4.1.89.88.31.1.4.1.120 i 1 \
1.3.6.1.4.1.89.88.31.1.6.1.120 i 8
```

1. Binding a rule by index-of-rule to an ACL by index-of-acl as permit.

```
snmpset -v2c -c <community> <IP address> \
  1.3.6.1.4.1.89.88.31.1.3.{index-of-acl}.{index-of-rule} i {permit (1)}  \
  1.3.6.1.4.1.89.88.31.1.4.{index-of-acl}.{index-of-rule} i {ip (1)} \
  1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {value of field 9}
\
  1.3.6.1.4.1.89.88.31.1.6.{index-of-acl}.{index-of-rule} i {value of field 10}
```

APPENDIX C. EXAMPLE OF CREATING, FILLING AND REMOVING AN OFFSET-LIST WITH MAC ACL

This appendix describes an example of creating and filling a MAC ACL with index-of-acl = 207 with the following rules:

```
mac access-list extended 7-mac
offset-list PADO l2 12 00 88 l2 13 00 63 l2 15 00 07
deny any any offset-list PADO ace-priority 20
```

Creating a mac access-list

MIB: qosclimib.mib

Tables used: rlQosAclTable — 1.3.6.1.4.1.89.88.7

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.7.1.2.{index-of-acl} s "{name-of-acl}" \
1.3.6.1.4.1.89.88.7.1.3.{index-of-acl} i {type-of-acl: mac(1), ip (2)} \
1.3.6.1.4.1.89.88.7.1.4.{index-of-acl} i {createAndGo(4), destroy(6)}
```

Example of creating MAC ACL with index 207

CLI command:

```
mac access-list extended 7-mac
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.7.1.2.1 s "7-mac" \
1.3.6.1.4.1.89.88.7.1.3.1 i 1 \
1.3.6.1.4.1.89.88.7.1.4.1 i 4
```

Creating an EtherType-based rule in MAC ACL

MIB: qosclimib.mib

Tables used:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Scheme: a rule is created in two requests.

1. The parameters of the rule are set.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 1} i {mac-src(10), mac-dest(11),
vlan(12)}  \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 1} x {protocol index (HEX)} \
 1.3.6.1.4.1.89.88.5.1.3.{value of field 1} i {Value in port table for protocol
= 0. Constant for this rule} \
 1.3.6.1.4.1.89.88.5.1.2.{value of field 2} i {ether-type(17)} \
 1.3.6.1.4.1.89.88.5.1.3.{value of field 2} i {ether-type (DEC)} \
 1.3.6.1.4.1.89.88.5.1.4.{value of field 2} x {Zero field is a constant}
```

2. Binding a rule by index-of-rule to an ACL by index-of-acl as permit.

```
snmpset -v2c -c <community> <IP address> \
.1.3.6.1.4.1.89.88.31.1.3.{index-of-acl}.{index-of-rule} i {permit(1)
.1.3.6.1.4.1.89.88.31.1.4.{index-of-acl}.{index-of-rule} i {mac(5)} \
.1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {value of field 1} \
.1.3.6.1.4.1.89.88.31.1.9.{index-of-acl}.{index-of-rule} i {value of field 2}
```

Example of adding a permit 00:1f:c6:8b:c6:8a 00:00:00:00:00:00 any 806 0000 rule to MAC ACL 7-mac (since the rule is assumed to be the first one, then index-of-rule=20)

CLI command:

```
mac access-list extended 7-mac
 permit 00:1f:c6:8b:c6:8a 00:00:00:00:00:00 any 806 0000 ace-priority 20
exit
```

SNMP command:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.1 i 10 \
1.3.6.1.4.1.89.88.5.1.2.2 i 17 \
1.3.6.1.4.1.89.88.5.1.4.1 x "0x001fc68bc68a000000000000" \
1.3.6.1.4.1.89.88.5.1.3.1 i 0 \
1.3.6.1.4.1.89.88.5.1.3.2 i 2054 \
1.3.6.1.4.1.89.88.5.1.4.2 x "0x00 00"

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.1.20 i 1 \
1.3.6.1.4.1.89.88.31.1.4.1.20 i 5 \
1.3.6.1.4.1.89.88.31.1.5.1.20 i 1 \
1.3.6.1.4.1.89.88.31.1.9.1.20 i 2
```

TECHNICAL SUPPORT

For technical assistance in issues related to operation of ELTEX Enterprise Ltd. equipment, please contact our Service Center:

Feedback form on the website: **https://eltex-co.ru/support/**
Servicedesk: **https://servicedesk.eltex-co.ru**

Visit ELTEX official website to get the relevant technical documentation and software, benefit from our knowledge base, send us online request or consult a Service Center Specialist.

Official website: **https://eltex-co.ru/**
Knowledge base: **https://docs.eltex-co.ru/display/EKB/Eltex+Knowledge+Base**
Download Center: **https://eltex-co.ru/support/downloads**