Комплексные решения для построения сетей

Мониторинг и управление Ethernet коммутаторами MES по SNMP

Версия документа	Дата выпуска	Содержание изменений
Версия 1.0	09.03.2016	Первая публикация.
Версия программного обеспечения	MES1000, MES2000 – **1.1.44** MES3000 – **2.5.44** MES5000 – **2.2.10**	

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 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
MES1024	- индексы 1-24 - fastethernet 1/0/1-24; - индексы 49-50 - gigabitethernet 1/0/1-2 (до версии 1.1.16 соответствие было следующим: индексы 73-74 - gigabitethernet 1/0/1-2); - индексы 1000-1007 - Port-Channel 1/0/1-8; - индексы 100000-104095 - VLAN 1-4096.
MES1124 MES1124M MES1124M DC	- индексы 1-24 - fastethernet 1/0/1-24; - индексы 49-52 - gigabitethernet 1/0/1-4 (до версии 1.1.16 соответствие было следующим: индексы 73-76 - gigabitethernet 1/0/1-4); - индексы 1000-1007 - Port-Channel 1/0/1-8; - индексы 100000-104095 - VLAN 1-4096.
MES2124 MES2124M MES2124MB MES2124P MES2124P rev. B MES2124P rev. C	- индексы 49-76 - gigabitethernet 1/0/1-28; - индексы 1000-1007 - Port-Channel 1/0/1-8; - индексы 100000-104095 - VLAN 1-4096.

2. Коммутаторы агрегации

Модель коммутатора	Индексы
MES3108 MES3108F MES3116 MES3116F MES3124 MES3124F MES3124F rev.B	- индексы 105-108 - tengigabitethernet 1/0/1-4; - индексы 49-72 - gigabitethernet 1/0/1-24; - индексы 1000-1011 - Port-Channel 1/0/1-12; - индексы 100000-104095 - VLAN 1-4096.

3. Коммутаторы для ЦОД

Модель коммутатора	Индексы
MES5148 MES5248	- индексы 1-48 - tengigabitethernet 1/0/1-48; - индексы 1000-1031 - Port-Channel 1/0/1-32; - 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 serve*r –** 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
```

3.2 Работа с TFTP-сервером

Копирование конфигурации из энергозависимой памяти на TFTP-сервер

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 {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://192.168.1.30/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://192.168.1.30/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://192.168.1.30/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: 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 {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 с TFTP-сервера в startup-config:

Команда CLI:

```
copy tftp://192.168.1.30/MES-config.cfg startup-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 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

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

MIB: rlcopy.mib, RADLAN-DEVICEPARAMS-MIB.mib
Используемые таблицы: rlCopyEntry - 1.3.6.1.4.1.89.87.2.1, rndActiveSoftwareFileAfterReset - 1.3.6.1.4.1.89.2.13.1.1.3

Проходит в 2 этапа

1. Загрузка образа ПО:

```
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 s "{ip address 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: загрузить firmware на flash коммутатора

Команда CLI:
```
copy tftp://192.168.1.30/MES3000-2544.ros image
```

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
"MES3000-2514.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 ПО коммутатора:

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

Пример: установить в качестве активного образа image-1

Команда CLI:
```
boot system image-1
```

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
```

Обновление boot коммутатора

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 s "{ip address of tftp server}"
 1.3.6.1.4.1.89.87.2.1.6.1 s "{boot 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 {boot (9)}
 1.3.6.1.4.1.89.87.2.1.17.1 i {createAndGo (4)}
```

Пример загрузки boot на flash коммутатора:

Команда CLI:

```
copy tftp://192.168.1.30/boot-0012.rfb boot
```

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
"boot-0012.rfb" 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
9 1.3.6.1.4.1.89.87.2.1.17.1 i 4
```

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 (для указания моментальной перезагрузки требуется указать значение ***t=0***)

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
```

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
```

Возможные варианты:

image1(1)
image2(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
```

Примечание: 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 версии начального загрузчика

MIB: RADLAN-DEVICEPARAMS-MIB.mib
Используемые таблицы: rndBaseBootVersion - 1.3.6.1.4.1.89.2.10

```
snmpwalk -v2c -c <community> <IP address>
 1.3.6.1.4.1.89.2.10
```

Пример:

Команда CLI:
```
show version
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.2.10
```

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 version
```

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 максимального количества tcam правил

MIB: rlsysmng.mib
Используемые таблицы: rlSysmngTcamAllocPoolSize - 1.3.6.1.4.1.89.204.1.1.1.6

```
snmpwalk -v2c -c <community> <IP address>
 1.3.6.1.4.1.89.204.1.1.1.6
```

Пример:

Команда CLI:
```
show system resources tcam
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.204.1.1.1.6
```

Просмотр используемого количества tcam правил

MIB: rlsysmng.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
```

Пример:

Команда CLI:
```
show system resources tcam
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.204.1.1.1.5
```

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 resources routing
```

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: rlIpFftNumStnRoutesNumber - 1.3.6.1.4.1.89.47.1.12.1.2

```
snmpwalk -v2c -c <community> <IP address>
 1.3.6.1.4.1.89.204.1.1.1.5
```

Пример:

Команда CLI:
```
show system resources routing
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.204.1.1.1.5
```

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 resources routing
```

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 resources routing
```

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 resources routing
```

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 resources routing
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.26.23
```

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 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
```

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: rlSctCpuRateTrafficTypeTable - 1.3.6.1.4.1.89.203.3

```
snmpwalk -v2c -c <community> <ip address>
 1.3.6.1.4.1.89.203.3.1
```

Пример:

Команда CLI:

```
show cpu input-rate detailed
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.203.3.1
```

Примечание:

1. Привязка индексов к типам трафика:

rlCPUTypeHttp(2)
rlCPUTypeTelnet(3)
rlCPUTypeSsh(4)
rlCPUTypeSnmp(5)
rlCPUTypeIp(6)
rlCPUTypeArp(7)
rlCPUTypeArpInspec(8)
rlCPUTypeStp(9)
rlCPUTypeIeee(10)
rlCPUTypeRouteUnknown(13)
rlCPUTypeIpHopByHop(14)
rlCPUTypeMtuExceeded(15)
rlCPUTypeIpv4Multicast(16)
rlCPUTypeIpv6Multicast(17)
rlCPUTypeDhcpSnooping(18)

rlCPUTypeIgmpSnooping(21)
rlCPUTypeMldSnooping(22)
rlCPUTypeTtlExceeded(23)
rlCPUTypeIpv4IllegalAddress(24)
rlCPUTypeIpv4HeaderError(25)
rlCPUTypeIpDaMismatch(26)
rlCPUTypeSflow(27)
rlCPUTypeLogDenyAces(28)

2. Количество обрабатываемых CPU в секунду пакетов (по типам трафика) можно посмотреть в:

rlSctCpuRateTrafficTypeRate - 1.3.6.1.4.1.89.203.3.1.2

Счетчик по типам трафика можно посмотреть в:

rlSctCpuRateTrafficTypeCounter - 1.3.6.1.4.1.89.203.3.1.3

Изменение лимитов CPU

MIB: eltSwitchRateLimiterMIB.mib
Используемые таблицы: eltCPURateLimiterTable - 1.3.6.1.4.1.35265.1.23.1.773.1.1

```
snmpset -v2c -c <community> <ip address>
 1.3.6.1.4.1.35265.1.23.1.773.1.1.1.2.{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 1.3.6.1.4.1.35265.1.23.1.773.1.1.1.2.4
i 512
```

Примечание: 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 последнюю секунду: 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 последнюю секунду:

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: mibs\rlmng.mib
Используемые таблицы: rlCpuTasksUtilEnable - 1.3.6.1.4.1.89.1.15

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

Пример:

Команда CLI:
```
service tasks-utilization
```
Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.1.15.0 i 1
```

Мониторинг загрузки CPU по процессам

MIB: rlmng.mib
Используемые таблицы: rlCpuTasksUtilValuesEntry - 1.3.6.1.4.1.89.1.16.1.1

- Загрузка за 5 секунд: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.1.16.1.1.3
- Загрузка за 1 минуту: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.1.16.1.1.4
- Загрузка за 5 минут: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.1.16.1.1.5

Пример просмотра загрузки по процессам за последние 5 секунд:

Команда CLI:
```
show tasks utilization
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.1.16.1.1.3
```

Примечание: привязка индексов к процессам

ROOT(0)
ROOT(1)
ROOT(2)
ROOT(3)
ROOT(4)
IDLE(5)
SYLG(6)
CDB(7)
SNMP(8)
SNPR(9)
DDFG(10)
CNLD(11)
IOTG(12)
IOUR(13)
IOTM(14)
IOD(15)
HDEB(16)
HOST(17)
WDHI(18)
WDLO(19)
TBI_(20)
BRMN(21)
TMNG(22)
COPY(23)
MROR(24)
SFTR(25)
SFMG(26)
HCLT(27)
EVLC(28)
SELC(29)
EVAU(30)
SEAU(31)
ESTC(32)
SSTC(33)
PLCT(34)
PLCR(35)
SWTR(36)
DSPT(37)
OUIs(38)
BOXS(39)
BSNC(40)
BOXM(41)
B_RS(42)
TRMT(43)
DACT(44)
SW2M(45)
exRX(46)
OAM(47)
3SMA(48)
3SWF(49)
3SWQ(50)
POLI(51)
NTST(52)
NTPL(53)
L2HU(54)
L2PS(55)
L2SC(56)
LBDR(57)
SFSM(58)
NSCT(59)
L2UT(60)
BRGS(61)
FLNK(62)
FFTT(63)
KEYM(64)
IPAT(65)
IP6C(66)
IP6M(67)
RPTS(68)
ARPG(69)
IPG(70)
ICMP(71)
TFTP(72)
IPRD(73)
DNSC(74)
PNGA(75)
UDPR(76)
VRRP(77)
TRCE(78)
SSLP(79)
WBSR(80)
GOAH(81)
TNSR(82)
TNSL(83)
SSHP(84)
SSHU(85)
SNTP(86)
IGMP(87)
TUNT(88)
PTPT(89)
FTPM(90)
FTPD(91)
IPSL(92)
SQIN(93)
XMOD(94)
SOCK(95)
AAAT(96)
AATT(97)
SCPT(98)
BTPC(99)
SETX(100)
EVTX(101)
SERX(102)
EVRX(103)
HLTX(104)
LACP(105)
GRN_(106)
CFM(107)
BFD(108)
TRIG(109)
MACT(110)
TCPP(111)
DHCP(112)
IPMT(113)
MSCm(114)
STSA(115)
STSB(116)
STSC(117)
STSD(118)
STSE(119)
STSF(120)
STSG(121)
STSH(122)
STSI(123)

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
```

Примечание: возможны следующие состояния

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

Мониторинг статуса и степени заряда АКБ (только для коммутаторов MES1124MB, MES2124MB)

MIB: eltEnv.mib
Используемые таблицы: eltEnvMonBatteryStatusTable - 1.3.6.1.4.1.35265.1.23.11.1

```
snmpwalk -v2c -c <community> <IP address>
 1.3.6.1.4.1.35265.1.23.11.1
```

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
```

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:

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
```

Примечание: возможны следующие состояния

normal (1)
notFunctioning (6)

Контроль показаний температурных датчиков (для MES3000)

MIB: rlphysdescription.mib
Используемые таблицы: rlPhdUnitEnvParamTable - 1.3.6.1.4.1.89.53.15

- Температурный датчик 1: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.53.15.1.9
- Температурный датчик 2: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.53.15.1.12
- Температурный датчик 3: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.53.15.1.14

Пример просмотра температуры датчика 3:

Команда CLI:
```
show system
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.53.15.1.14
```

Примечание:

1. Для MES5000 мониторинг температурных датчиков проводится в OID 1.3.6.1.4.1.89.83.2.1.1.1.1.
2. В коммутаторах доступа MES1000/MES2000 установлен только один температурный датчик, имеющий OID 1.3.6.1.4.1.89.53.15.1.9.

Контроль состояния температурных датчиков (для MES3000)

MIB: rlphysdescription.mib
Используемые таблицы: rlPhdUnitEnvParamTable - 1.3.6.1.4.1.89.53.15

- Температурный датчик 1: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.53.15.1.10
- Температурный датчик 2: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.53.15.1.13
- Температурный датчик 3: snmpwalk -v2c -c <community> <ip address> 1.3.6.1.4.1.89.53.15.1.15

Пример просмотра состояния температурного датчика 3:

Команда CLI:
```
show system
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.53.15.1.15
```

Примечание: состояния температурных датчиков

ok (1)
unavailable (2)
nonoperational (3)

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
```

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: Просмотр Description GigabitEthernet 1/0/1 коммутатора MES3124

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
```

Установка 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 GigabitEthernet 0/2:

Команда CLI:

```
interface GigabitEthernet 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
```

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

Примечание: 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 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: Просмотр статуса порта GigabitEthernet 1/0/1 коммутатора MES3124

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
```

Примечание: возможные варианты

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

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: Просмотр статуса порта GigabitEthernet 1/0/1 коммутатора MES3124

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
```

Примечание: возможные варианты

up (1)
down (2)
testing (3)
unknown (4)
dormant (5)
notPresent (6)
lowerLayerDown (7)

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:

Команда CLI:
```
show interfaces status
```
Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.43.1.1.7.60
```

Примечание: возможные результаты

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

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
```

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.{parametr}{ifIndex}
```

Примечание: 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 Настройка интерфейса VLAN

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.{vlan id}
```

Пример: gросмотр Description vlan 100

Команда CLI:

```
show interfaces description vlan 100
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.2.1.17.7.1.4.3.100
```

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 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
```

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 0/2 в режим trunk:

Команда CLI:

```
interface GigabitEthernet 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
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000004
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.48.68.1.2.50 x
00000000000000000000
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.48.68.1.3.50 x
00000000000000000000
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 0/2 в качестве native vlan:

Команда CLI:

```
interface GigabitEthernet 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
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000004
```

```
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.48.68.1.5.50 x
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000004
```

Примечание:

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 Приложении А.

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

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
```

Пример просмотра vlan на порту GigabitEthernet 1/0/1:

Команда CLI:
show running-config interfaces GigabitEthernet 1/0/1

Команда SNMP:
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.48.68

Примечание: выведенные значения представляют из себя битовую маску, методика расчета которой приведена в Приложении А.

7 НАСТРОЙКА IPV4-АДРЕСАЦИИ

Установка IP адреса на interface vlan:

MIB: rlip.mib
Используемые таблицы: rsIpAdEntIfIndex - 1.3.6.1.4.1.89.26.1.1.2

```
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
```

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 10.10.10.24 1.3.6.1.4.1.89.76.3.1.2.100029 i 4
```

8 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
```

9 ПРОТОКОЛ EAPS

Просмотр состояния кольца

MIB: RADLAN-BRIDGE**MIB**OBJECTS-**MIB**.mib
Используемые таблицы: RlEapsRingState - 1.3.6.1.4.1.89.57.9.5.1.10

```
snmpwalk -v2c -c <community> <IP address>
 1.3.6.1.4.1.89.57.9.5.1.10
```

Пример:

Команда CLI:
```
show eaps
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.57.9.5.1.10
```

10 ГРУППОВАЯ АДРЕСАЦИЯ

10.1 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>
 1.3.6.1.4.1.89.116.5.1.7.{vlan id}.1.4.{ip address(DEC)}.1.4.0.0.0.0 x
{Битовая маска для интерфейса gi 0/1(методику расчета битовой маски
смотреть в начале документа)} 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 порту 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
1.3.6.1.4.1.89.116.5.1.7.622.1.4.239.200.200.17.1.4.0.0.0.0 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
```

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 gi
0/20,21(методику расчета битовой маски смотреть в начале документа)}"
```

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

Команда CLI:

```
interface range GigabitEthernet 1/0/21-22
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"
```

Примечание: для удаления настройки надо заменить соответствующие портам поля в битовой маске на 0.

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
```

10.2 Функция посредника протокола IGMP (IGMP Snooping)

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
```

Примечание: 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.

10.3 Функции ограничения multicast-трафика

Создание multicast snooping profile

MIB: eltIpMulticast.mib
Используемые таблицы: eltIgmpMldSnoopProfileTable - 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: eltIgmpMldSnoopFilterTable - 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 указания 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
```

Примечание: ***index of rule*** - считается по сумме всех правил во всех профилях

Назначение multicast snooping profile на порт

MIB: eltIpMulticast.mib
Используемые таблицы: eltIgmpMldSnoopIfProfileExtEntry - 1.3.6.1.4.1.35265. 1.23.46.7.1

```
snmpset -v2c -c <community> <ip>
 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 порту

MIB: eltIpMulticast.mib
Используемые таблицы: eltIgmpMldSnoopIfMaxGroupsEntry - 1.3.6.1.4.1.35265.1.23.46.6.1

```
snmpset -v2c -c <community> <ip>
 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
```

11 ФУНКЦИИ УПРАВЛЕНИЯ

11.1 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 HEX, 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 HEX, 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 HEX, каждая буква
логина отделяется от следующей точкой} i {privelege level}
 1.3.6.1.4.1.89.79.17.1.4.{number of letters}.{Login in HEX, каждая буква
логина отделяется от следующей точкой} i {create and go}
```

Пример добавления пользователя *techsup* с паролем *password* и уровнем привелегий 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
```

Примечание:
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 #.

Авторизация enable по enable паролю

MIB: rlaaa.mib
Используемые таблицы: rlAAAMethodType1 - 1.3.6.1.4.1.89.79.15.1.2

```
snmpset -v2c -c <community> <ip address>

1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.99.95.100.101.102.97.1
17.108.116 i {rlAAAMethodDeny(0), rlAAAMethodLinePassword(1),
rlAAAMethodSystemPassword(2), rlAAAMethodLocalUserTable(3),
rlAAAMethodRadius(4), rlAAAMethodTacacs(5), rlAAAMethodSucceed(6)}
```

Пример:

Команда CLI:

```
aaa authentication enable default enable
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30
1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.99.95.100.101.102.97.1
17.108.116 i 2
```

Примечание: 101.110.97.98.108.101.95.99.95.100.101.102.97.117.108.116 переводится из ASCII таблицы (расшифровывается enable_c_default)

12 ЗЕРКАЛИРОВАНИЕ (МОНИТОРИНГ) ПОРТОВ

Настройка зеркалирования портов:

MIB: SMON-**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 по vlan:

MIB: SMON-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
```

13 ФУНКЦИИ ДИАГНОСТИКИ ФИЗИЧЕСКОГО УРОВНЯ

13.1 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 addres>
 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
```

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

Примечание: 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}.4
```

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
```

13.2 Диагностика оптического трансивера

Снятие показаний DDM через SNMP

MIB: rlphy.mib
Используемые таблицы: rlPhyTestGetStatus - 1.3.6.1.4.1.89.90.1.2.1.2

```
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 GigabitEthernet 1/0/2 (для всех параметров):

Команда CLI:
```
show fiber-ports optical-transceiver detailed interface GigabitEthernet
1/0/2
```

Команда SNMP:

```
snmpwalk -v2c -c  public 192.168.1.30 1.3.6.1.4.1.89.90.1.2.1.3.50
```

Примечание:

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.

14 ФУНКЦИИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

14.1 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 adresses}
```

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 {чистота 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"
```

Примечание:

Методика расчета битовой маски приведена в Приложении А.

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 порту 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
```

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> <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 (secure - режим при включении port security. 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
```

14.2 Контроль протокола DHCP и опция 82

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 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 trusted
```
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
```

14.3 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
```

Включение/выключение ip source guard на порту

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
```

14.4 Контроль протокола ARP (ARP Inspection)

Включение/выключение arp inspection глобально

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 192.168.1.10 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
```

15 КОНФИГУРИРОВАНИЕ ACL (СПИСКИ КОНТРОЛЯ ДОСТУПА)

15.1 Конфигурирование ACL на базе IPv4

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
```

Примечание: пример наполнения ACL правилами подробно рассмотрен в Приложении Б.

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 -c -v2c <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 (Ifindex 50)

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

Команда SNMP:

```
snmpset -c private -v2c 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
```

Примечание: для удаления ACL с порта достаточно индекс ACL заменить на 0.
snmpset -c -v2c private 192.168.1.20 .1.3.6.1.4.1.89.88.13.1.14.50.2 i 0
1.3.6.1.4.1.89.88.13.1.13.50.2 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 -c -v2c <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 -c -v2c 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
```

Примечание: для удаления ACL с порта достаточно индекс IP и MAC ACL заменить на 0.
snmpset -c -v2c 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 3 запроса.

1. Создаем class и назначаем ему свойства

```
snmpset -c -v2c <community> <IP address>
 1.3.6.1.4.1.89.88.9.1.2.{index-of-class} "{name-of-class-map}"
 1.3.6.1.4.1.89.88.9.1.3.{index-of-class} i {matchAll (1)}
 1.3.6.1.4.1.89.88.9.1.4.{index-of-class} i {setDSCP (3), setQueue (4),
setCos (5)}
 1.3.6.1.4.1.89.88.9.1.5.{index-of-class} i {Mark value of DSCP/cos(DEC)}
 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 {4 - create and go, 6 -
destroy}
```

2. Создаем policy-map и включаем его

```
snmpset -c -v2c <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 {4 - create and go, 6 -
destroy}
```

3. Привязываем class-map к policy-map

```
snmpset -c -v2c <community> <IP address>
 1.3.6.1.4.1.89.88.39.1.2.1.20 i {index-of-class}
 1.3.6.1.4.1.89.88.39.1.3.1.20 i {index-of-policy-map}
```

Пример: IP ACL с index-of-acl=107 привязывается к class-map с именем test и выставляется метка DSCP=36(DEC) 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
```

Команда SNMP:

```
snmpset -c -v2c 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 1.3.6.1.4.1.89.88.9.1.4.20 i 3
1.3.6.1.4.1.89.88.9.1.5.20 i 36 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 -c -v2c 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 -c -v2c 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
```

Назначение Policy-map на порт

MIB: qosclimib.mib
Используемые таблицы: rlQosIfPolicyMapPointerIn - 1.3.6.1.4.1.89.88.13.1.3

```
snmpset -c -v2c <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 (Ifindex 51)

Команда CLI:

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

Команда SNMP:

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

Просмотр счетчиков пакетов, подпавших под deny ip/mac acl с пометкой log-input

MIB: qosclimib.mib

Используемые таблицы: rlQosDenyAceStatisticsTable - 1.3.6.1.4.1.89.88.42

```
snmpwalk -v2c -c <community> <IP address>
 1.3.6.1.4.1.89.88.42
```

Пример просмотра счетчиков пакетов для интерфейса GigabitEthernet 1/0/12 коммутатора mes3124.

Команда CLI:
```
show interfaces access-lists counters GigabitEthernet 1/0/12
```

Команда SNMP:
```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.88.42.60
```

15.2 Конфигурирование ACL на базе MAC

Создание mac 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 207:

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

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.88.7.1.2.207 s "7-mac"
1.3.6.1.4.1.89.88.7.1.3.207 i 1 1.3.6.1.4.1.89.88.7.1.4.107 i 4
```

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
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
```

16 КАЧЕСТВО ОБСЛУЖИВАНИЯ – QOS

16.1 Настройка QoS

Ограничение исходящей скорости на Ethernet порта*x*

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 {1-disable, 2-enable}
 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 порту до значения 20Мбит/с.

Команда CLI:
```
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: qosclimib.mib
Используемые таблицы: rlQosIfPolicyEntry - 1.3.6.1.4.1.89.88.13.1

```
snmpset -v2c -c <community> <IP address>
1.3.6.1.4.1.89.88.13.1.17.{ifindex порта}.2 i {1-disable, 2-enable}
1.3.6.1.4.1.89.88.13.1.18.{ifindex порта}.2 i {rate-limit}
 1.3.6.1.4.1.89.88.13.1.19.{ifindex порта}.2 i {Burst size in bytes}
```

Пример: ограничить входящую скорость на порту до значения 10Мбит/с

Команда CLI:
```
rate-limit 10240 500000
```

Команда SNMP:
```
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.88.13.1.17.49.2 i 2
1.3.6.1.4.1.89.88.13.1.18.49.2 i 10240 1.3.6.1.4.1.89.88.13.1.19.49.2 i
500000
```

16.2 Статистика QoS

Просмотр Tail drop счетчика qos статистики

MIB: qosclimib.mib
Используемые таблицы: rlQosOutQueueStatisticsCounterTailDropValue –

```
1.3.6.1.4.1.89.88.35.4.1.10.1 для 1 счетчика и
1.3.6.1.4.1.89.88.35.4.1.10.2 для 2 счетчика
snmpwalk -v2c -c <community> <IP address>
 1.3.6.1.4.1.89.88.35.4.1.10.{номер tail drop счетчика}
```

Пример: настроена QoS статистика на 8 очередь интерфейса GigabitEthernet 1/0/1 и на 5 очередь интерфейса GigabitEthernet 1/0/2 коммутатора mes3124.

Команда CLI:

```
qos statistics queues 1 8 all GigabitEthernet 1/0/1
qos statistics queues 1 5 all GigabitEthernet 1/0/2
```

Количество tail drop пакетов в 8 очереди интерфейса GigabitEthernet 1/0/1 можно посмотреть следующим образом:

```
show qos statistics
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.88.35.4.1.10.1
```

Количество tail drop пакетов в 5 очереди интерфейса GigabitEthernet 1/0/2 можно посмотреть следующим образом:

Команда CLI:

```
show qos statistics
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.88.35.4.1.10.2
```

Просмотр Total packets счетчика QoS статистики

MIB: qosclimib.mib
Используемые таблицы: rlQosOutQueueStatisticsCounterAllValue –
1.3.6.1.4.1.89.88.35.4.1.11.1 для 1 счетчика и 1.3.6.1.4.1.89.88.35.4.1.11.2 для 2 счетчика
snmpwalk -v2c -c <community> <IP address>
1.3.6.1.4.1.89.88.35.4.1.11.{номер total packets счетчика}

Пример: Настроена QoS статистика на 8 очередь интерфейса GigabitEthernet 1/0/1 и на 5 очередь интерфейса GigabitEthernet 1/0/2 коммутатора mes3124

Команда CLI:

```
qos statistics queues 1 8 all GigabitEthernet 1/0/1
qos statistics queues 1 5 all GigabitEthernet 1/0/2
```

Количество переданных пакетов в 8 очереди интерфейса GigabitEthernet 1/0/1 можно посмотреть следующим образом:

```
show qos statistics
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.88.35.4.1.11.1
```

Количество переданных пакетов в 5 очереди интерфейса GigabitEthernet 1/0/2 можно посмотреть следующим образом:

Команда CLI:

```
show qos statistics
```

```
Команда SNMP:
snmpwalk -v2c -c public 192.168.1.30 1.3.6.1.4.1.89.88.35.4.1.11.2
```

17 КОНФИГУРАЦИЯ ПРОТОКОЛОВ МАРШРУТИЗАЦИИ

17.1 Конфигурация статической маршрутизации

Просмотр таблицы маршрутизации

MIB: IP-FORWARD-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
```

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.

Пример 1.

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.

Пример 2.

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:
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000004

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
 permit igmp any any
 deny ip any any any 224.0.0.0 15.255.255.255
 permit ip any any 37.193.119.7 0.0.0.0 any
 permit ip any any 10.130.8.3 0.0.0.0 any
 permit ip any any 192.168.0.0 0.0.0.15 any
 permit ip 00:19:16:15:14:16 00:00:00:00:00:00 any 37.193.119.7 0.0.0.0 any
 permit ip any 01:00:0c:00:00:00 00:00:00:ff:ff:ff any any
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
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
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
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
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
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
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
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
"0x00191615141600000000000"
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>
 1.3.6.1.4.1.89.88.5.1.2.{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
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 1.3.6.1.4.1.89.88.31.1.6.107.160 i 11
```

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

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

Российская Федерация ,630020, г. Новосибирск, ул. Окружная, дом 29 в.
Телефон:
+7(383) 274-47-87
+7(383) 272-83-31
E-mail: techsupp@eltex.nsk.ru

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

Официальный сайт компании: http://eltex.nsk.ru
Технический форум: http://eltex.nsk.ru/forum
База знаний: http://eltex.nsk.ru/support/knowledge
Центр загрузок: http://eltex.nsk.ru/support/downloads