Коммутаторы Ethernet

MES23xx, MES33xx, MES35xx, MES53xx

Мониторинг и управление Ethernet-коммутаторами MES по SNMP, версия ПО 4.0.16.2

Версия документа	Дата выпуска	Содержание изменений
Версия 1.10	**31.07.2021**	**Синхронизация с версией ПО 4.0.16.2**
Версия 1.9	**30.03.2021**	**Изменены разделы:** **19.2 Статистика QoS**
Версия 1.8	**02.03.2021**	**Синхронизация с версией ПО 4.0.15.3**
Версия 1.7	**10.02.2021**	**Изменены разделы:** **6.1 Параметры Ethernet-интерфейсов**
Версия 1.6	**28.11.2020**	**Изменены разделы:** **10.2 Настройка протокола Spanning-tree**
Версия 1.5	**27.10.2020**	**Добавлен раздел:** **18 Конфигурация защиты от DoS-атак**
Версия 1.4	**16.10.2020**	**Синхронизация с версией 4.0.14.2**
Версия 1.3	**14.09.2020**	**Изменены разделы:** **16.3 Защита IP-адреса клиента (IP source Guard)**
Версия 1.2	**19.02.2020**	**Добавлены разделы:** **4.3 Параметры стэка** **8 Настройка IPv6-адресации** **15 Электропитание по линиям Ethernet (POE)** **Изменены разделы:** **1 Настройка SNMP-сервера и отправки SNMP-TRAP** **2 Краткие обозначения** **4.1 Системные ресурсы** **4.2 Системные параметры** **4.4 Управление устройством** **5 Настройка системного времени** **6.1 Параметры Ethernet-интерфейсов** **6.3 Настройка и мониторинг errdisable-состояния** **10.2 Настройка протокола Spanning-tree** **12.1 Механизм AAA** **12.2 Настройка доступа** **13 Зеркалирование портов** **16.6 Механизм обнаружения петель (loopback-detection)**
Версия 1.1	**13.07.2018**	**Первая публикация.**
Версия программного обеспечения	**4.0.16.2**	

СОДЕРЖАНИЕ

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

Обозначение	Описание
[ ]	В квадратных скобках в командной строке указываются необязательные параметры, но их ввод предоставляет определенные дополнительные опции.
{}	В фигурных скобках в командной строке указываются обязательные параметры.
«,» **«-»**	Данные знаки в описании команды используются для указания диапазонов.
«|»	Данный знак в описании команды обозначает «или».
«/»	Данный знак при указании значений переменных разделяет возможные значения и значения по умолчанию.
Курсив Calibri	Курсивом Calibri указываются переменные или параметры, которые необходимо заменить соответствующим словом или строкой.
Полужирный курсив	Полужирным курсивом выделены примечания и предупреждения.
<Полужирный курсив>	Полужирным курсивом в угловых скобках указываются названия клавиш на клавиатуре.
`Courier New`	Полужирным Шрифтом Courier New записаны примеры ввода команд.

Примечания и предупреждения

Примечания содержат важную информацию, советы или рекомендации по использованию и настройке устройства.

Предупреждения информируют пользователя о ситуациях, которые могут нанести вред устройству или человеку, привести к некорректной работе устройства или потере данных.

1 НАСТРОЙКА SNMP-СЕРВЕРА И ОТПРАВКИ SNMP-TRAP

```
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 КРАТКИЕ ОБОЗНАЧЕНИЯ

- **ifIndex -** индекс порта;

Может принимать следующие значения:

1. Коммутаторы доступа

Модель коммутатора	Индексы
MES2308 MES2308R MES2308P MES2324 MES2324B MES2324F MES2324FB MES2348 MES2348B MES2324P MES2348P	- индексы 49-96 — gigabitethernet 1/0/1-48; - индексы 157-204 — gigabitethernet 2/0/1-48; - индексы 256-303 — gigabitethernet 3/0/1-48; - индексы 373-420 — gigabitethernet 4/0/1-48; - индексы 481-528 — gigabitethernet 5/0/1-48; - индексы 589-636 — gigabitethernet 6/0/1-48; - индексы 697-744 — gigabitethernet 7/0/1-48; - индексы 805-852 — gigabitethernet 8/0/1-48; - индексы 105-108 — tengigabitethernet 1/0/1-4; - индексы 213-216 — tengigabitethernet 2/0/1-4; - индексы 321-324 — tengigabitethernet 3/0/1-4; - индексы 429-432 — tengigabitethernet 4/0/1-4; - индексы 537-540 — tengigabitethernet 5/0/1-4; - индексы 645-648 — tengigabitethernet 6/0/1-4; - индексы 753-756 — tengigabitethernet 7/0/1-4; - индексы 861-864 — tengigabitethernet 8/0/1-4; - индексы 1000-1047 — Port-Channel 1/0/1-48; - индексы 100000-104095 — VLAN 1-4096.

2. Коммутаторы агрегации

Модель коммутатора	Индексы
MES3324 MES3324F MES3308F MES3316F MES3348 MES3348F	- индексы 49-96 — gigabitethernet 1/0/1-48; - индексы 157-204 — gigabitethernet 2/0/1-48; - индексы 256-303 — gigabitethernet 3/0/1-48; - индексы 373-420 — gigabitethernet 4/0/1-48; - индексы 481-528 — gigabitethernet 5/0/1-48;

	- индексы 589-636 — gigabitethernet 6/0/1-48; - индексы 697-744 — gigabitethernet 7/0/1-48; - индексы 805-852 — gigabitethernet 8/0/1-48; - индексы 105-108 — tengigabitethernet 1/0/1-4; - индексы 105-108 — tengigabitethernet 1/0/1-4; - индексы 213-216 — tengigabitethernet 2/0/1-4; - индексы 321-324 — tengigabitethernet 3/0/1-4; - индексы 429-432 — tengigabitethernet 4/0/1-4; - индексы 537-540 — tengigabitethernet 5/0/1-4; - индексы 645-648 — tengigabitethernet 6/0/1-4; - индексы 753-756 — tengigabitethernet 7/0/1-4; - индексы 861-864 — tengigabitethernet 8/0/1-4; - индексы 1000-1047 — Port-Channel 1/0/1-48; - индексы 100000-104095 — VLAN 1-4096.

3. Индустриальные коммутаторы

Модель коммутатора	Индексы
MES3508 MES3508P MES3510P	- индексы 49-62 — gigabitethernet 1/0/1-14; - индексы 1000-1047 — Port-Channel 1/0/1-48; - индексы 100000-104095 — VLAN 1-4096.

4. Коммутаторы для ЦОД

Модель коммутатора	Индексы
MES5324	- индексы 1-24 — tengigabitethernet 1/0/1-24; - индексы 53-76 — tengigabitethernet 2/0/1-24; - индексы 105-128 — tengigabitethernet 3/0/1-24; - индексы 157-180 — tengigabitethernet 4/0/1-24; - индексы 209-232 — tengigabitethernet 5/0/1-24; - индексы 261-284 — tengigabitethernet 6/0/1-24; - индексы 313-336 — tengigabitethernet 7/0/1-24; - индексы 365-388 — tengigabitethernet 8/0/1-24; - индексы 25-28 — fortygigabitethernet1/0/1-4; - индексы 77-80 — fortygigabitethernet2/0/1-4; - индексы 129-132 — fortygigabitethernet3/0/1-4; - индексы 181-184 — fortygigabitethernet4/0/1-4; - индексы 233-236 — fortygigabitethernet5/0/1-4; - индексы 285-288 — fortygigabitethernet6/0/1-4; - индексы 337-340 — fortygigabitethernet7/0/1-4; - индексы 389-392 — fortygigabitethernet8/0/1-4;

	- индексы 1000-1047 — Port-Channel 1/0/1-48; - индексы 100000-104095 — VLAN 1-4096.

- **index-of-rule —** индекс правила в ACL. Всегда кратен 20! Если при создании правил будут указаны индексы не кратные 20, то после перезагрузки коммутатора порядковые номера правил в ACL станут кратны 20;
- **Значение поля N —** в IP и MAC ACL любое правило занимает от одного до 3 полей в зависимости от его структуры;
- **IP address —** IP-адрес для управления коммутатором;

  В приведенных в документе примерах используется следующий IP-адрес для управления: **192.168.1.30**;
- **ip address of tftp server —** IP-адрес TFTP-сервера;

  В приведенных в документе примерах используется следующий IP-адрес TFTP-сервера: **192.168.1.1**;
- **community —** строка сообщества (пароль) для доступа по протоколу SNMP.

В приведенных в документе примерах используются следующие *community*:

private — права на запись (rw);
public — права на чтение (ro).

3 РАБОТА С ФАЙЛАМИ

3.1 *Сохранение конфигурации*

Сохранение конфигурации в энергонезависимую память

MIB: rlcopy.mib

Используемые таблицы: 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)}
```

Пример

Команда CLI:

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

Команда SNMP:

```
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
```

Сохранение конфигурации в энергозависимую память из энергонезависимой:

MIB: rlcopy.mib

Используемые таблицы: 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)}
```

Пример

Команда CLI:

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

Команда SNMP:

```
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
```

Удаление конфигурации из энергонезависимой памяти

MIB: RADLAN-rndMng

Используемые таблицы: rndAction — 1.3.6.1.4.1.89.1.2

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

Пример удаления startup-config

Команда CLI:
```
delete startup-config
```

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

3.2 *Работа с TFTP-сервером*

Копирование конфигурации из энергозависимой памяти на TFTP-сервер

MIB: RADLAN-COPY-MIB

Используемые таблицы: 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)}
```

Пример копирования из running-config на TFTP-сервер

Команда CLI:
```
copy running-config tftp://192.168.1.1/MES-config.cfg
```

Команда SNMP:
```
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 "MES-config.cfg" \
1.3.6.1.4.1.89.87.2.1.17.1 i 4
```

Копирование конфигурации в энергозависимую память с TFTP-сервера

MIB: rlcopy.mib

Используемые таблицы: 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)}
```

Пример копирования с TFTP-сервера в running-config

Команда CLI:

```
copy tftp://192.168.1.1/MES-config.cfg running-config
```

Команда SNMP:

```
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 "MES-config.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
```

Копирование конфигурации из энергонезависимой памяти на TFTP-сервер

MIB: файл rlcopy.mib

Используемые таблицы: 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)}
```

Пример копирования из startup-config на TFTP-сервер

Команда CLI:

```
copy startup-config tftp://192.168.1.1/MES-config.cfg
```

Команда SNMP:

```
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 3 \
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 "MES-config.cfg" \
1.3.6.1.4.1.89.87.2.1.17.1 i 4
```

Копирование конфигурации в энергонезависимую память с TFTP-сервера

MIB: RADLAN-COPY-MIB

Используемые таблицы: 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)}
```

Пример копирования startup-config с TFTP-сервера

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

Команда SNMP:
```
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 "MES-config.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 *Автоконфигурирование коммутатора*

Включение автоматического конфигурирования, базирующегося на DHCP (включено по умолчанию)

MIB: radlan-dhcpcl-mib.mib

Используемые таблицы: 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)}
```

Пример

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

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

3.4 *Обновление программного обеспечения*

Обновление программного обеспечения коммутатора

Проходит в два этапа:

1. Загрузка образа ПО

MIB: RADLAN-COPY-MIB

Используемые таблицы: 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)}
```

Пример

Команда CLI:

```
boot system tftp://192.168.1.1/mes3300-409-R478.ros
```

Команда SNMP:

```
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 "mes3300-409-R478.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
```

2. Смена активного образа коммутатора

MIB: RADLAN-DEVICEPARAMS-MIB

Используемые таблицы: 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)}
```

Пример

Команда CLI:

```
boot system inactive-image
```

Команда SNMP:

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

После загрузки ПО с tftp-сервера данная команда применяется автоматически

Перезагрузка коммутатора

MIB: rlmng.mib

Используемые таблицы: 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 {задержка времени перед перезагрузкой}
```

Пример: перезагрузка, отложенная на 8 минут

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

Для указания моментальной перезагрузки требуется указать значение t=0

Просмотр образа ПО

MIB: RADLAN-DEVICEPARAMS-MIB.mib

Используемые таблицы: 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
```

Пример

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

1) Возможные варианты:

image1(1)
image2(2)

2) Посмотреть активный образ ПО после перезагрузки можно в rndActiveSoftwareFileAfterReset — 1.3.6.1.4.1.89.2.13.1.1.3

Просмотр загруженных образов ПО

MIB: RADLAN-DEVICEPARAMS-MIB.mib

Используемые таблицы: 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
```

Пример

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

Просмотр текущей версии ПО коммутатора

MIB: RADLAN-DEVICEPARAMS-MIB.mib

Используемые таблицы: 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
```

Пример

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

Просмотр текущей HW версии

MIB: RADLAN-DEVICEPARAMS-MIB.mib

Используемые таблицы: 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
```

Пример

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

4 УПРАВЛЕНИЕ СИСТЕМОЙ

4.1 *Системные ресурсы*

Просмотр серийного номера коммутатора

MIB: rlphysdescription.mib

Используемые таблицы: 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
```

Пример

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

Просмотр информации о загрузке tcam

MIB:RADLAN-QOS-CLI-MIB

Используемые таблицы: 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
```

Пример

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

Просмотр максимального количества хостов

MIB: rltuning.mib

Используемые таблицы: 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
```

Пример

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

Просмотр используемого количества хостов

MIB: rlfft.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show system router resources
```

Команда SNMP:
```
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
```

Просмотр максимального количества маршрутов

MIB: rltuning.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show system router resources
```

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

Просмотр используемого количества маршрутов

MIB: rlip.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show system router resources
```

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

Просмотр максимального количества IP-интерфейсов

MIB: rltuning.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show system router resources
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.29.8.25.1
```

Просмотр используемого количества IP-интерфейсов

MIB: rlip.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show system router resources
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.26.23
```

Просмотр системного MAC-адреса коммутатора

MIB: rlphysdescription.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show system
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.53.4.1.7
```

Просмотр Uptime коммутатора

MIB: SNMPv2-MIB

Используемые таблицы: sysUpTime — 1.3.6.1.2.1.1.3

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

Пример

Команда CLI:
```
show system
```

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

Просмотр Uptime порта

MIB: SNMPv2-MIB, IF-MIB

Используемые таблицы:
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}
```

Пример: просмотра Uptime порта Gigabitethernet1/0/2

Команда CLI:
```
show interface status Gigabitethernet1/0/2
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.1.3
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.9.50
```

Примечание: из вывода первой команды необходимо отнять вывод второй команды. Полученное значение и будет являться uptime порта.

Включение сервиса мониторинга, приходящего на CPU трафика

MIB: rlsct.mib

Используемые таблицы: 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)}
```

Пример

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

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

Просмотр счетчиков и количества обрабатываемых CPU в секунду пакетов (по типам трафика)

MIB: rlsct.mib

Используемые таблицы: 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)}
```

Пример просмотра количества обрабатываемых CPU в секунду пакетов

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

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

Привязка индексов к типам трафика:

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)

Изменение лимитов CPU

MIB: eltSwitchRateLimiterMIB.mib

Используемые таблицы: 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}
```

Пример установки ограничения snmp трафика для CPU в 512 pps

Команда CLI:
```
service cpu-rate-limits snmp 512
```
Команда SNMP:
```
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
```

Список индексов:

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)

Мониторинг загрузки CPU

MIB: rlmng.mib

Используемые таблицы:
rlCpuUtilDuringLastSecond — 1.3.6.1.4.1.89.1.7
rlCpuUtilDuringLastMinute — 1.3.6.1.4.1.89.1.8
rlCpuUtilDuringLast5Minutes — 1.3.6.1.4.1.89.1.9

- Загрузка за последних пять секунд: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.1.7
- Загрузка за 1 минуту: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.1.8
- Загрузка за 5 минут: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.1.9

Пример просмотра загрузки CPU за последних пять секунд

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

Включение мониторинга загрузки CPU по процессам

MIB: RADLAN-rndMng

Используемые таблицы: 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)}
```

Пример

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

Мониторинг загрузки CPU по процессам

MIB: ELTEX-MES-MNG-MIB

Используемые таблицы:
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}
```

Пример просмотра загрузки по процессам за последние 5 секунд

Команда CLI:
```
show tasks utilization
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.9.1.2.1.1.3
```

Привязка индексов к процессам

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)
STSA(12)
STSB(13)
STSC(14)
STSD(15)
STSE(16)
CPUT(17)
EVAP(18)
HCLT(19)
EVLC(20)
SELC(21)
SEAU(22)
ESTC(23)
SSTC(24)
BOXS(25)
BSNC(26)
BOXM(27)
TRMT(28)
D_SP(29)
D_LM(30)
PLCT(31)
PLCR(32)
exRX(33)
3SWF(34)
MSRP(35)
HSES(36)
HSCS(37)
MRDP(38)
MLDP(39)
SETX(40)
EVTX(41)
SERX(42)
EVRX(43)
HLTX(44)
LBDR(45)
DDFG(46)
SYLG(47)
CDB_(48)
SNMP(49)
HCPT(61)
EVAU(62)
EVFB(63)
EVRT(64)
EPOE(65)
DSPT(66)
B_RS(67)
TRIG(68)
MACT(69)
SW2M(70)
3SWQ(71)
POLI(72)
OBSR(73)
NTPL(74)
L2HU(75)
L2PS(76)
SFSM(77)
NSCT(78)
NSFP(79)
NVCT(80)
NACT(81)
NSTM(82)
NINP(83)
L2UT(84)
BRGS(85)
FHSS(86)
FHSF(87)
FFTT(88)
IPAT(89)
IP6M(90)
IP6L(91)
IP6C(92)
IP6R(93)
RPTS(94)
ARPG(95)
IPG_(96)
DNSC(97)
ICMP(98)
TFTP(99)
TNSL(111)
SSHP(112)
PTPT(113)
NBBT(114)
SQIN(115)
MUXT(116)
DMNG(117)
DSYN(118)
HSEU(119)
DTSA(120)
SS2M(121)
DSND(122)
STMB(123)
AAAT(124)
AATT(125)
SCPT(126)
DH6C(127)
RCLA(128)
RCLB(129)
RCDS(130)
GRN_(131)
IPMT(132)
SNTP(133)
DHCP(134)
DHCp(135)
RELY(136)
MSSS(137)
WBAM(138)
WNTT(139)
RADS(140)
SNAS(141)
SNAE(142)
SNAD(143)
MNGT(144)
UTST(145)
SOCK(146)
TCPP(147)
UNQt(148)

Просмотр общего объема оперативной памяти

MIB: ELTEX-PROCESS-MIB.mib

Используемые таблицы: eltexProcessMemoryTotal — 1.3.6.1.4.1.35265.41.1.2.1.1.3

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.41.1.2.1.1.3.0
```

Пример

Команда CLI:

```
show cpu utilization
```

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

Просмотр свободного объема оперативной памяти

MIB: ELTEX-PROCESS-MIB.mib

Используемые таблицы: eltexProcessMemoryFree — 1.3.6.1.4.1.35265.41.1.2.1.1.7

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.41.1.2.1.1.7.0
```

Пример

Команда CLI:
```
show cpu utilization
```

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

Включение поддержки сверхдлинных кадров (jumbo-frames)

MIB: radlan-jumboframes-mib.mib

Используемые таблицы: 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)}
```

Пример

Команда CLI:
```
port jumbo-frame
```

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

4.2 *Системные параметры*

Контроль состояния блоков питания

MIB: rlphysdescription.mib

Используемые таблицы: rlPhdUnitEnvParamTable — 1.3.6.1.4.1.89.53.15

- Основной блок питания: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.2
- Резервный блок питания: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.3

Пример просмотра состояния основного блока питания

Команда CLI:
```
show system
```

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

1) для основного блока питания возможны следующие состояния:

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

2) для резервного блока питания возможны следующие состояния:

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

Мониторинг статуса АКБ

MIB: eltEnv.mib

Используемые таблицы: eltEnvMonBatteryState — 1.3.6.1.4.1.35265.1.23.11.1.1.2

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.11.1.1.2
```

Пример

Команда CLI:
```
show system battery
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.11.1.1.2
```

Возможные состояния:

normal (1) — батарея заряжена
warning (2) — батарея разряжается
critical (3) — низкий уровень заряда батареи
notPresent (5) — батарея отсутствует
notFunctioning (6) — авария расцепителя тока питания батареи
restore(7) — батарея заряжается

Мониторинг уровня заряда АКБ

MIB: eltEnv.mib

Используемые таблицы: eltEnvMonBatteryStatusCharge — 1.3.6.1.4.1.35265.1.23.11.1.1.3

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.11.1.1.3
```

Пример

Команда CLI:
```
show system battery
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.11.1.1.3
```

Контроль состояния вентиляторов

MIB: rlphysdescription.mib

Используемые таблицы: rlPhdUnitEnvParamTable — 1.3.6.1.4.1.89.53.15

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

Пример просмотра состояния вентилятора 3 коммутатора MES3324F

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

Возможны следующие состояния:

normal (1)
notFunctioning (5)

Контроль показаний температурных датчиков

MIB: rlphysdescription.mib

Используемые таблицы: rlPhdUnitEnvParamTable — 1.3.6.1.4.1.89.53.1

- Температурный датчик 1: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.10

Пример просмотра температуры датчика

Команда CLI:
```
show system
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.53.15.1.10
```

MES5324 имеет 4 температурных датчика, показания которых можно посмотреть командой CLI: show system sensors

Команда SNMP:
1 датчик:
**snmpwalk -v2c -c <community> <IP address> **
1.3.6.1.4.1.89.83.2.1.1.1.4.68420481
2 датчик:
**snmpwalk -v2c -c <community> <IP address> **

1.3.6.1.4.1.89.83.2.1.1.1.4.68420482
3 датчик:
**snmpwalk -v2c -c <community> <IP address> **
1.3.6.1.4.1.89.83.2.1.1.1.4.68420483
4 датчик:
**snmpwalk -v2c -c <community> <IP address> **
1.3.6.1.4.1.89.83.2.1.1.1.4.68420484

Контроль состояния температурных датчиков

MIB: rlphysdescription.mib

Используемые таблицы: rlPhdUnitEnvParamTable — 1.3.6.1.4.1.89.53.15

Температурный датчик 1: snmpwalk -v2c -c <community> <IP address> 1.3.6.1.4.1.89.53.15.1.11

Пример

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

MES5324 имеет 4 температурных датчика, состояния которых можно посмотреть командой CLI: show system sensors

Команда SNMP:
1 датчик:
**snmpwalk -v2c -c <community> <IP address> **
1.3.6.1.4.1.89.83.2.1.1.1.5.68420481
2 датчик:
**snmpwalk -v2c -c <community> <IP address> **
1.3.6.1.4.1.89.83.2.1.1.1.5.68420482
3 датчик:
**snmpwalk -v2c -c <community> <IP address> **
1.3.6.1.4.1.89.83.2.1.1.1.5.68420483
4 датчик:
**snmpwalk -v2c -c <community> <IP address> **
1.3.6.1.4.1.89.83.2.1.1.1.5.68420484

Состояния температурных датчиков
ok (1)
unavailable (2)
nonoperational (3)

4.3 *Параметры стэка*

Мониторинг параметров стэка

MIB: rlphysdescription.mib

Используемые таблицы: 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
```

Пример просмотра параметров стэка

Команда CLI:
```
show stack
```

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

Мониторинг стэковых портов

MIB: rlphysdescription.mib

Используемые таблицы: 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
```

Пример просмотра состояния стэковых портов

Команда CLI:
```
show stack links
```

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

4.4 *Управление устройством*

Задать/сменить hostname на устройстве

MIB: SNMPv2-MIB

Используемые таблицы: 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}"
```

Пример присвоения hostname "mes2324"

Команда CLI:
```
hostname mes2324
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.1.5.0 s "mes2324"
```

Включение/отключение management acl

MIB: RADLAN-MNGINF-MIB

Используемые таблицы:
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}
```

Пример включения management acl с именем eltex

Команда CLI:
```
management access-class eltex
```
Команда SNMP:
```
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
```

Использование утилиты ping

MIB: rlapplication.mib

Используемые таблицы: 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)}
```

Пример команды ping узла 192.168.1.1

Команда CLI:
```
ping 192.168.1.1 count 10 size 250 timeout 1000
```
Команда SNMP:
```
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
```

При установке в поле rsPingEntryStatus значения 4 (createAndGo) создаётся и активируется операция ping.
Чтобы повторно пропинговать удалённый хост, требуется в поле rsPingEntryStatus выставить значение 1(active).
После окончания операции обязательно надо удалить все записи, выставив в поле rsPingEntryStatus значение 6 (destroy). Иначе через CLI и SNMP операцию ping до другого хоста выполнить не удастся.

Пример удаления:
```
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

Мониторинг утилиты ping

MIB: rlapplication.mib

Используемые таблицы: 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.{Количество отправленых пакетов(7), Количество принятых
пакетов(8), Минимальное время ответа(9), Средние время ответа(10), Максимальное
время ответа(11)}
```

Пример просмотра количества принятых пакетов

Команда CLI:
```
ping 192.168.1.31
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.35.4.1.1.8
```

При установке в поле rsPingEntryStatus значения 6 (destroy) мониторинг будет запрещён до создания новой операции.

Настройка системного журнала

MIB: DRAFT-IETF-SYSLOG-DEVICE-MIB

Используемые таблицы: 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)}
```

Пример добавления сервера для логирования

Команда CLI:
```
logging host 192.168.1.1 description 11111
```

Команда SNMP:
```
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 задается следующим образом:

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 НАСТРОЙКА СИСТЕМНОГО ВРЕМЕНИ

Настройка адреса SNTP-сервера

MIB: rlsntp.mib

Используемые таблицы: 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. Байты IP-адреса
разделяются точками} i {true(1), false(2). Указание значения poll} \
 1.3.6.1.4.1.89.92.2.2.17.1.9.1.4.{ip address in DEC. Байты IP-адреса
разделяются точками} u 0 \
 1.3.6.1.4.1.89.92.2.2.17.1.10.1.4.{ip address in DEC. Байты IP-адреса
разделяются точками} i {createAndGo(4), destroy(6)}
```

Пример указания SNTP-сервера с IP-адресом 91.226.136.136

Команда CLI:

```
sntp server 91.226.136.136 poll
```

Команда SNMP:

```
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
```

Установка времени опроса для SNTP-клиента

MIB: rlsntp.mib

Используемые таблицы: 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}
```

Пример установки времени опроса в 60 секунд

Команда CLI:

```
sntp client poll timer 60
```

Команда SNMP:

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

Чтобы вернуться к настройкам по умолчанию достаточно установить время в 1024 сек.

Настройка работы одноадресных SNTP-клиентов

MIB: rlsntp.mib

Используемые таблицы: 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)}
```

Пример разрешения последовательного опроса SNTP-серверов

Команда CLI:
```
sntp unicast client poll
```

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

Добавление часового пояса

MIB: rlsntp.mib

Используемые таблицы: 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}"
```

Пример добавления часового пояса на устройстве

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

Команда SNMP:
```
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 КОНФИГУРИРОВАНИЕ ИНТЕРФЕЙСОВ

6.1 *Параметры Ethernet-интерфейсов*

Просмотр Description порта

MIB: IF-MIB или eltMng.mib

Используемые таблицы: ifAlias — 1.3.6.1.2.1.31.1.1.1.18 или 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}
```

Пример просмотра Description на интерфейсе GigabitEthernet1/0/1

Команда CLI:
```
show interfaces description GigabitEthernet 1/0/1
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.31.1.1.1.18.49

snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.1.31.1.1.1.1.49
```

Просмотр Description vlan

MIB: Q-BRIDGE-MIB

Используемые таблицы: 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}
```

Пример просмотра Description vlan 100

Команда CLI:
```
show interfaces description vlan 100
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.17.7.1.4.3.1.1.100
```

Просмотр скорости на интерфейсе

MIB: IF-MIB

Используемые таблицы: 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}
```

Пример выключения negotiation на GigabitEthernet1/0/2

Команда CLI:
```
show interface status GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.31.1.1.1.15.50
```

Включение/выключение автосогласования скорости на интерфейсе

MIB: rlinterfaces.mib

Используемые таблицы: 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)}
```

Пример выключения negotiation на GigabitEthernet1/0/2

Команда CLI:
```
interface GigabitEthernet1/0/2
no negotiation
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.16.50 i 2
```

Включение пропуска процедуры автосогласования, если партнер на встречной стороне не отвечает.

MIB: eltinterfaces.mib

Используемые таблицы: eltSwifAutoNegotiationBypass — 1.3.6.1.4.1.35265.1.23.43.1.1.3

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.43.1.1.3.{ifIndex} i {negotiationbypass(1), no
negotiation bypass(2)}
```

Пример выключения negotiation на TenGigabitEthernet1/0/2

Команда CLI:
```
interface TenGigabitEthernet1/0/2
no negotiation bypass
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.43.1.1.3.106 i 2
```

Установка режимов автосогласования скорости на интерфейсе

MIB: rlinterfaces.mib

Используемые таблицы: 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)}
```

Пример настройки автосогласования на скорости 10f и 100f на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
interface GigabitEthernet1/0/2
negotiation 10f 100f
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.40.50 x 14
```

1) В двоичной системе 10f и 100f записывается как 00010100. В HEX системе счисления это 14

2) *Описание битов*

default(0),
unknown(1),
tenHalf(2),
tenFull(3),
fastHalf(4),
fastFull(5),
gigaHalf(6),
gigaFull(7).

Порядок битов

0 1 2 3 4 5 6 7

Просмотр duplex режима порта

MIB: EtherLike-MIB

Используемые таблицы: 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}
```

Пример просмотра режима duplex порта GigabitEthernet 1/0/1

Команда CLI:
```
show interfaces status GigabitEthernet 1/0/1
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.2.1.19.49
```

Расшифровка выдаваемых значений

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

Смена duplex режима на интерфейсе

MIB: RADLAN-rlInterfaces

Используемые таблицы: 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)}
```

Пример смены режима duplex порта GigabitEthernet1/0/1

Команда CLI:
```
interface GigabitEthernet1/0/1
 duplex half
```

Команда SNMP:
```
snmpset-v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.3.49 i 2
```

Просмотр среды передачи интерфейса

MIB: EtherLike-MIB

Используемые таблицы: 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}
```

Пример просмотра среды передачи порта GigabitEthernet 1/0/1

Команда CLI:
```
show interfaces status GigabitEthernet 1/0/1
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.7.49
```

Расшифровка выдаваемых значений

Copper (1)
FiberOptics (2)
ComboCopper (3)
ComboFiberOptics (4)

Управление потоком (flowcontrol)

MIB: RADLAN-rlInterfaces

Используемые таблицы: 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)}
```

Пример включения управления потоком на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
interface GigabitEthernet1/0/2
flowcontrol on
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.14.50 i 1
```

Просмотр административного состояния порта

MIB: IF-MIB

Используемые таблицы: 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}
```

Пример просмотра статуса порта GigabitEthernet1/0/1

Команда CLI:
```
show interfaces status GigabitEthernet 1/0/1
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.7.49
```

Возможные варианты

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

Включить/выключить конфигурируемый интерфейс

MIB: IF-MIB

Используемые таблицы: 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)}
```

Пример

Команда CLI:
```
interface GigabitEthernet 1/0/1
shutdown
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.2.2.1.7.49 i 2
```

Просмотр оперативного состояния порта

MIB: IF-MIB

Используемые таблицы: 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}
```

Пример просмотра статуса порта GigabitEthernet1/0/1

Команда CLI:
```
show interfaces status GigabitEthernet 1/0/1
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.8.49
```

Возможные варианты

up(1)
down(2)

Определение типа подключения порта

MIB: rlinterfaces.mib

Используемые таблицы: 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}
```

Пример определения типа порта GigabitEthernet1/0/1

Команда CLI:
```
show interfaces status
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.7.49
```

Возможные варианты

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

Просмотр счетчика unicast-пакетов на интерфейсе

MIB: IF-MIB

Используемые таблицы: 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}
```

Пример просмотра счетчика входящих unicast-пакетов на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show interface counters GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.11.50
```

Просмотр счетчика multicast-пакетов на интерфейсе

MIB: IF-MIB

Используемые таблицы: 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}
```

Пример просмотра счетчика входящих multicast-пакетов на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show interface counters GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.31.1.1.1.2.50
```

Просмотр счетчика broadcast-пакетов на интерфейсе

MIB: IF-MIB

Используемые таблицы: 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}
```

Пример просмотра счетчика входящих broadcast-пакетов на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show interface counters GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.31.1.1.1.3.50
```

Просмотр счетчика октетов на интерфейсе

MIB: IF-MIB

Используемые таблицы:
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}
```

Пример просмотра счетчика принятых октетов на интерфейсе GigabitEthernet1/0/2

Команда CLI:

```
show interface counters gigabitethernet1/0/2
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.2.2.1.10.50
```

Под октетом имеется в виду количество байт.

1 октет = 1 байт

Просмотр счетчика FCS Errors на интерфейсе

MIB: EtherLike-MIB

Используемые таблицы: 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}
```

Пример просмотра счетчика FCS Errors на интерфейсе GigabitEthernet1/0/2

Команда CLI:

```
show interface counters GigabitEthernet1/0/2
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.2.1.3.50
```

Просмотр счетчика Internal MAC Rx Errors на интерфейсе

MIB: EtherLike-MIB

Используемые таблицы: 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}
```

Пример просмотра счетчика Internal MAC Rx Errors на интерфейсе GigabitEthernet1/0/2

Команда CLI:

```
show interface counters GigabitEthernet1/0/2
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.2.1.16.50
```

Просмотр счетчика Transmitted Pause Frames на интерфейсе

MIB: EtherLike-MIB

Используемые таблицы: 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}
```

Пример просмотра счетчика Transmitted Pause Frames на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show interface counters GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.10.1.4.50
```

Просмотр счетчика Received Pause Frames на интерфейсе

MIB: EtherLike-MIB

Используемые таблицы: 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}
```

Пример просмотра счетчика Received Pause Frames на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show interface counters GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.10.7.10.1.3.50
```

Очистка счетчиков интерфейсов

MIB: rlInterfaces.mib

Используемые таблицы: 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 {битовая маска}
```

Пример очистки счетчика интерфейсов для коммутатора MES2324B

Команда CLI:
```
clear counters
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.54.4.0 x
000000000000ffffff00000000f000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000001fffe0000000000
```

Пример очистки счетчика интерфейсов для коммутатора MES5324

Команда CLI:
```
clear counters
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.54.4.0 x
FFFFFFF000008FFFFFFF000008FFFFFFF000008FFFFFFF000008FFFFFFF000008FFFFFFF000008F
FFFFFFF000008FFFFFFF000008000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000001FFFE0000000000
```

1) В значение очистки счетчиков в стеке задается битовая маска для всех портов
всех юнитов стека:

**snmpset -v2c -c private 192.168.1.30 **
1.3.6.1.4.1.89.54.4.0 x
000000000000FFFFFF00000000F000000000000FFFFFF00000000F
000000000000FFFFFF00000000F000000000000FFFFFF00000000F000000000000FFFFFF0
0000000F000000000000FFFFFF00000000F000000000000FFFFFF00000000F00000000000
0FFFFFF00000000F000000000000000000000000000000001FFFE0000000000

2) Посмотреть значение битовой маски можно командой:

**snmpwalk -v2c -c public <IP address> **
1.3.6.1.4.1.89.54.9.0

Мониторинг загрузки портов коммутатора

MIB: eltMes.mib

Используемые таблицы: 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}
```

Пример

Команда CLI:
```
show interfaces utilization
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.43.2.1.1
```

Список возможных параметров

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

6.2 *Конфигурирование VLAN*

Добавление vlan в vlan database

MIB: rlvlan.mib

Используемые таблицы:
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 {битовая маска}
```

Пример создания 994 vlan'a в vlan database

Команда CLI:
```
vlan database
Vlan 994
```

Команда SNMP:
```
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) При расчете битовой маски для vlan 1025-2048, выполнить вычитание 1024 от необходимого vlan, а затем уже выполнять расчет маски. Аналогично для vlan 2049-3072: необходимо отнять 2048 перед расчетом. Для 3073-4094 — вычесть 3072. Битовая маска должна включать в себя не менее 10 символов.

2) Пример расчета битовой маски приведен в разделе «Приложение А. Методика расчета битовой маски».

Добавление vlan на порт

MIB: rlvlan.mib

Используемые таблицы: 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 в виде битовой маски}
```

Пример добавления vlan 622 и 3100 на интерфейс GigabitEthernet 1/0/2 в режим trunk

Команда CLI:
```
interface GigabitEthernet 1/0/2
 switchport mode trunk
 switchport trunk allowed vlan add 622,3100
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.1.50 x
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000004
```

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.4.50 x 0000001000
```

Пример добавления vlan 622 на интерфейс GigabitEthernet 1/0/2 в качестве native vlan

Команда CLI:

```
interface GigabitEthernet 1/0/2
 switchport mode trunk
 switchport trunk native vlan 622
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.1.50 x
0000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000004

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.5.50 x
0000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000004
```

Пример добавления vlan 622 на интерфейс GigabitEthernet 1/0/2 в режиме access

Команда CLI:

```
interface GigabitEthernet 1/0/2
 switchport access vlan 622
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.1.50 x
0000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000004 \
1.3.6.1.4.1.89.48.68.1.5.50 x
0000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000004
```

1. Перечень таблиц

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. Пример составления битовой маски приведен в разделе «Приложение А. Методика расчета битовой маски».

3. Битовая маска должна включать в себя не менее 10 символов.

Запретить default VLAN на порте

MIB: eltVlan.mib

Используемые таблицы: 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 {порт в виде битовой маски}
```

Пример запрета default vlan на порте GigabitEthernet 1/0/5

Команда CLI:
```
interface GigabitEthernet1/0/5
switchport forbidden default-vlan
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.5.5.1.0 x 00000000000008
```

1. Пример составления битовой маски приведен в разделе «Приложение А. Методика расчета битовой маски».

2. Битовая маска должна включать в себя не менее 10 символов.

Просмотр имени VLAN

MIB: rlvlan.mib

Используемые таблицы: 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}
```

Пример просмотра имени vlan 5

Команда CLI:
```
show vlan tag 5
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.48.70.1.1.5
```

Просмотр членства порта во VLAN

MIB: rlvlan.mib

Используемые таблицы: 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}
```

Пример просмотра vlan на GigabitEthernet1/0/5

Команда CLI:

```
show interfaces switchport GigabitEthernet1/0/5
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.1.54
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.48.68.1.5.54
```

1. В примере представленны 2 команды snmpwalk. Если порт Tagged — значения в выводе второй команды принимают нулевое значение и номер Vlan соответствует значениям вывода первой команды. Если порт Untagged — в выводе второй команды присутствуют значения, отличные от нуля, и номер Vlan соответствует этим значениям.

2. Перечень таблиц

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. Полученные в результате выполнения запроса значения представляют из себя битовую маску, методика расчета которой приведена в разделе «Приложение А. Методика расчета битовой маски».

Настройка режима работы порта

MIB: rlvlan.mib

Используемые таблицы: 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)}
```

Пример настройки интерфейса GigabitEthernet 1/0/2 в режим trunk

Команда CLI:

```
interface GigabitEthernet 1/0/2
 switchport mode trunk
```

Команда SNMP:

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

Просмотр режима порта

MIB: rlvlan.mib

Используемые таблицы: 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}
```

Пример просмотра режима на GigabitEthernet1/0/2

Команда CLI:

```
show interfaces switchport GigabitEthernet1/0/2
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.48.22.1.1.50
```

Возможные варианты

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

Назначить pvid на интерфейс

MIB: Q-BRIDGE-MIB.mib

Используемые таблицы: 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}
```

Пример назначения pvid 15 для GigabitEthernet 1/0/2

Команда CLI:

```
interface GigabitEthernet 1/0/2
 switchport general pvid 15
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.17.7.1.4.5.1.1.50 u 15
```

Настройка map mac

MIB: rlvlan.mib

Используемые таблицы: 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)}
```

Пример

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

Команда SNMP:
```
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
```

Установка правила классификации VLAN, основанного на привязке к MAC-адресу, для интерфейса

MIB: rlvlan.mib

Используемые таблицы: 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)}
```

Пример включения правила классификации VLAN для интерфейса gigabitethernet 1/0/10

Команда CLI:
```
interface Gigabitethernet 1/0/10
switchport general map macs-group 1 vlan 20
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.46.1.2.58.1 u 20 \
1.3.6.1.4.1.89.48.46.1.3.58.1 i 4
```

6.3 *Настройка и мониторинг errdisable-состояния*

Просмотр настроек для автоматической активации интерфейса

MIB: rlinterfaces_recovery.mib

Используемые таблицы: 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
```

Пример: просмотр настроек для автоматической активации интерфейса

Команда CLI:
```
show errdisable recovery
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.128.2.1.2
```

Просмотр причины блокировки порта

MIB: rlErrdisableRecoveryIfReason

Используемые таблицы: 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
```

Пример

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

Возможные варианты:

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)

Настройка автоматической активации интерфейса

MIB: rlinterfaces_recovery.mib

Используемые таблицы: 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)}
```

Пример включения автоматической активации интерфейса в случае loopback detection

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

Возможные значения index of reason, в зависимости от типа выполняемой настройки:

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)

Настройка интервала выхода интерфейса из errdisable состояния

MIB: rlinterfaces_recovery.mib

Используемые таблицы: 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}
```

Пример настройки 30-ти секундного интервала выхода из errdisable состояния

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

6.4 *Настройка voice vlan*

Добавление voice vlan

MIB: RADLAN-vlanVoice-MIB

Используемые таблицы: 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}
```

Пример добавления voice vlan id 10

Команда CLI:
```
voice vlan id 10
```

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

Активация voice vlan на интерфейсе

MIB: RADLAN-vlanVoice-MIB

Используемые таблицы: 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}
```

Пример

Команда CLI:
```
interface GigabitEthernet1/0/3
voice vlan enable
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.54.12.5.1.1.51 i 1 \
1.3.6.1.4.1.89.48.54.12.5.1.2.51 u 10
```

Редактирование таблицы OUI

MIB: rlvlanVoice.mib

Используемые таблицы: 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. Байты разделяются точками} i
{createAndGo(4), destroy(6)}
```

Пример

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

Команда SNMP:
```
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
```

6.5 *Настройка LLDP*

Глобальное включение/отключение lldp

MIB: rlLldp.mib

Используемые таблицы: 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)}
```

Пример отключения LLDP

Команда CLI:
```
no Lldp run
```

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

Настройка lldp-med политики с указанием номера voice vlan для тегированного трафика voice vlan

MIB: rllldb.mib

Используемые таблицы: 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)}
```

Пример настройки lldp-med политики с указанием VLAN 10, указанием приоритета 4

Команда CLI:

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

Команда SNMP:

```
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
```

Настройка lldp-med политики для тегированного трафика voice vlan

MIB: rllldb.mib

Используемые таблицы: 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)}
```

Пример настройки lldp-med политики в режиме auto

Команда CLI:

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

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.110.1.7.0 i 0
```

7 НАСТРОЙКА IPV4-АДРЕСАЦИИ

Создание IP-адреса на interface vlan:

MIB: rlip.mib

Используемые таблицы: 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}
```

Пример настройки адреса 192.168.10.30/24 на vlan 30

Команда CLI:

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

Команда SNMP:

```
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
```

Удаление IP-адреса на interface vlan:

MIB: rlip.mib

Используемые таблицы: 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
```

Пример удаления IP-адреса 192.168.10.30 на интерфейсе vlan 30

Команда CLI:

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

Команда SNMP:

```
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
```

Получение IP-адреса по DHCP на interface vlan

MIB: radlan-dhcpcl-mib.mib

Используемые таблицы: 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)}
```

Пример

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

Добавить/удалить шлюз по умолчанию

MIB: rlip.mib

Используемые таблицы: 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)}
```

Пример добавления ip default-gateway 192.168.1.10

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

Команда SNMP:
```
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 НАСТРОЙКА IPV6-АДРЕСАЦИИ

Включение/выключение IPv6 адресации на interface vlan

MIB: ip-mib.mib

Используемые таблицы: 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)}
```

Пример включения IPv6 адресации на vlan 2

Команда CLI:

```
interface vlan 2
ipv6 enable
```

Команда SNMP:

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

Создание/Удаление IPv6 адреса на interface vlan

MIB: rlip.mib

Используемые таблицы: 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.{количество байт в адресе}.{каждый байт в десятичном
виде через разделитель}.2.0 i {ifindex} \
1.3.6.1.4.1.89.26.36.1.13.{количество байт в адресе}.{каждый байт в десятичном
виде через разделитель}.2.0 u {маска в десятичном виде}\
1.3.6.1.4.1.89.26.36.1.11.{количество байт в адресе}.{каждый байт в десятичном
виде через разделитель}.2.0 i {createAndGo (4), destroy(6)}
```

Пример добавления адреса 2001::1/64 на vlan 2

Команда CLI:

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

Команда SNMP:

```
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 НАСТРОЙКА GREEN ETHERNET

Глобальное отключение green-ethernet short-reach

MIB: rlgreeneth.mib

Используемые таблицы: 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)}
```

Пример

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

Глобальное отключение green-ethernet energy-detect

MIB: rlgreeneth.mib

Используемые таблицы: 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)}
```

Пример

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

Просмотр параметров green-ethernet

MIB: rlGreenEth.mib

Используемые таблицы: 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
```

Пример

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

10 НАСТРОЙКА КОЛЬЦЕВЫХ ПРОТОКОЛОВ

10.1 *Протокол ERPS*

Определение номера west порта

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

Используемые таблицы: 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
```

Пример

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

Просмотр состояния west порта

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

Используемые таблицы: 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
```

Пример

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

Возможные состояния порта:

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

Определение номера east порта

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

Используемые таблицы: 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
```

Пример

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

Просмотр состояния east порта

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

Используемые таблицы: 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
```

Пример

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

Возможные состояния порта:

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

Просмотр состояния кольца

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

Используемые таблицы: 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
```

Пример

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

Возможные состояния кольца erps:

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

10.2 *Настройка протокола Spanning-tree*

Включение/отключение протокола Spanning-tree

MIB: radlan-brgmacswitch.mib

Используемые таблицы: 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)}
```

Пример отключения Spanning-tree

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

Включение/выключение протокола spanning-tree на конфигурируемом интерфейсе

MIB: BRIDGE-MIB

Используемые таблицы: 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)}
```

Пример отключения работы spanning-tree на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
interface GigabitEthernet1/0/2
spanning-tree disable
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.17.2.15.1.4.50 i 2
```

Включение/выключение режима обработки пакетов BPDU интерфейсом, на котором выключен протокол STP

MIB: radlan-bridgemibobjects-mib.mib

Используемые таблицы: 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)}
```

Пример включения фильтрации BPDU на интерфейсе Gigabitethernet 1/0/2

Команда CLI:
```
interface gigabitethernet 1/0/2
spanning-tree bpdu filtering
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.57.2.13.1.4.50 i 1
```

Настройка режима работы протокола spanning-tree

MIB: draft-ietf-bridge-rstpmib.mib

Используемые таблицы: 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)}
```

Пример установки режима работы протокола Spanning-tree

Команда CLI:
```
spanning-tree mode rstp
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.17.2.16.0 i 2
```

Просмотр роли порта в STP

MIB: radlan-bridgemibobjects-mib.mib

Используемые таблицы: 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}
```

Пример просмотра роли Gigabitethernet0/2 в STP

Команда CLI:
```
show spanning-tree Gigabitethernet0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.57.2.13.1.7.50
```

Возможные состояния порта:

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

Просмотр состояния порта в MSTP

MIB: radlan-bridgemibobjects-mib.mib

Используемые таблицы: 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}
```

Пример просмотра состояния Gigabitethernet0/2 в mstp

Команда CLI:
```
show spanning-tree Gigabitethernet0/2
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.57.6.2.1.4.1.50
```

Возможные состояния порта:

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

Просмотр времени с последнего перестроения (topology change)

MIB: BRIDGE-MIB

Используемые таблицы: 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
```

Пример просмотра времени с последнего перестроения

Команда CLI:
```
show spanning-tree
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.2.1.17.2.3.0
```

Количество перестроений (topology change)

MIB: BRIDGE-MIB

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show spanning-tree
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.2.1.17.2.4.0
```

Просмотр интерфейса, с которого принят последний topology change

MIB: eltBridgeExtMIB.mib

Используемые таблицы: 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
```

Пример просмотра интерфейса, с которого принят последний topology change

Команда CLI:

```
show spanning-tree
```

Команда SNMP:

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

11 ГРУППОВАЯ АДРЕСАЦИЯ

11.1 *Правила групповой адресации (multicast addressing)*

Запрещение динамического добавления порта к многоадресной группе

MIB: rlbrgmulticast.mib

Используемые таблицы: 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
{Битовая маска интерфейса} \
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)}
```

Пример запрета изучения группы 239.200.200.17 на порте GigabitEthernet 1/0/1 в vlan 622

Команда CLI:

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

Команда SNMP:

```
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 0000000000008000
\
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) Суммарное количество цифр в OID 1.3.6.1.4.1.89.116.5.1.6 и OID 1.3.6.1.4.1.89.116.5.1.7 должно быть одинаковым и чётным.

2) Методику расчета битовой маски можно посмотреть в разделе «Приложение А. Методика расчета битовой маски».

Запрещение прохождения незарегистрированного Multicast-трафика

MIB: rlbrgmulticast.mib

Используемые таблицы: 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 "{Битовая маска для интерфейсов}"
```

Пример запрещения прохождения незарегистрированного Multicast-трафика для портов GigabitEthernet 1/0/20-21

Команда CLI:

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

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.55.4.1.0 x "0000000000000018"
```

1) Для удаления настройки надо заменить соответствующие портам поля в битовой маске на 0.

2) Методику расчета битовой маски можно посмотреть в разделе «Приложение А. Методика расчета битовой маски».

Фильтрация многоадресного трафика

MIB: rlbrgmulticast.mib

Используемые таблицы: 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)}
```

Пример включения фильтрации многоадресного трафика

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

Глобальное включение igmp snooping

MIB: rlbrgmulticast.mib

Используемые таблицы: 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)}
```

Пример

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

Включение igmp snooping в vlan

MIB: rlbrgmulticast.mib

Используемые таблицы: 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)}
```

Пример включения igmp snooping в vlan 30

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

Просмотр таблицы igmp snooping

MIB: rlbrgmulticast.mib

Используемые таблицы: 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
```

Пример

Команда CLI:

```
show ip igmp snooping groups
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.55.5.4
```

Настройка multicast-tv vlan (MVR)

MIB: rlvlan.mib

Используемые таблицы: 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)}
```

Пример настройки multicast-tv vlan 622 на интерфейсе gigabitethernet 1/0/2

Команда CLI:

```
interface gigabitethernet 1/0/2
switchport access multicast-tv vlan 622
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.44.1.1.50 u 622 \
1.3.6.1.4.1.89.48.44.1.2.50 i 4
```

Настройка режима работы multicast-tv vlan <customer/access/trunk/general> зависит от режима настройки порта, т.е. от команды switchport mode customer/access/trunk/general.

11.2 *Функции ограничения multicast-трафика*

Создание multicast snooping profile

MIB: eltIpMulticast.mib

Используемые таблицы: 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)}
```

Пример создания профиля с именем IPTV (предположим, что профиль будет иметь порядковый номер 3)

Команда CLI:
```
multicast snooping profile IPTV
```
Команда SNMP:
```
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
```

Указание диапазонов Multicast-адресов в multicast snooping profile

MIB: eltIpMulticast.mib

Используемые таблицы: 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-
адрес начала диапазона в шестнадцатеричном виде} \
 1.3.6.1.4.1.35265. 1.23.46.3.1.5.{index of rule}.{Index of profile} x {ip-
адрес конца диапазона в шестнадцатеричном виде} \
 1.3.6.1.4.1.35265. 1.23.46.3.1.6.{index of rule}.{Index of profile} i
{createAndGo(4), destroy(6)}
```

Пример ограничения мультикаст групп 233.7.70.1-233.7.70.10 для профиля с именем IPTV (предположим, что профиль имеет порядковый номер 3. В первом профиле 2 правила, во втором—одно)

Команда CLI:
```
multicast snooping profile IPTV
 match ip 233.7.70.1 233.7.70.10
```
Команда SNMP:
```
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* — считается по сумме всех правил во всех профилях**

Назначение multicast snooping profile на порт

MIB: eltIpMulticast.mib

Используемые таблицы: 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)}
```

Пример добавления профиля test (с индексом профиля 3) на интерфейс Gigabitethernet 1/0/2

Команда CLI:

```
interface Gigabitethernet 1/0/2
 multicast snooping add test
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.46.7.1.1.50.3 i 50 \
1.3.6.1.4.1.35265.1.23.46.7.1.2.50.3 i 3 \
1.3.6.1.4.1.35265.1.23.46.7.1.3.50.3 i 4
```

Настройка ограничения количества Multicast-групп на порте

MIB: eltIpMulticast.mib

Используемые таблицы: 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}
```

Пример настройки ограничения в три Multicast-группы на интерфейсе Gigabitethernet 1/0/2

Команда CLI:

```
interface Gigabitethernet 1/0/2
 multicast snooping max-groups 3
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.46.6.1.2.50 i 3
```

12 ФУНКЦИИ УПРАВЛЕНИЯ

12.1 *Механизм AAA*

Добавление нового пользователя

MIB: rlaaa.mib

Используемые таблицы: 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, каждая буква
логина отделяется от следующей точкой} s {login} \
 1.3.6.1.4.1.89.79.17.1.2.{number of letters}.{Login in DEC, каждая буква
логина отделяется от следующей точкой} s "#{encoding password}" \
 1.3.6.1.4.1.89.79.17.1.3.{number of letters}.{Login in DEC, каждая буква
логина отделяется от следующей точкой} i {privelege level(1-15)} \
 1.3.6.1.4.1.89.79.17.1.4.{number of letters}.{Login in DEC, каждая буква
логина отделяется от следующей точкой} i {create and go(4)}
```

Пример добавления пользователя techsup с паролем password и уровнем привилегий 15

Команда CLI:

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

Команда SNMP:

```
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. Логин переводится из ASCII в HEX с помощью таблицы, которую можно найти по ссылке https://ru.wikipedia.org/wiki/ASCII

2. Пароль задается исключительно в шифрованном виде, пишется обязательно в кавычках, перед паролем добавляется #.

Настройка методов авторизации для login-пользователя

MIB: rlaaa.mib

Используемые таблицы: rlAAAMethodListEntry — 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.15.1.2.15.{“login_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(2),radius(4),tacacs(5),local(3)} \
1.3.6.1.4.1.89.79.15.1.3.15.{“login_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(2),radius(4),tacacs(5),local(3)} \
1.3.6.1.4.1.89.79.15.1.4.15.{“login_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(2),radius(4),tacacs(5),local(3)} \
1.3.6.1.4.1.89.79.15.1.10.15.{“login_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i 1 \
1.3.6.1.4.1.89.79.15.1.10.15.{“login_n_default” in DEC, каждая буква логина
отделяется от следующей точкой} i 1
```

Пример

Команда CLI:

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

Команда SNMP:

```
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 \
1.3.6.1.4.1.89.79.15.1.3.15.108.111.103.105.110.95.99.95.100.101.102.97.117.108
.116 i 0 \
1.3.6.1.4.1.89.79.15.1.4.15.108.111.103.105.110.95.99.95.100.101.102.97.117.108
.116 i 0 \
1.3.6.1.4.1.89.79.15.1.10.15.108.111.103.105.110.95.99.95.100.101.102.97.117.10
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
```

108.111.103.105.110.95.99.95.100.101.102.97.117.108.116 переводится из ASCII таблицы (расшифровывается login_c_default)

Удаление настройки методов авторизации для login-пользователя

MIB: rlaaa.mib

Используемые таблицы: rlAAAMethodListEntry — 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.15.1.2.15.{"login_c_default" in DEC, каждая буква логина
отделяется от следующей точкой} i 3 \
1.3.6.1.4.1.89.79.15.1.3.15.{"login_c_default" in DEC, каждая буква логина
отделяется от следующей точкой} i 0 \
1.3.6.1.4.1.89.79.15.1.4.15.{"login_c_default" in DEC, каждая буква логина
отделяется от следующей точкой} i 0 \
1.3.6.1.4.1.89.79.15.1.10.15.{"login_c_default" in DEC, каждая буква логина
отделяется от следующей точкой} i 0 \
1.3.6.1.4.1.89.79.15.1.10.15.{"login_n_default" in DEC, каждая буква логина
отделяется от следующей точкой} i 0
```

Пример удаления методов авторизации для enable пользователя

Команда CLI:

```
no aaa authentication login default
```

Команда SNMP:

```
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 \
1.3.6.1.4.1.89.79.15.1.3.15.108.111.103.105.110.95.99.95.100.101.102.97.117.108
.116 i 0 \
1.3.6.1.4.1.89.79.15.1.4.15.108.111.103.105.110.95.99.95.100.101.102.97.117.108
.116 i 0
\1.3.6.1.4.1.89.79.15.1.10.15.108.111.103.105.110.95.99.95.100.101.102.97.117.1
08.116 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 0
```

Настройка методов авторизации для enable-пользователя

MIB: rlaaa.mib

Используемые таблицы: rlAAAMethodListEntry — 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.15.1.2.16.{“enable_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(2),radius(4),tacacs(5)} \
1.3.6.1.4.1.89.79.15.1.3.16.{“enable_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(2),radius(4),tacacs(5)} \
1.3.6.1.4.1.89.79.15.1.4.16.{“enable_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(2),radius(4),tacacs(5)} \
1.3.6.1.4.1.89.79.15.1.10.16.{“enable_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i 1 \
1.3.6.1.4.1.89.79.15.1.10.16.{“enable_n_default” in DEC, каждая буква логина
отделяется от следующей точкой} i 1
```

Пример

Команда CLI:

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

Команда SNMP:

```
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 5 \
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 4 \
1.3.6.1.4.1.89.79.15.1.4.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.1
08.116 i 2 \
1.3.6.1.4.1.89.79.15.1.10.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.
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
```

101.110.97.98.108.101.95.99.95.100.101.102.97.117.108.116 переводится из ASCII таблицы (расшифровывается enable_c_default)

Удаление настройки методов авторизации для enable-пользователя

MIB: rlaaa.mib

Используемые таблицы: rlAAAMethodListEntry — 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.79.15.1.2.16.{“enable_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(0),radius(0),tacacs(2)} \
1.3.6.1.4.1.89.79.15.1.3.16.{“enable_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(0),radius(0),tacacs(2)} \
1.3.6.1.4.1.89.79.15.1.4.16.{“enable_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i {enable(0),radius(0),tacacs(2)} \
1.3.6.1.4.1.89.79.15.1.10.16.{“enable_c_default” in DEC, каждая буква логина
отделяется от следующей точкой} i 0 \
1.3.6.1.4.1.89.79.15.1.10.16.{“enable_n_default” in DEC, каждая буква логина
отделяется от следующей точкой} i 0
```

Пример удаления методов авторизации для enable пользователя

Команда CLI:
```
no aaa authentication enable default
```
Команда SNMP:
```
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 2 \
1.3.6.1.4.1.89.79.15.1.3.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.1
08.116 i 0 \
1.3.6.1.4.1.89.79.15.1.4.16.101.110.97.98.108.101.95.99.95.100.101.102.97.117.1
08.116 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 \
1.3.6.1.4.1.89.79.15.1.10.16.101.110.97.98.108.101.95.110.95.100.101.102.97.117
.108.116 i 0
```

12.2 *Настройка доступа*

Включение telnet-сервера

MIB: radlan-telnet-mib.mib

Используемые таблицы: 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)}
```

Пример включения telnet-сервера

Команда CLI:
```
ip telnet server
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.58.7.0 i 1
```

Включение ssh сервера

MIB: rlssh.mib

Используемые таблицы: 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)}
```

Пример включения ssh-сервера

Команда CLI:
```
ip ssh server
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.78.2.102.0 i 1
```

Просмотр активных сессий

MIB: rlAAA.mib

Используемые таблицы: 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
```

Пример просмотра активных сессий

Команда CLI:

```
Show users
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.79.57.1.5
```

13 ЗЕРКАЛИРОВАНИЕ ПОРТОВ

Настройка зеркалирования портов

MIB: rfc2613.mib

Используемые таблицы: portCopyTable — 1.3.6.1.2.1.16.22.1.3.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.2.1.16.22.1.3.1.1.4.{ifindex src port}.{ifindex dst port} i
{copyRxOnly(1), copyTxOnly(2), copyBoth(3)} \
 1.3.6.1.2.1.16.22.1.3.1.1.5.{ifindex src port}.{ifindex dst port} i
{createAndGo(4), destroy(6)}
```

Пример зеркалирования трафика с интерфейса GigabitEthernet 1/0/1 на интерфейс GigabitEthernet 1/0/2

Команда CLI:

```
interface GigabitEthernet 1/0/2
 port monitor GigabitEthernet 1/0/1
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.16.22.1.3.1.1.4.49.50 i 3 \
1.3.6.1.2.1.16.22.1.3.1.1.5.49.50 i 4
```

Настройка зеркалирования vlan

MIB: rfc2613.mib

Используемые таблицы: portCopyTable — 1.3.6.1.2.1.16.22.1.3.1

```
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)}
```

Пример настройки зеркалирования vlan 622 на интерфейс GigabitEthernet 1/0/2

Команда CLI:

```
interface GigabitEthernet 1/0/2
 port monitor vlan 622
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.16.22.1.3.1.1.4.100621.50 i 1 \
1.3.6.1.2.1.16.22.1.3.1.1.5.100621.50 i 4
```

14 ФУНКЦИИ ДИАГНОСТИКИ ФИЗИЧЕСКОГО УРОВНЯ

14.1 *Диагностика медного кабеля*

Запуск TDR теста для порта

MIB: rlphy.mib

Используемые таблицы: rlPhyTestSetType — 1.3.6.1.4.1.89.90.1.1.1.1

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.90.1.1.1.1.{ifIndex} i 2
```

Пример запуска tdr для порта GigabitEthernet 1/0/12

Команда CLI:
```
test cable-diagnostics tdr interface GigabitEthernet 1/0/12
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.90.1.1.1.1.60 i 2
```

Для запуска теста tdr-fast указать параметр i 25.

Чтение информации по парам при тестировании методом TDR

MIB: eltPhy.mib

Используемые таблицы: eltPhyTdrTestTable — 1.3.6.1.4.1.35265.1.23.90.1.1

- Статус 1 (1-2) пары:

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.90.1.1.1.2.{ifIndex}
```

- Статус 2 (3-6) пары:

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.90.1.1.1.3.{ifIndex}
```

- Статус 3 (4-5) пары:

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.90.1.1.1.4.{ifIndex}
```

- Статус 4 (7-8) пары:

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.90.1.1.1.5.{ifIndex}
```

Пример просмотра статуса пары 1 на интерфейсе GigabitEthernet 1/0/12

Команда CLI:
```
show cable-diagnostics tdr interface GigabitEthernet 1/0/12
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.90.1.1.1.2.60
```

Варианты статусов пар:

test-failed(0) — физическая неисправность; либо в момент запроса идет диагностика линии;
ok(1) — пара в порядке;
open(2) — разрыв;
short(3) — контакты пары замкнуты;
impedance-mismatch(4) — разница в сопротивлении (слишком большое затухание в линии);
short-with-pair-1(5) — замыкание между парами;
short-with-pair-2(6) — замыкание между парами;
short-with-pair-3(7) — замыкание между парами;
short-with-pair-4(8) — замыкание между парами.

Измерение длины пар для метода TDR

MIB: eltPhy.mib

Используемые таблицы: eltPhyTdrTestTable — 1.3.6.1.4.1.35265. 1.23.90.1.1

- Длина 1 (1-2) пары:

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.90.1.1.1.6.{ifIndex}
```

- Длина 2 (3-6) пары:

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.90.1.1.1.7.{ifIndex}
```

- Длина 3 (4-5) пары:

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.90.1.1.1.8.{ifIndex}
```

- Длина 4 (7-8) пары:

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265. 1.23.90.1.1.1.9.{ifIndex}
```

Пример измерения длины пары 4 для метода tdr на интерфейсе GigabitEthernet 1/0/12

Команда CLI:
```
show cable-diagnostics tdr interface GigabitEthernet 1/0/12
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
```

```
1.3.6.1.4.1.35265.1.23.90.1.1.1.9.60
```

Измерение длины кабеля методом, основанном на затухании

MIB: rlphy.mib

Используемые таблицы: 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.{ifIndex}
```

Пример измерения длины кабеля на всех активных портах

Команда CLI:
```
show cable-diagnostics cable-length
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.90.1.2.1.3
```

14.2 *Диагностика оптического трансивера*

Снятие показаний DDM

MIB: rlphy.mib

Используемые таблицы: 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.{индекс порта}.{тип параметра}
```

Пример запроса показаний DDM с интерфейса TengigabitEthernet1/0/1 (для всех параметров)

Команда CLI:
```
show fiber-ports optical-transceiver interface TengigabitEthernet0/1
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.90.1.2.1.3.105
```

Тип параметра может принимать следующие значения:

rlPhyTestTableTransceiverTemp (5) — температура SFP трансивера;
rlPhyTestTableTransceiverSupply (6) — напряжение питания в мкВ;
rlPhyTestTableTxBias (7) — ток смещения в мкА;
rlPhyTestTableTxOutput (8) — уровень мощности на передаче в mDbm;
rlPhyTestTableRxOpticalPower (9) — уровень мощности на приеме в mDbm.

Просмотр серийного номера SFP трансивера

MIB: eltMes.mib

Используемые таблицы: 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.{индекс порта}
```

Пример просмотра серийного номера SFP с интерфейса GigabitEthernet 1/0/2 (для всех параметров)

Команда CLI:

```
show fiber-ports optical-transceiver interface GigabitEthernet 1/0/2
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.53.1.1.1.6.50
```

15 ЭЛЕКТРОПИТАНИЕ ПО ЛИНИЯМ ETHERNET (POE)

Просмотр потребляемой/номинальной мощности PoE

MIB: rfc3621.mib

Используемые таблицы: pethMainPseEntry — 1.3.6.1.2.1.105.1.3.1.1

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

1.3.6.1.2.1.105.1.3.1.1.{nominal(2), consumed(4)}.{unit}

Пример просмотра потребляемой мощности

Команда CLI:
```
show power inline
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.105.1.3.1.1.4.1
```

Просмотр показаний температурного датчика PoE

MIB: rlPoe.mib

Используемые таблицы: rlPethPowerPseTemperatureSensor — 1.3.6.1.4.1.89.108.3.1.6

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

1.3.6.1.4.1.89.108.3.1.6.{unit}

Пример просмотра показаний температурного датчика

Команда CLI:
```
show power inline
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.108.3.1.6.1
```

Просмотр лимита мощности на интерфейсе PoE

MIB: rlPoe.mib

Используемые таблицы: rlpethPsePortOperPowerLimit — 1.3.6.1.4.1.89.108.1.1.9

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

Пример просмотра лимита мощности на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show power inline GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.108.1.1.9.1.50
```

Просмотр значения мощности на интерфейсе PoE

MIB: rfc3621.mib

Используемые таблицы: pethPsePortActualPower — 1.3.6.1.2.1.105.1.1.1.15

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

Пример просмотра значения мощности на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show power inline GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.105.1.1.1.15.1.50
```

Просмотр значения тока на интерфейсе PoE

MIB: rlPoe.mib

Используемые таблицы: rlpethPsePortOutputCurrent — 1.3.6.1.4.1.89.108.1.1.4

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

Пример просмотра значения тока на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show power inline GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.108.1.1.4.1.50
```

Просмотр значения напряжения на интерфейсе PoE

MIB: rlPoe.mib

Используемые таблицы: rlpethPsePortOutputVoltage — 1.3.6.1.4.1.89.108.1.1.3

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

Пример просмотра значения напряжения на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show power inline GigabitEthernet1/0/2
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.108.1.1.3.1.50
```

Отключение Power over Ethernet на порте

MIB: rfc3621.mib

Используемые таблицы: pethPsePortAdminEnable — 1.3.6.1.2.1.105.1.1.1.3

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.2.1.105.1.1.1.3.{unit}.{ifindex} i {auto(1), never(2)}
```

Пример отключения PoE на порте `GigabitEthernet1/0/2`

Команда CLI:

```
interface GigabitEthernet1/0/2
power inline never
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.105.1.1.1.3.1.50 i 2
```

16 ФУНКЦИИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

16.1 *Функции обеспечения защиты портов*

Ограничение количества MAC-адресов, изучаемых на Ethernet-портах

MIB: rlinterfaces.mib

Используемые таблицы: 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}
```

Пример ограничения в 20 MAC-адресов на порт GigabitEthernet 1/0/2

Команда CLI:

```
interface GigabitEthernet1/0/2
 port security max 20
```

Команда SNMP:

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

Включение port security

MIB: rlinterfaces.mib

Используемые таблицы: 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),
действие над пакетом, не попавшим под правила port security} \
 1.3.6.1.4.1.89.43.6.1.5.1 i {true(1), false(2). Для отправки трапов} \
 1.3.6.1.4.1.89.43.6.1.6.1 i {частота отправки трапов (сек)} \
 1.3.6.1.4.1.89.43.6.1.2.1 x {ifindex в виде битовой маски}
```

Пример настройки port security для интерфейсов GigabitEthernet 1/0/1-2

Команда CLI:

```
interface range GigabitEthernet 1/0/1-2
 port security discard trap 30
```

Команда SNMP:

```
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 "000000000000C0"
```

Методика расчета битовой маски приведена в разделе «Приложение А. Методика расчета битовой маски».

Установка режима работы port security

MIB: rlinterfaces.mib

Используемые таблицы: 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)}
```

Пример настройки режима ограничения по количеству изученных MAC-адресов на порте GigabitEthernet 1/0/2

Команда CLI:
```
interface GigabitEthernet 1/0/2
 port security mode max-addresses
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.37.50 i 2
```

Просмотр статуса port security

MIB: rlinterfaces.mib

Используемые таблицы: 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
```

Пример просмотра статуса port security

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

Просмотр типа port security

MIB: rlinterfaces.mib

Используемые таблицы: 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
```

Пример просмотра типа port security

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

Просмотр максимально заданного количества MAC-адресов, изучаемых на Ethernet портах

MIB: rlinterfaces.mib

Используемые таблицы: 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
```

Пример просмотра максимально заданного количества MAC адресов, изучаемых на Ethernet-портах

Команда CLI:
```
show ports security
```

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

Перевод порта в режим изоляции и внутри группы портов

MIB: rlprotectedport.mib

Используемые таблицы: 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)}
```

Пример настройки изоляции на портах GigabitEthernet 1/0/1 и GigabitEthernet 1/0/2

Команда CLI:
```
interface range GigabitEthernet 1/0/1-2
switchport protected-port
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.132.1.1.1.49 i 2 \
1.3.6.1.4.1.89.132.1.1.1.50 i 2
```

Настройка отправки трафика на uplink-port

MIB: RADLAN-vlan-MIB

Используемые таблицы: vlanPrivateEdgeStatus — 1.3.6.1.4.1.89.48.37.1.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.48.37.1.1.{Ifindex} i {ifindex} \
1.3.6.1.4.1.89.48.37.1.2.{Ifindex} i {createandGo(4),destroy(6)}
```

Пример

Команда CLI:
```
interface GigabitEthernet 1/0/6
switchport protected GigabitEthernet 1/0/8
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.48.37.1.1.54 i 56 \
1.3.6.1.4.1.89.48.37.1.2.54 i 4
```

Создание статической привязки в MAC-таблице

MIB: Q-BRIDGE-MIB

Используемые таблицы: 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). Байты MAC-адреса
разделяются точками}.{ifIndex} i {other(1), invalid(2), permanent(3),
deleteOnReset(4), deleteOnTimeout(5)}
```

Пример привязки MAC-адреса 00:22:68:7d:0f:3f в vlan 622 к интерфейсу Gigabitethernet1/0/2 в режиме secure (По дефолту используется режим permanent)

Команда CLI:
```
mac address-table static 00:22:68:7d:0f:3f vlan 622 interface
gigabitethernet1/0/2 secure
```

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

Просмотр MAC-таблицы

MIB: Q-BRIDGE-MIB

Используемые таблицы: 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
```

Пример

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

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

Создание статической привязки в arp-таблице

MIB: RFC1213-MIB

Используемые таблицы: 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
```

Пример привязки ip 192.168.1.21 и MAC aa:bb:cc:dd:ee:ff к vlan 1

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

Команда SNMP:
```
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. Для удаления привязки необходимо в поле 1.3.6.1.2.1.4.22.1.4 присвоить значение 2.

2. IP-адрес устройства и IP-адрес создаваемой статической записи в arp-таблице должны находиться в одной подсети.

Просмотр arp-таблицы

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

Используемые таблицы:
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
```

Пример просмотра arp-таблицы

Команда CLI:
```
show arp
```

Команда SNMP:
```
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. Значение таблицы pNetToMediaPhysAddress отображает IP-адрес и MAC-адрес vlan

2. Значение таблицы dot1qTpFdbEntry — отображает статус и идентификационный номер порта, с которого доступно устройство

16.2 *Контроль протокола DHCP и опция 82*

Включение/выключение DHCP сервера на коммутаторе

MIB: rldhcp.mib

Используемые таблицы: 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)}
```

Пример включения DHCP сервера на коммутаторе

Команда CLI:
```
ip dhcp server
```

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

Просмотр записей таблицы dhcp snooping

MIB: rlBridgeSecurity.mib

Используемые таблицы: 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
```

Пример просмотра таблицы dhcp snooping

Команда CLI:
```
Show ip dhcp snooping binding
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.112.1.11.1
```

Включение/выключение dhcp snooping глобально

MIB: rlbridge-security.mib

Используемые таблицы: 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)}
```

Пример глобального включения dhcp snooping

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

Включение/выключение dhcp snooping во vlan

MIB: rlbridge-security.mib

Используемые таблицы: 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)}
```

Пример включения dhcp snooping в vlan 622

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

Настройка ip DHCP information option

MIB: rlbridgesecurity.mib

Используемые таблицы: 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)}
```

Пример

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

Настройка доверенного порта dhcp

MIB: rlbridge-security.mib

Используемые таблицы: 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)}
```

Пример настройки доверенного интерфейса GigabitEthernet 1/0/2

Команда CLI:
```
interface GigabitEthernet 1/0/2
 ip dhcp snooping trust
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.1.13.1.2.50 i 4
```

Настройка DHCP relay в vlan

MIB: rldhcp.mib

Используемые таблицы:
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 {битовая маска}
```

Пример настройки ip DHCP relay enable на vlan 1

Команда CLI:
```
Interface vlan 1
Ip dhcp relay enable
```

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

Пример настройки ip DHCP relay enable на 1026 vlan

Команда CLI:
```
Interface vlan 1026
Ip dhcp relay enable
```

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

Пример расчета битовой маски можно посмотреть в разделе «Приложение А. Методика расчета битовой маски».

16.3 *Защита IP-адреса клиента (IP source Guard)*

Включение/выключение ip source guard глобально

MIB: rlbridge-security.mib

Используемые таблицы: 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)}
```

Пример глобального включения ip source guard

Команда CLI:
```
ip source-guard
```

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

Создание статической привязки ip source guard

MIB: rlbridge-security.mib

Используемые таблицы: 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. Каждый байт MAC-адреса
отделяется от предыдущего точкой} a {ip address (DEC)} \
 1.3.6.1.4.1.89.112.1.10.1.4.{vlan id}.{MAC in DEC. Каждый байт MAC-адреса
отделяется от предыдущего точкой} i {ifIndex} \
 1.3.6.1.4.1.89.112.1.10.1.5.{vlan id}.{MAC in DEC. Каждый байт MAC-адреса
отделяется от предыдущего точкой} i {createAndGo(4), destroy(6)}
```

Пример привязки MAC адреса 00:11:22:33:44:55 к IP 192.168.1.34, vlan 622, интерфейсу GigabitEthernet 1/0/9

Команда CLI:
```
ip source-guard binding 00:11:22:33:44:55 622 192.168.1.34 GigabitEthernet 1/0/9
```

Команда SNMP:
```
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 57 \
1.3.6.1.4.1.89.112.1.10.1.5.622.0.17.34.51.68.85 i 4
```

<u>Включение/выключение ip source guard на порте</u>

MIB: rlbridge-security.mib

Используемые таблицы: 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)}
```

Пример включения ip source guard на интерфейсе GigabitEthernet 1/0/9

Команда CLI:
```
interface GigabitEthernet 1/0/9
 ip source-guard
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.2.5.1.2.57 i 4
```

<u>Включение/выключение ip source guard во vlan</u>

MIB: eltBridgeSecurity.mib

Используемые таблицы: eltIpSourceGuardPortTable — 1.3.6.1.4.1.35265.1.23.112.8.1, 1.3.6.1.4.1.35265.1.23.112.8.1.1.1 — vlan1-1024, 1.3.6.1.4.1.35265.1.23.112.8.1.1.2 — vlan1025-2048, 1.3.6.1.4.1.35265.1.23.112.8.1.1.3 — vlan2049-3072, 1.3.6.1.4.1.35265.1.23.112.8.1.1.4 — vlan3073-4094

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.112.8.1.1.1.<ifindex> x {битовая маска}
```

Пример включения ip source guard на интерфейсе GigabitEthernet 1/0/1 во Vlan 2

Команда CLI:
```
interface GigabitEthernet 1/0/9
 ip source-guard vlan 2
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.112.8.1.1.1.49 x
"4000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000"
```

16.4 *Контроль протокола ARP (ARP Inspection)*

<u>Включение/выключение arp inspection глобально</u>

MIB: rlbridge-security.mib

Используемые таблицы: 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)}
```

Пример глобального включения arp inspection

Команда CLI:
```
ip arp inspection
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.3.2.0 i 1
```

Включение/выключение arp inspection во vlan

MIB: rlbridge-security.mib

Используемые таблицы: 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)}
```

Пример включения arp inspection в vlan 622

Команда CLI:
```
ip arp inspection vlan 622
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.3.6.1.3.622 i 4
```

Настройка доверенного порта arp inspection

MIB: rlbridge-security.mib

Используемые таблицы: 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)}
```

Пример настройки доверенного интерфейса GigabitEthernet 1/0/2

Команда CLI:
```
interface GigabitEthernet 1/0/2
 ip arp inspection trust
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.112.3.7.1.2.50 i 4
```

Привязка ip arp inspection list к vlan

MIB: rlbridge-security.mib

Используемые таблицы: 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}
```

Пример привязки листа с именем test к vlan 622

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

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

16.5 *Проверка подлинности клиента на основе порта (802.1x)*

Включение аутентификации 802.1X на коммутаторе

MIB: dot1xPaeSystem.mib

Используемые таблицы: 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)}
```

Пример включения 802.1x

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

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

Включение периодической повторной проверки подлинности (переаутентификации) клиента

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

Используемые таблицы: 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)}
```

Пример включения периодической повторной проверки подлинности клиента на интерфейсе Gigabitethernet 1/0/2

Команда CLI:
```
interface gigabitethernet 1/0/2
dot1x reauthentication
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.0.8802.1.1.1.1.2.1.1.13.50 i 1
```

Установка периода между повторными проверками подлинности

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

Используемые таблицы: 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}
```

Пример установки периода в 300 сек между повторными проверками на интерфейсе Gigabitethernet 1/0/2

Команда CLI:
```
interface gigabitethernet 1/0/2
dot1x timeout reauth-period 300
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.0.8802.1.1.1.1.2.1.1.12.50 u 300
```

Настройка режимов аутентификации 802.1X на интерфейсе

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

Используемые таблицы: 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)}
```

Пример настройки аутентификации 802.1X в режиме auto на интерфейсе Gigabitethernet 1/0/2

Команда CLI:
```
interface gigabitethernet 1/0/2
dot1x port-control auto
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.0.8802.1.1.1.1.2.1.1.6.50 i 2
```

Включение аутентификации, основанной на MAC-адресах пользователей

MIB: radlan-dot1x-mib.mib

Используемые таблицы: 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)}
```

Пример включения аутентификации, основанной только на MAC-адресах на интерфейсе GigabitEthernet 1/0/3

Команда CLI:
```
interface gigabitethernet 1/0/3
dot1x authentication mac
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.95.10.1.1.51 i 3
```

Разрешение наличия одного/нескольких клиентов на авторизованном порте 802.1X

MIB: rlInterfaces.mib

Используемые таблицы: 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)}
```

Пример разрешения наличия нескольких клиентов на интерфейсе Gigabitethernet 1/0/3

Команда CLI:
```
interface Gigabitethernet 1/0/3
dot1x host-mode multi-sessions
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.43.1.1.30.51 i 3
```

Включение одного или двух методов проверки подлинности, авторизации и учета (AAA) для использования на интерфейсах IEEE 802.1x

MIB: rlaaa.mib

Используемые таблицы: 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, каждая буква отделяется от
следующей точкой} s {authentication list} \1.3.6.1.4.1.89.97.1.1.2.7.{“default”
in DEC, каждая буква отделяется от следующей точкой} i {Deny(0), radius(1),
none(2)} \
1.3.6.1.4.1.89.97.1.1.3.7.{“default” in DEC, каждая буква отделяется от
следующей точкой} i {Deny(0), radius(1), none(2)} \
1.3.6.1.4.1.89.97.1.1.7.7.{“default” in DEC, каждая буква отделяется от
следующей точкой} i 1
```

Пример включения списка RADIUS-серверов для аутентификации пользователя

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

Команда SNMP:
```
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) Для того, чтобы вернуться к настройкам по умолчанию, достаточно значения поменять на Deny(0).

2) Default переводится из ASCII в HEX с помощью таблицы, которую можно найти по ссылке https://ru.wikipedia.org/wiki/ASCII

Добавление указанного сервера в список используемых RADIUS серверов

MIB: rlaaa.mib

Используемые таблицы: 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)}
```

Пример

Команда CLI:
```
radius-server host 192.168.1.10 encrypted key da90833f59be
```

Команда SNMP:
```
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
```

16.6 *Механизм обнаружения петель (loopback-detection)*

Глобальное включение loopback-detection

MIB: rllbd.mib

Используемые таблицы: 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) }
```

Пример глобального включения loopback-detection

Команда CLI:
```
loopback-detection enable
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.127.1.0 i 1
```

Изменение интервала loopback-detection

MIB: rllbd.mib

Используемые таблицы: 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 }
```

Пример изменения интервала loopback-фреймов на 23 секунды

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

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

Изменение режима работы loopback-detection

MIB: rllbd.mib

Используемые таблицы: 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) }
```

Пример изменения режима работы loopback-detection на source-mac-addr

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

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

Включение/отключение loopback-detection на интерфейсах

MIB: rllbd.mib

Используемые таблицы: 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)}
```

Пример включения loopback-detection на интерфейсе TengigabitEthernet1/0/2

Команда CLI:
```
interface TengigabitEthernet1/0/2
loopback-detection enable
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.127.4.1.1.106 i 1
```

Просмотр рабочего состояния loopback-detection на интерфейсе

MIB: rllbd.mib

Используемые таблицы: 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}
```

Пример просмотра состояния loopback-detection на интерфейсе GigabitEthernet1/0/2

Команда CLI:
```
show loopback-detection GigabitEthernet1/0/2
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.127.4.1.2.50
```

При использовании snmp команды:

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

Просмотр заблокированных VLAN в режиме vlan-based

MIB: rllbd.mib

Используемые таблицы: 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}
```

Пример просмотра состояния vlan 2 на порте GigabitEthernet1/0/2

Команда CLI:
```
show loopback-detection GigabitEthernet1/0/2
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.127.4.1.3.50.2
```

Возможные состояния:

1 — active,
2 — blocked

16.7 *Контроль широковещательного шторма (storm-control)*

Настройка storm-control на интерфейсе

MIB: radlan-mib.mib

Используемые таблицы: 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)}
```

Пример включения storm-control для broadcast-трафика на интерфейсе GigabitEthernet1/0/1

Команда CLI:
```
interface GigabitEthernet1/0/1
storm-control broadcast kbps 10000 trap shutdown
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.12.1.3.49.1 i 1 \
1.3.6.1.4.1.89.77.12.1.2.49.1 u 1000 \
1.3.6.1.4.1.89.77.12.1.4.49.1 i 4
```

Пример отключения storm-control для broadcast-трафика на интерфейсе GigabitEthernet1/0/1

Команда CLI:
```
interface GigabitEthernet1/0/1
 no storm-control broadcast
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.12.1.2.49.1 u 0
```

Включить/выключить storm-control для unknown unicast трафика

MIB: radlan-stormctrl.mib

Используемые таблицы: 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,отключить (0)}
```

Пример включения контроля неизвестного одноадресного трафика до 50 кбит/с

Команда CLI:
```
interface GigabitEthernet1/0/2
storm-control unicast Kbps 50
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.12.1.2.50.5 u 50
```

17 КОНФИГУРИРОВАНИЕ IP И MAC ACL (СПИСКИ КОНТРОЛЯ ДОСТУПА)

Создание mac access-list

MIB: qosclimib.mib

Используемые таблицы: 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)}
```

Пример создания MAC ACL с индексом 207

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

Команда SNMP:
```
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
```

Создание ip access-list (ACL)

MIB: qosclimib.mib

Используемые таблицы: 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)}
```

Пример создания IP ACL с индексом 107

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

Команда SNMP:
```
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
```

Пример наполнения ACL правилами подробно рассмотрен в разделе «Приложение Б: Пример создания типового IP ACL».

Привязка IP или MAC ACL к порту

MIB: qosclimib.mib

Используемые таблицы:
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
```

Пример: назначаем правило с индексом 107 (название ACL 7-ip) на порт GigabitEthernet 1/0/2

Команда CLI:
```
interface GigabitEthernet 1/0/2
 service-acl input 7-ip
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.13.1.14.50.2 i 107 \
1.3.6.1.4.1.89.88.13.1.13.50.2 i 1
```

Для удаления ACL с порта достаточно индекс ACL заменить на 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

Привязка IP и MAC ACL к порту

MIB: qosclimib.mib

Используемые таблицы:
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
```

Пример: назначаем правило с индексом 107 и 207 (название ACL 7-ip для IP ACL и 7-mac для MAC ACL) на порт GigabitEthernet 1/0/2 (Ifindex 50)

Команда CLI:
```
interface GigabitEthernet 1/0/2
 service-acl input 7-mac 7-ip
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.13.1.14.50.2 i 207 \
1.3.6.1.4.1.89.88.13.1.20.50.2 i 107 \
1.3.6.1.4.1.89.88.13.1.13.50.2 i 1
```

Для удаления ACL с порта достаточно индекс IP и MAC ACL заменить на 0.

**snmpset -v2c -c private 192.168.1.30 **
**1.3.6.1.4.1.89.88.13.1.14.50.2 i 0 **
**1.3.6.1.4.1.89.88.13.1.20.50.2 i 0 **

1.3.6.1.4.1.89.88.13.1.13.50.2 i 1

Создание policy-map и привязка к нему ACL

MIB: qosclimib.mib

Используемые таблицы:
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

Схема: создание policy-map проводится в несколько запросов

1. Создаем class и назначаем ему свойства
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. Создаем policy-map и включаем его
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. Привязываем class-map к 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. Создаем ограничение скорости для 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. Привязываем ограничение скорости к 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. Задаем значение метки трафику DSCP и/или cos, указываем выходную очередь
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.88.5.1.1.{index-of-class}.{setDSCP(3), setQueue(4), setCos(5)} i {setDSCP(3), setQueue(4), setCos(5)} \
1.3.6.1.4.1.35265.1.23.88.5.1.2.{index-of-class}.{setDSCP(3), setQueue(4), setCos(5)} i {Mark value of DSCP/queue/cos(DEC)} \
1.3.6.1.4.1.35265.1.23.88.5.1.3.{index-of-class}.{setDSCP(3), setQueue(4), setCos(5)} i {createAndGo(4), destroy(6)}

Пример: IP ACL с index-of-acl = 107 привязывается к class-map с именем test и выставляется метка DSCP = 36(DEC), cos = 4 и queue = 8 для трафика, подпавшего под IP ACL. Class test привязывается к policy-map с именем test1

Команда CLI:

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

Команда SNMP:

```
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 \
1.3.6.1.4.1.35265.1.23.88.5.1.1.20.3 i 3 \
1.3.6.1.4.1.35265.1.23.88.5.1.2.20.3 i 36 \
1.3.6.1.4.1.35265.1.23.88.5.1.3.20.3 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.88.5.1.1.20.4 i 4 \
1.3.6.1.4.1.35265.1.23.88.5.1.2.20.4 i 8 \
1.3.6.1.4.1.35265.1.23.88.5.1.3.20.4 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.88.5.1.1.20.5 i 5 \
1.3.6.1.4.1.35265.1.23.88.5.1.2.20.5 i 4 \
1.3.6.1.4.1.35265.1.23.88.5.1.3.20.5 i 4
```

Назначение Policy-map на порт

MIB: qosclimib.mib

Используемые таблицы: 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}
```

Пример: назначаем policy-map с индексом 1 на порт GigabitEthernet 1/0/3

Команда CLI:

```
interface GigabitEthernet 1/0/3
 service-policy input test1
```

Команда SNMP:

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

18 КОНФИГУРАЦИЯ ЗАЩИТЫ ОТ DOS-АТАК

Включение security-suite

MIB: rlSecuritySuiteMib

Используемые таблицы:
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)}
```

Пример включения класса команд security-suite для всех правил

Команда CLI:
```
security-suite enable
```

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

Настройка режима работы security-suite

MIB: rlSecuritySuiteMib

Используемые таблицы: 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)}
```

Пример включения режима работы "report"

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

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

Выключить защиту от tcp-пакетов с одновременно установленными SYN- и FIN- флагами

MIB: rlSecuritySuiteMib

Используемые таблицы: 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)}
```

Пример включения режима report

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

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
```

19 КАЧЕСТВО ОБСЛУЖИВАНИЯ — QOS

19.1 *Настройка QoS*

Ограничение исходящей скорости на Ethernet-портах

MIB: qosclimib.mib

Используемые таблицы: 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.{ifindex порта}.2 i {disable(1),enable
(1)} \
 1.3.6.1.4.1.89.88.13.1.7.{ifindex порта}.2 i {traffic-shape} \
 1.3.6.1.4.1.89.88.13.1.8.{ifindex порта}.2 i {Burst size in bytes}
```

Пример: ограничить исходящую скорость на порте до значения 20Мбит/с

Команда CLI:
```
interface GigabitEthernet 1/0/1
traffic-shape 20480 500000
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.13.1.6.49.2 i 2 \
1.3.6.1.4.1.89.88.13.1.7.49.2 i 20480 \
1.3.6.1.4.1.89.88.13.1.8.49.2 i 500000
```

Ограничение входящей скорости на Ethernet-портах

MIB: radlan-mib.mib

Используемые таблицы: 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}
```

Пример: ограничить входящую скорость на интерфейсе GigabitEthernet 1/0/1 до значения 10Мбит/с

Команда CLI:
```
interface GigabitEthernet 1/0/1
rate-limit 10240 burst 500000
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.12.1.2.49.6 u 10240 \
1.3.6.1.4.1.89.77.12.1.5.49.6 u 500000
```

Для отключения rate-limit на интерфейсе необходимо выполнить (на примере интерфейса GigabitEthernet1/0/1):

snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.77.12.1.2.49.6 u 0 1.3.6.1.4.1.89.77.12.1.5.49.6 u 128000

Создание профиля qos tail-drop и расширение дескрипторов для очередей

MIB: eltQosTailDropMIB.mib

Используемые таблицы: 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.{Номер профиля (1-4)}.{номер очереди(1-8)}
i {size (0-400)}
```

Пример

Команда CLI:
```
qos tail-drop profile 2
queue 1 limit 400
```

Команда SNMP:
```
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 400
```

Чтобы вернуться к настройкам по умолчанию достаточно установить значение равным 12

Установка размера пакетного разделяемого пула для порта

MIB: eltQosTailDropMIB.mib

Используемые таблицы: 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{номер профиля(1-4)} i {size (0-400)}
```

Пример

Команда CLI:
```
qos tail-drop profile 2
port-limit 400
```

Команда SNMP:
```
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 400
```

Назначение созданного профиля на интерфейс

MIB: eltQosTailDropMIB.mib

Используемые таблицы: 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 {номер профиля (1-4)}
```

Пример

Команда CLI:
```
interface GigabitEthernet 1/0/1
qos tail-drop profile 2
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.12.1.1.2.1.1.49 i 2
```

Просмотр отображения глобальных лимитов, дескрипторов, буферов

MIB: radlan-mib.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show qos tail-drop
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.12.1.1.3
```

Просмотр таблицы вывода текущих аллоцированных ресурсов qos (лимитов, дескрипторов, буферов)

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

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show qos tail-drop
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.12.1.2.1
```

Перемаркировка DSCP в COS

MIB: eltQosclimib.mib

Используемые таблицы: eltQosCos — 1.3.6.1.4.1.35265.1.23.88.6.1.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.88.6.1.2.{метка DSCP} i {метка COS}
```

Пример настройки перемаркировки DSCP 30 в метку 5 COS

Команда CLI:
```
qos map dscp-cos 30 to 5
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.88.6.1.2.30 i 5
```

19.2 *Статистика QoS*

Просмотр Tail Drop-счетчиков по очередям

MIB: eltMesCounters.mib

Используемые таблицы: 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.{Dropped packets(5), Passed
packets(7)}.{ifIndex}.{1-8}.0
```

Пример просмотра счетчиков для первой очереди

Команда CLI:

```
show interface GigabitEthernet1/0/6
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.1.8.1.2.1.1.1.7.54.1.0
```

20 МАРШРУТИЗАЦИЯ

20.1 *Статическая маршрутизация*

Просмотр таблицы маршрутизации

MIB: radlan-mib.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show ip route
```

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

Просмотр статических маршрутов

MIB: rlip.mib

Используемые таблицы: 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
```

Пример

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

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

20.2 *Динамическая маршрутизация*

Просмотр соседства OSPF

MIB: rlip.mib

Используемые таблицы: 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
```

Пример

Команда CLI:
```
show ip ospf neighbor
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.89.210.11
```

ПРИЛОЖЕНИЕ А. МЕТОДИКА РАСЧЕТА БИТОВОЙ МАСКИ

Битовые маски состоят из 128 байт (шестнадцатеричных разрядов всего 256).
Каждый разряд обозначает четыре VLAN/порта. По номеру VLAN/порта определяется нужное поле.

<u>Пример 1</u>
Записать битовую маску для интерфейсов GigabitEthernet 1/0/20-21:

- для 1G интерфейсов ifIndex начинается с 49;
- для порта gi1/0/20 ifIndex равен 68, для gi1/0/21 - 69.

Определение номера разряда:

68/4=17 69/4=17,25 (Каждый разряд отвечает за 4 ifIndex. При делении ifindex на 4 для определения № разряда для записи, полученное значение округляется в большую сторону).

Если нам нужны порты gi0/20-21 (ifindex 68,69), то они должны быть записаны в 17 и 18 поле.

В двоичной последовательности 17 поле будет записано следующим образом 0001 (Последняя 1 - 68 индекс). Переводим в HEX, получаем 1.

В двоичной последовательности 18 поле будет записано следующим образом 1000 (Первая 1 - 69 индекс). Переводим в HEX, получаем 8.

Итого в битовой маске будет 16 нулей, 1, 8: 000000000000000018.

<u>Пример 2</u>
Записать битовую маску для vlan 622, 3100.

- 622/4=155,5 (Каждый 0 отвечает за 4 vlan. При делении vlan на 4 для определения № поля для записи округление всегда идет вверх).
  Если нам нужен vlan 622, то он должен быть записан в 156 поле.
  В двоичной последовательности 156 поле будет записано следующим образом: 0100 (вторая 1 - 622 vlan). Переводим в HEX, получаем 4.
  Итого в битовой маске будет 155 нулей и 4:
  000000000000000000000000000000000000000000000000000000000000000000000000
  000000000000000000000000000000000000000000000000000000000000000000000000
  000000000004

- 3100/4=775
  Требуется принять во внимание, что для указания номера VLAN берутся таблицы
  rldot1qPortVlanStaticEgressList1to1024
  rldot1qPortVlanStaticEgressList1025to2048
  rldot1qPortVlanStaticEgressList2049to3072
  rldot1qPortVlanStaticEgressList3073to4094
  Так как наш vlan попадает в 4 таблицу, то 775-256*3=7.
  Vlan 3100 будет записан в 7 поле данной таблицы.
  В двоичной последовательности 7 поле будет записано следующим образом: 0001 (Последняя 1 - 3100 vlan). Переводим в HEX, получаем 1.
  Итого в битовой маске будет 6 нулей и 1: 0000001.

ПРИЛОЖЕНИЕ Б: ПРИМЕР СОЗДАНИЯ ТИПОВОГО IP ACL

В данном приложении рассмотрен пример наполнения IP ACL с index-of-acl = 107 правилами вида:

```
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 any any 224.0.0.0 15.255.255.255 ace-priority 60
 permit ip any any 37.193.119.7 0.0.0.0 any ace-priority 80
 permit ip any any 10.130.8.3 0.0.0.0 any ace-priority 100
 permit ip any any 192.168.0.0 0.0.0.15 any ace-priority 120
 permit ip 00:19:16:15:14:16 00:00:00:00:00:00 any 37.193.119.7 0.0.0.0 any
ace-priority 140
 permit ip any 01:00:0c:00:00:00 00:00:00:ff:ff:ff any any ace-priority 160
exit
```

Создание правила deny udp any bootps any bootpc

MIB: qosclimib.mib

Используемые таблицы: rlQosTupleTable — 1.3.6.1.4.1.89.88.5, rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 1} i {protocol(1)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 1} x {protocol index (HEX)} \
 1.3.6.1.4.1.89.88.5.1.3.{значение поля 1} i {Значение в таблице порта для
протокола = 0. Константа для этого правила} \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 2} i {udp-port-src(6)} \
 1.3.6.1.4.1.89.88.5.1.3.{значение поля 2} i {Number of source port (DEC)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 2} x {source ip(HEX)} \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 3} i { udp-port-dst(6)} \
 1.3.6.1.4.1.89.88.5.1.3.{значение поля 3} i {Number of dst port (DEC)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 3} x {dst ip(HEX)}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 1} \
 1.3.6.1.4.1.89.88.31.1.7.{index-of-acl}.{index-of-rule} i {значение поля 3} \
 1.3.6.1.4.1.89.88.31.1.9.{index-of-acl}.{index-of-rule} i {значение поля 2}
```

Пример добавления правила deny udp any bootps any bootpc в IP ACL 7-ip (т.к. предполагается, что правило первое по счету, то index-of-rule=20)

Команда CLI:

```
ip access-list extended 7-ip
 deny udp any bootps any bootpc ace-priority 20
exit
```

Команда SNMP:

```
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 \
1.3.6.1.4.1.89.88.31.1.3.107.20 i 2 \
1.3.6.1.4.1.89.88.31.1.4.107.20 i 3 \
1.3.6.1.4.1.89.88.31.1.5.107.20 i 1 \
1.3.6.1.4.1.89.88.31.1.7.107.20 i 2 \
1.3.6.1.4.1.89.88.31.1.9.107.20 i 3
```

Создание правила permit igmp any any

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 4} i {protocol(1)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 4} x {protocol index (HEX)}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 4}
```

Пример добавления правила permit igmp any any в IP ACL 7-ip (т.к. предполагается, что правило второе по счету, то index-of-rule=40)

Команда CLI:

```
ip access-list extended 7-ip
 permit igmp any any ace-priority 40
exit
```

Команда SNMP:

```
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 \
1.3.6.1.4.1.89.88.31.1.3.107.40 i 1 \
1.3.6.1.4.1.89.88.31.1.4.107.40 i 8 \
1.3.6.1.4.1.89.88.31.1.5.107.40 i 4
```

Создание правила deny ip any any any 224.0.0.0 15.255.255.255

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 5} i {ip-dest(3)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 5} x {dst ip +wildcard mask (HEX)}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 5}
```

Пример добавления правила deny ip any any any 224.0.0.0 15.255.255.255 в IP ACL 7-ip (т.к. предполагается, что правило третье по счету, то index-of-rule=60)

Команда CLI:
```
ip access-list extended 7-ip
 deny ip any any any 224.0.0.0 15.255.255.255 ace-priority 60
exit
```

Команда SNMP:
```
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 \
1.3.6.1.4.1.89.88.31.1.3.107.60 i 2 \
1.3.6.1.4.1.89.88.31.1.4.107.60 i 1 \
1.3.6.1.4.1.89.88.31.1.5.107.60 i 5
```

Создание правила permit ip any any 37.193.119.7 0.0.0.0 any

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 6} i {ip-source(2)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 6} x {source ip +wildcard mask (HEX)}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 6}
```

Пример добавления правила permit ip any any 37.193.119.7 0.0.0.0 any в IP ACL 7-ip (т.к. предполагается, что правило четвертое по счету, то index-of-rule=80)

Команда CLI:

```
ip access-list extended 7-ip
 permit ip any any 37.193.119.7 0.0.0.0 any ace-priority 80
exit
```

Команда SNMP:

```
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 \
1.3.6.1.4.1.89.88.31.1.3.107.80 i 1 \
1.3.6.1.4.1.89.88.31.1.4.107.80 i 1 \
1.3.6.1.4.1.89.88.31.1.6.107.80 i 6
```

Создание правила permit ip any any 10.130.8.3 0.0.0.0 any

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 7} i {ip-source(2)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 7} x {source ip +wildcard mask (HEX)}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 7}
```

Пример добавления правила permit ip any any 10.130.8.3 0.0.0.0 any в IP ACL 7-ip (т.к. предполагается, что правило пятое по счету, то index-of-rule=100)

Команда CLI:

```
ip access-list extended 7-ip
 permit ip any any 10.130.8.3 0.0.0.0 any ace-priority 100
exit
```

Команда SNMP:

```
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 \
1.3.6.1.4.1.89.88.31.1.3.107.100 i 1 \
1.3.6.1.4.1.89.88.31.1.4.107.100 i 1 \
1.3.6.1.4.1.89.88.31.1.6.107.100 i 7
```

Создание правила permit ip any any 192.168.0.0 0.0.0.15 any

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила.

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 8} i {ip-source(2)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 8} x {source ip +wildcard mask (HEX)}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 8}
```

Пример добавления правила permit ip any any 192.168.0.0 0.0.0.15 any в IP ACL 7-ip (т.к. предполагается, что правило шестое по счету, то index-of-rule=120)

Команда CLI:

```
ip access-list extended 7-ip
 permit ip any any 192.168.0.0 0.0.0.15 any ace-priority 120
exit
```

Команда SNMP:

```
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 \
1.3.6.1.4.1.89.88.31.1.3.107.120 i 1 \
1.3.6.1.4.1.89.88.31.1.4.107.120 i 1 \
1.3.6.1.4.1.89.88.31.1.6.107.120 i 8
```

Создание правила permit ip 00:19:16:15:14:16 00:00:00:00:00:00 any 37.193.119.7 0.0.0.0 any

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 9} i {ip-source(2)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 9} x {source ip +wildcard mask (HEX)} \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 10} i {mac-src(10)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 10} x {source mac +wildcard mask (HEX)}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 9} \
 1.3.6.1.4.1.89.88.31.1.6.{index-of-acl}.{index-of-rule} i {значение поля 10}
```

Пример добавления правила permit ip 00:19:16:15:14:16 00:00:00:00:00:00 any 37.193.119.7 0.0.0.0 any в IP ACL 7-ip (т.к. предполагается, что правило седьмое по счету, то index-of-rule=140)

Команда CLI:

```
ip access-list extended 7-ip
 permit ip 00:19:16:15:14:16 00:00:00:00:00:00 any 37.193.119.7 0.0.0.0 any ace-
priority 140
exit
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.9 i 2 \
1.3.6.1.4.1.89.88.5.1.4.9 x "0x25 C1 77 07 00 00 00 00" \
1.3.6.1.4.1.89.88.5.1.2.10 i 10 \
1.3.6.1.4.1.89.88.5.1.4.10 x "0x001916151416000000000000"

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.107.140 i 1 \
1.3.6.1.4.1.89.88.31.1.4.107.140 i 1 \
1.3.6.1.4.1.89.88.31.1.5.107.140 i 9 \
1.3.6.1.4.1.89.88.31.1.6.107.140 i 10
```

Создание правила permit ip any 01:00:0c:00:00:00 00:00:00:ff:ff:ff any any

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила.

```
snmpset -v2c -c <community> <IP address> \
 .{значение поля 11} i {mac-dest (11)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 11} x {dst mac +wildcard mask (HEX)}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 11}
```

Пример добавления правила permit ip any 01:00:0c:00:00:00 00:00:00:ff:ff:ff any any в IP ACL 7-ip (т.к. предполагается, что правило восьмое по счету, то index-of-rule=160)

Команда CLI:

```
ip access-list extended 7-ip
 permit ip any 01:00:0c:00:00:00 00:00:00:ff:ff:ff any any ace-priority 160
exit
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.11 i 11 \
1.3.6.1.4.1.89.88.5.1.4.11 x "0x01000c000000000000ffffff"

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.107.160 i 1 \
1.3.6.1.4.1.89.88.31.1.4.107.160 i 1 \
1.3.6.1.4.1.89.88.31.1.5.107.160 i 11
```

ПРИЛОЖЕНИЕ В: ПРИМЕР СОЗДАНИЯ, НАПОЛНЕНИЯ И УДАЛЕНИЯ OFFSET-LIST С ПРИВЯЗКОЙ К MAC ACL

В данном приложении рассмотрен пример создания и наполнения MAC ACL с index-of-acl = 207 правилами вида:

```
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
```

Создание mac access-list

MIB: qosclimib.mib

Используемые таблицы: 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)}
```

Пример создания MAC ACL с индексом 207

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

Команда SNMP:
```
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
```

Создание offset-list

MIB: qosclimib.mib

Используемые таблицы:
rlQosOffsetTable — 1.3.6.1.4.1.89.88.4
eltMesQosCliMib — 1.3.6.1.4.1.35265.1.23.88

Пример создания offset-list PADO l2 12 00 88 l2 13 00 63 l2 15 00 07:

Создание правила производится в два запроса.

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.4.1.2.{значение поля 1 в offset-list} i {layer2-start(2) -
указание заголовка пакета или отдельных параметров заголовка} \
1.3.6.1.4.1.89.88.4.1.3.{значение поля 1 в offset-list} i {Порядковый номер
байта в заголовке} \
1.3.6.1.4.1.89.88.4.1.4.{значение поля 1 в offset-list} i {WildcardMask в байте
в DEC} \
1.3.6.1.4.1.89.88.4.1.5.{значение поля 1 в offset-list} i {Значение байта с
учетом WildcardMask в DEC} \
```

```
1.3.6.1.4.1.89.88.4.1.7.{значение поля 1 в offset-list} i {createAndGo(4),
destroy(6)}
1.3.6.1.4.1.89.88.4.1.2.{значение поля 2 в offset-list} i {layer2-start(2) -
указание заголовка пакета или отдельных параметров заголовка} \
1.3.6.1.4.1.89.88.4.1.3.{значение поля 2 в offset-list} i {Порядковый номер
байта в заголовке} \
1.3.6.1.4.1.89.88.4.1.4.{значение поля 2 в offset-list} i {WildcardMask в байте
в DEC} \
1.3.6.1.4.1.89.88.4.1.5.{значение поля 2 в offset-list} i {Значение байта с
учетом WildcardMask в DEC} \
1.3.6.1.4.1.89.88.4.1.7.{значение поля 2 в offset-list} i {createAndGo(4),
destroy(6)}

1.3.6.1.4.1.89.88.4.1.2.{значение поля 3 в offset-list} i {layer2-start(2) -
указание заголовка пакета или отдельных параметров заголовка} \
1.3.6.1.4.1.89.88.4.1.3.{значение поля 3 в offset-list} i {Порядковый номер
байта в заголовке} \
1.3.6.1.4.1.89.88.4.1.4.{значение поля 3 в offset-list} i {WildcardMask в байте
в DEC} \
1.3.6.1.4.1.89.88.4.1.5.{значение поля 3 в offset-list} i {Значение байта с
учетом WildcardMask в DEC} \
1.3.6.1.4.1.89.88.4.1.7.{значение поля 3 в offset-list} i {createAndGo(4),
destroy(6)}
```

2. Привязка offset-list по названию к индексу ACL (index-of-acl).

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.88.1.1.1.{index-of-acl}.{Количество букв в названии
offset-list}.{Название offset-list in DEC, каждая буква названия отделяется от
следующей точкой} i {index-of-acl}
1.3.6.1.4.1.35265.1.23.88.1.1.3.{index-of-acl}.{Количество букв в названии
offset-list}.{Название offset-list in DEC, каждая буква названия отделяется от
следующей точкой} i {значение поля 1 в offset-list} \
1.3.6.1.4.1.35265.1.23.88.1.1.4.{index-of-acl}.{Количество букв в названии
offset-list}.{Название offset-list in DEC, каждая буква названия отделяется от
следующей точкой} i {значение поля 2 в offset-list} \
1.3.6.1.4.1.35265.1.23.88.1.1.5.{index-of-acl}.{Количество букв в названии
offset-list}.{Название offset-list in DEC, каждая буква названия отделяется от
следующей точкой} i {значение поля 3 в offset-list} \
1.3.6.1.4.1.35265.1.23.88.1.1.8.{index-of-acl}.{Количество букв в названии
offset-list}.{Название offset-list in DEC, каждая буква названия отделяется от
следующей точкой} i {createAndGo(4), destroy(6)}
```

Пример добавления правила deny udp any bootps any bootpc в MAC ACL 7-mac (т.к. предполагается, что правило первое по счету, то index-of-rule=20)

Команда CLI:

```
mac access-list extended 7-mac
offset-list PADO l2 12 00 88 l2 13 00 63 l2 15 00 07
exit
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.4.1.2.1 i 2 \
1.3.6.1.4.1.89.88.4.1.3.1 i 12 \
1.3.6.1.4.1.89.88.4.1.4.1 i 0 \
1.3.6.1.4.1.89.88.4.1.5.1 i 136 \
1.3.6.1.4.1.89.88.4.1.7.1 i 4 \
1.3.6.1.4.1.89.88.4.1.2.2 i 2 \
1.3.6.1.4.1.89.88.4.1.3.2 i 13 \
1.3.6.1.4.1.89.88.4.1.4.2 i 0 \
1.3.6.1.4.1.89.88.4.1.5.2 i 99 \
1.3.6.1.4.1.89.88.4.1.7.2 i 4 \
```

```
1.3.6.1.4.1.89.88.4.1.2.3 i 2 \
1.3.6.1.4.1.89.88.4.1.3.3 i 15 \
1.3.6.1.4.1.89.88.4.1.4.3 i 0 \
1.3.6.1.4.1.89.88.4.1.5.3 i 7 \
1.3.6.1.4.1.89.88.4.1.7.3 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.88.1.1.1.207.4.80.65.68.79 i 207 \
1.3.6.1.4.1.35265.1.23.88.1.1.3.207.4.80.65.68.79 i 1 \
1.3.6.1.4.1.35265.1.23.88.1.1.4.207.4.80.65.68.79 i 2 \
1.3.6.1.4.1.35265.1.23.88.1.1.5.207.4.80.65.68.79 i 3 \
1.3.6.1.4.1.35265.1.23.88.1.1.8.207.4.80.65.68.79 i 4
```

Название offset-list переводится из ASCII в HEX с помощью таблицы, которую можно найти по ссылке https://ru.wikipedia.org/wiki/ASCII

Создание правила deny any any offset-list PADO

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable —1.3.6.1.4.1.89.88.31

Создание правила проводится в два запроса:

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.88.5.1.2.{значение поля 1 в ACL} i {general(15)} \
1.3.6.1.4.1.89.88.5.1.2.{значение поля 2 в ACL} i {general(15)} \
1.3.6.1.4.1.89.88.5.1.2.{значение поля 2 в ACL} i {general(15)} \
1.3.6.1.4.1.89.88.5.1.3.{значение поля 1 в ACL} i {значение поля 1 в offset-
list} \
1.3.6.1.4.1.89.88.5.1.3.{значение поля 2 в ACL} i {значение поля 2 в offset-
list} \
1.3.6.1.4.1.89.88.5.1.3.{значение поля 3 в ACL} i {значение поля 3 в offset-
list} \
1.3.6.1.4.1.89.88.5.1.5.{значение поля 1 в ACL} i {createAndGo(4), destroy(6)}
\
1.3.6.1.4.1.89.88.5.1.5.{значение поля 2 в ACL} i {createAndGo(4), destroy(6)}
\
1.3.6.1.4.1.89.88.5.1.5.{значение поля 3 в ACL} i {createAndGo(4), destroy(6)}
2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl)
как 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 {mac(5)} \
1.3.6.1.4.1.89.88.31.1.5.{index-of-acl}.{index-of-rule} i {значение поля 1 в
ACL} \
1.3.6.1.4.1.89.88.31.1.6.{index-of-acl}.{index-of-rule} i {значение поля 2 в
ACL} \
1.3.6.1.4.1.89.88.31.1.7.{index-of-acl}.{index-of-rule} i {значение поля 3 в
ACL}
```

Пример добавления правила deny any any offset-list PADO в MAC ACL 7-mac (т.к. предполагается, что правило первое по счету, то index-of-rule=20)

Команда CLI:

```
mac access-list extended 7-mac
deny any any offset-list PADO ace-priority 20
```

```
exit
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.1 i 15 \
1.3.6.1.4.1.89.88.5.1.2.2 i 15 \
1.3.6.1.4.1.89.88.5.1.2.3 i 15 \
1.3.6.1.4.1.89.88.5.1.3.1 i 1 \
1.3.6.1.4.1.89.88.5.1.3.2 i 2 \
1.3.6.1.4.1.89.88.5.1.3.3 i 3 \
1.3.6.1.4.1.89.88.5.1.5.1 i 4 \
1.3.6.1.4.1.89.88.5.1.5.2 i 4 \
1.3.6.1.4.1.89.88.5.1.5.3 i 4

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.31.1.3.207.20 i 2 \
1.3.6.1.4.1.89.88.31.1.4.207.20 i 5 \
1.3.6.1.4.1.89.88.31.1.5.207.20 i 1 \
1.3.6.1.4.1.89.88.31.1.6.207.20 i 2 \
1.3.6.1.4.1.89.88.31.1.7.207.20 i 3
```

Создание правила в MAC ACL на основе EtherType

MIB: qosclimib.mib

Используемые таблицы:
rlQosTupleTable — 1.3.6.1.4.1.89.88.5
rlQosAceTidxTable — 1.3.6.1.4.1.89.88.31

Схема: создание правила проводится в два запроса.

1. Задаются параметры правила

```
snmpset -v2c -c <community> <IP address> \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 1} i {mac-src(10), mac-dest(11),
vlan(12)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 1} x {protocol index (HEX)} \
 1.3.6.1.4.1.89.88.5.1.3.{значение поля 1} i {Значение в таблице порта для
протокола = 0. Константа для этого правила} \
 1.3.6.1.4.1.89.88.5.1.2.{значение поля 2} i {ether-type(17)} \
 1.3.6.1.4.1.89.88.5.1.3.{значение поля 2} i {ether-type (DEC)} \
 1.3.6.1.4.1.89.88.5.1.4.{значение поля 2} x {Нулевое поле - константа}
```

2. Привязка правила по индексу (index-of-rule) к ACL по индексу (index-of-acl) как 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 {значение поля 1} \
.1.3.6.1.4.1.89.88.31.1.9.{index-of-acl}.{index-of-rule} i {значение поля 2}
```

Пример добавления правила permit 00:1f:c6:8b:c6:8a 00:00:00:00:00:00 any 806 0000 в MAC ACL 7-mac (т.к. предполагается, что правило первое по счету, то index-of-rule=20)

Команда CLI:

```
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
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.88.5.1.2.1 i 10 \
1.3.6.1.4.1.89.88.5.1.4.1 x "0x001fc68bc68a000000000000" \
1.3.6.1.4.1.89.88.5.1.3.1 i 0 1.3.6.1.4.1.89.88.5.1.2.2 i 17 \
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 \
1.3.6.1.4.1.89.88.31.1.3.207.20 i 1 \
1.3.6.1.4.1.89.88.31.1.4.207.20 i 5 \
1.3.6.1.4.1.89.88.31.1.5.207.20 i 1 \
1.3.6.1.4.1.89.88.31.1.9.207.20 i 2
```

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Для получения технической консультации по вопросам эксплуатации оборудования ООО «Предприятие «ЭЛТЕКС» вы можете обратиться в Сервисный центр компании:

Форма обратной связи на сайте: **https://eltex-co.ru/support/**
Servicedesk: **https://servicedesk.eltex-co.ru/**

На официальном сайте компании вы можете найти техническую документацию и программное обеспечение для продукции ООО «Предприятие «ЭЛТЕКС», обратиться к базе знаний, оставить интерактивную заявку или проконсультироваться у инженеров Сервисного центра на техническом форуме:

Официальный сайт компании: **https://eltex-co.ru/**
Технический форум: **https://eltex-co.ru/forum**
База знаний: **https://docs.eltex-co.ru/display/EKB/Eltex+Knowledge+Base**
Центр загрузок: **https://eltex-co.ru/support/downloads**