

**Коммутаторы магистрального уровня,  
коммутаторы уровня агрегации, коммутаторы уровня доступа**

**MES1024, MES11xx, MES21xx, MES2208P,  
MES31xx, MES5148, MES5248**

**Мониторинг и управление Ethernet-коммутаторами MES по SNMP,  
версии ПО 1.1.48.1, 2.5.48, 2.2.14.2**

Версия документа	Дата выпуска	Содержание изменений
Версия 1.1	13.07.2018	<p>Добавлены разделы:</p> <ul style="list-style-type: none"> <li>4.3 Управление устройством</li> <li>4.4 IP Service Level Agreements (IP SLA)</li> <li>6.3 Настройка и мониторинг errdisable-состояния</li> <li>6.4 Настройка VOICE VLAN</li> <li>9.1 Настройка протокола ERPS</li> <li>9.2 Настройка протокола Spanning-tree</li> <li>9.3 Протокол EAPS</li> <li>14.5 Проверка подлинности клиента на основе порта (802.1x)</li> <li>14.6 Механизм обнаружения петель (loopback-detection)</li> <li>14.7 Контроль широковещательного шторма</li> <li>17.2 Динамическая маршрутизация</li> </ul> <p>ПРИЛОЖЕНИЕ В: ПРИМЕР СОЗДАНИЯ, НАПОЛНЕНИЯ И УДАЛЕНИЯ OFFSET-LIST С ПРИВЯЗКОЙ К MAC ACL</p> <p>Удалены разделы:</p> <ul style="list-style-type: none"> <li>10.2 Функция посредника протокола IGMP (IGMP Snooping)</li> <li>15.2 Конфигурирование ACL на базе MAC</li> </ul>
Версия 1.0	09.03.2016	Первая публикация.
<b>Версия программного обеспечения</b>	MES1000, MES2000 – <b>1.1.48.1</b> MES3000 – <b>2.5.48</b> MES5000 – <b>2.2.14.2</b>	

## СОДЕРЖАНИЕ

1 НАСТРОЙКА SNMP-СЕРВЕРА И ОТПРАВКИ SNMP-TRAP .....	6
2 КРАТКИЕ ОБОЗНАЧЕНИЯ .....	6
3 РАБОТА С ФАЙЛАМИ .....	8
3.1 Сохранение конфигурации .....	8
3.2 Работа с TFTP-сервером .....	9
3.3 Автоконфигурирование коммутатора .....	11
3.4 Обновление программного обеспечения .....	11
4 УПРАВЛЕНИЕ СИСТЕМОЙ .....	15
4.1 Системные ресурсы .....	15
4.2 Системные параметры .....	23
4.3 Управление устройством .....	26
4.4 IP Service Level Agreements (IP SLA) .....	28
5 НАСТРОЙКА СИСТЕМНОГО ВРЕМЕНИ .....	32
6 КОНФИГУРИРОВАНИЕ ИНТЕРФЕЙСОВ .....	34
6.1 Параметры Ethernet-интерфейсов .....	34
6.2 Конфигурирование VLAN .....	42
6.3 Настройка и мониторинг errdisable-состояния .....	45
6.4 Настройка voice vlan .....	47
6.5 Настройка LLDP .....	48
7 НАСТРОЙКА IPV4-АДРЕСАЦИИ .....	50
8 НАСТРОЙКА GREEN ETHERNET .....	52
9 НАСТРОЙКА КОЛЬЦЕВЫХ ПРОТОКОЛОВ .....	53
9.1 Настройка протокола ERPS .....	53
9.2 Настройка протокола Spanning-tree .....	55
9.3 Протокол EAPS .....	56
10 ГРУППОВАЯ АДРЕСАЦИЯ .....	57
10.1 Правила групповой адресации (multicast addressing) .....	57
10.2 Функции ограничения multicast-трафика .....	59
11 ФУНКЦИИ УПРАВЛЕНИЯ .....	62
11.1 Механизм AAA .....	62
12 ЗЕРКАЛИРОВАНИЕ ПОРТОВ .....	65
13 ФУНКЦИИ ДИАГНОСТИКИ ФИЗИЧЕСКОГО УРОВНЯ .....	67
13.1 Диагностика медного кабеля .....	67
13.2 Диагностика оптического трансивера .....	69
14 ФУНКЦИИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ .....	71
14.1 Функции обеспечения защиты портов .....	71
14.2 Контроль протокола DHCP и опция 82 .....	75
14.3 Защита IP-адреса клиента (IP-source Guard) .....	78
14.4 Контроль протокола ARP (ARP Inspection) .....	79
14.5 Проверка подлинности клиента на основе порта (802.1x) .....	80
14.6 Механизм обнаружения петель (loopback-detection) .....	84
14.7 Контроль широковещательного шторма (storm-control) .....	86
15 КОНФИГУРИРОВАНИЕ IP И MAC ACL (СПИСКИ КОНТРОЛЯ ДОСТУПА) .....	89
16 КАЧЕСТВО ОБСЛУЖИВАНИЯ – QoS .....	95
16.1 Настройка QoS .....	95
16.2 Статистика QoS .....	98
17 МАРШРУТИЗАЦИЯ .....	100
17.1 Статическая маршрутизация .....	100
17.2 Динамическая маршрутизация .....	100
ПРИЛОЖЕНИЕ А. МЕТОДИКА РАСЧЕТА БИТОВОЙ МАСКИ .....	102
ПРИЛОЖЕНИЕ Б. ПРИМЕР СОЗДАНИЯ ТИПОВОГО IP ACL .....	103

ПРИЛОЖЕНИЕ В. ПРИМЕР СОЗДАНИЯ, НАПОЛНЕНИЯ И УДАЛЕНИЯ OFFSET-LIST С ПРИВЯЗКОЙ К MAC  
ACL..... 110

## 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.
<i>Курсив Calibri</i>	Курсивом Calibri указываются переменные или параметры, которые необходимо заменить соответствующим словом или строкой.
<b><i>Полужирный курсив</i></b>	Полужирным курсивом выделены примечания и предупреждения.
<b>&lt;Полужирный курсив&gt;</b>	Полужирным курсивом в угловых скобках указываются названия клавиш на клавиатуре.
<i>Courier New</i>	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 MES1124 MES1124M MES1124MB	- индексы 1-24 - fastethernet 1/0/1-24;
MES2124 MES2124M MES2124MB MES2124F	- индексы 49-72 - gigabitethernet 1/0/1-24; - индексы 1000-1015 - Port-Channel 1/0/1-16; - индексы 100000-104095 - VLAN 1-4096.
MES2208P MES2124P	

### 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 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:** rfcopy.mib

**Используемые таблицы:** rfcopyEntry - 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:** rfcopy.mib

**Используемые таблицы:** rfcopyEntry - 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:** rfile.mib

**Используемые таблицы:** rFileRowStatus - 1.3.6.1.4.1.89.96.2.1.6

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.96.2.1.6.102.108.97.115.104.58.47.47.115.116.97.114.116.117.
112.45.99.111.110.102.105.103 i { destroy (6) }
```

#### **Пример удаления startup-config**

**Команда CLI:**

```
delete startup-config
```

**Команда SNMP:**

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.96.2.1.6.102.108.97.115.104.58.47.47.115.116.97.114.116.117.
112.45.99.111.110.102.105.103 i 6
```

## **3.2 Работа с TFTP-сервером**

### Копирование конфигурации из энергозависимой памяти на TFTP-сервер

**MIB:** rcopy.mib

**Используемые таблицы:** rCopyEntry - 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:** rcopy.mib

**Используемые таблицы:** rCopyEntry - 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:** rfcopy.mib

**Используемые таблицы:** rfcopyEntry - 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 rfcopy.mib

**Используемые таблицы:** rfcopyEntry - 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:** rIDhcpCLOption67Enable - 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:** rIcopy.mib, RADLAN-DEVICEPARAMS-MIB.mib

**Используемые таблицы:** rICopyEntry - 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 a "{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/mes2000_205_r318.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 " mes2000_205_r318.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), image2(2)}
```

**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:** rfcopy.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

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: (1) - image1

(2) - image2

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 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 серийного номера коммутатора

**MIB:** rIphysdescription.mib

**Используемые таблицы:** rIPhdUnitGenParamSerialNum - 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 максимального количества tcam правил

**MIB:** rlsysmng.mib

**Используемые таблицы:** rISysmngTcamAllocPoolSize - 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:** rISysmngTcamAllocInUseEntries - 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 информации о загрузке tcam

**MIB:** OBJECT-TYPE.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 для MES3124

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



На коммутаторах серии MES1000/MES2000 данная функция аппаратно не поддерживается.

### 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.{ifindex}.1
```

#### Пример для MES3124

Команда CLI:  
show system resources routing

Команда SNMP:  
snmpwalk -v2c -c public 192.168.1.30 \  
1.3.6.1.4.1.89.204.1.1.1.5.5.116.99.97.109.49.1



На коммутаторах серии MES1000/MES2000 данная функция аппаратно не поддерживается.



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 для MES3124**

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



**На коммутаторах серии MES1000/MES2000 данная функция аппаратно не поддерживается.**

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 для MES3124**

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



**На коммутаторах серии MES1000/MES2000 данная функция аппаратно не поддерживается.**

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 для MES3124**

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



На коммутаторах серии MES1000/MES2000 данная функция аппаратно не поддерживается.

Просмотр используемого количества IP-интерфейсов

**MIB:** rIip.mib

**Используемые таблицы:** rIIPAddressesNumber - 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 для MES3124**

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



На коммутаторах серии MES1000/MES2000 данная функция аппаратно не поддерживается.

Просмотр системного MAC-адреса коммутатора

**MIB:** rIphysdescription.mib

**Используемые таблицы:** rIPhdStackMacAddr - 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> \
  1.3.6.1.4.1.89.203.3.1.2.{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 \
  1.3.6.1.4.1.89.203.3.1.2
```



**1. Привязка индексов к типам трафика:**

- rICPUTypeHttp(2)
- rICPUTypeTelnet(3)
- rICPUTypeSsh(4)
- rICPUTypeSnmp(5)
- rICPUTypeIp(6)
- rICPUTypeArp(7)
- rICPUTypeArpInspec(8)
- rICPUTypeStp(9)
- rICPUTypeIeee(10)
- rICPUTypeRouteUnknown(13)
- rICPUTypeIpHopByHop(14)
- rICPUTypeMtuExceeded(15)
- rICPUTypeIpv4Multicast(16)
- rICPUTypeIpv6Multicast(17)
- rICPUTypeDhcpSnooping(18)
- rICPUTypeIcmpSnooping(21)
- rICPUTypeMldSnooping(22)
- rICPUTypeTtlExceeded(23)
- rICPUTypeIpv4IllegalAddress(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
```



**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)  
eltCPURLTypeIcmpSnoop(16)  
eltCPURLTypeMldSnoop(17)  
eltCPURLTypeSflow(18)  
eltCPURLTypeLogDenyAces(19)  
eltCPURLTypeIpErrors(20)  
eltCPURLTypeOther(22)

**Мониторинг загрузки CPU**

**MIB:** rlmng.mib

**Используемые таблицы:** rICpuUtilDuringLastSecond - 1.3.6.1.4.1.89.1.7, rICpuUtilDuringLastMinute - 1.3.6.1.4.1.89.1.8, rICpuUtilDuringLast5Minutes - 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:** rlmng.mib

**Используемые таблицы:** rICpuTasksUtilEnable - 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 tasksutilization
```

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:** rICpuTasksUtilValuesEntry - 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
```



Привязка индексов к процессам

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

### Включение поддержки сверхдлинных кадров (jumbo-frames)

**MIB:** radlan-jumboframes-mib.mib

**Используемые таблицы:** rJumboFrames - 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:** rPhysDescription.mib

**Используемые таблицы:** rPhdUnitEnvParamTable - 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:** rlEnvMonSupplyStatusState — 1.3.6.1.4.1.89.83.1.2.1.3

```
snmpwalk -v2c -c <community> <IP address> \  
1.3.6.1.4.1.89.83.1.2.1.3
```

**Пример:**

```
Команда CLI:  
show system battery  
  
Команда SNMP:  
snmpwalk -v2c -c public 192.168.1.30 \  
1.3.6.1.4.1.89.83.1.2.1.3
```



**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

### Мониторинг уровня заряда АКБ

**MIB:** eltEnv.mib

**Используемые таблицы:** rlEnvMonSupplyStatusDescr — 1.3.6.1.4.1.89.83.1.2.1.2

```
snmpwalk -v2c -c <community> <IP address> \  
1.3.6.1.4.1.89.83.1.2.1.2
```

**Пример**

```
Команда CLI:  
show system battery  
  
Команда SNMP:  
snmpwalk -v2c -c public 192.168.1.30 \  
1.3.6.1.4.1.89.83.1.2.1.2
```

### 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 для MES3124**

```
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:** rphysdescription.mib

**Используемые таблицы:** rIPhdUnitEnvParamTable - 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 для MES3124**

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



**1. Для MES5000 мониторинг температурных датчиков проводится в OID**

**1.3.6.1.4.1.89.83.2.1.1.1.4.68420481 – 1 датчик**  
**1.3.6.1.4.1.89.83.2.1.1.1.4.68420482 – 2 датчик**  
**1.3.6.1.4.1.89.83.2.1.1.1.4.68420483 – 3 датчик**  
**1.3.6.1.4.1.89.83.2.1.1.1.4.68420484 – 4 датчик**

**Статусы температурных датчиков для MES5248:**

**1.3.6.1.4.1.89.83.2.1.1.1.5.68420481 — 1 датчик**  
**1.3.6.1.4.1.89.83.2.1.1.1.5.68420482 — 2 датчик**  
**1.3.6.1.4.1.89.83.2.1.1.1.5.68420483 — 3 датчик**  
**1.3.6.1.4.1.89.83.2.1.1.1.5.68420484 — 4 датчик**

**Возможные значения:**

**enum1=1,ok**  
**enum2=2,unavailable**  
**enum3=3,nonoperational**

**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

**MIB:** rphysdescription.mib

**Используемые таблицы:** rIPhdUnitEnvParamTable - 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
```



#### 1) Состояния температурных датчиков

ok (1)

unavailable (2)

nonoperational (3)

#### 2) Просмотр состояния температурных датчик для MES5000:

1.3.6.1.4.1.89.83.2.1.1.1.5.68420481

1.3.6.1.4.1.89.83.2.1.1.1.5.68420482

1.3.6.1.4.1.89.83.2.1.1.1.5.68420483

1.3.6.1.4.1.89.83.2.1.1.1.5.68420484

## 4.3 Управление устройствам

### 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 "mes2124"

Команда CLI:

```
hostname mes2124
```

Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.2.1.1.5.0 s "mes2124"
```

### Включение/отключение management acl

**MIB:** mnginf.mib

**Используемые таблицы:** rIMngInfEnable — 1.3.6.1.4.1.89.89.2, rIMngInfActiveListName - 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)} \
1.3.6.1.4.1.89.89.2.0 i 1 \
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
```

### 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)**

#### 4.4 IP Service Level Agreements (IP SLA)

##### Создание операции ICMP Echo операции

**MIB:** ELTEX-IPSLA-MIB

**Используемые таблицы:** eltexIpSlaAdminCtrlTable — 1.3.6.1.4.1.35265.32.1.2.1,  
 eltexIpSlaAdminIcmpEchoTable - 1.3.6.1.4.1.35265.32.1.2.2

```

snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.32.1.2.1.1.1.{index of operation} i {index of operation}
\
1.3.6.1.4.1.35265.32.1.2.1.1.2.{index of operation} i {ICMP Echo(1)} \
1.3.6.1.4.1.35265.32.1.2.1.1.7.{index of operation} i {createandGo(4),
delete(6)} \
1.3.6.1.4.1.35265.32.1.2.2.1.1.{index of operation} i {active(1)} \
1.3.6.1.4.1.35265.32.1.2.2.1.2.{index of operation} a {source-IP} \
1.3.6.1.4.1.35265.32.1.2.2.1.3.{index of operation} a {destination-IP} \
1.3.6.1.4.1.35265.32.1.2.2.1.8.{index of operation} i {createandGo(4),
destroy(6)}
  
```

##### **Пример**

**Команда CLI:**

```

ip sla operation 1
icmp-echo 192.168.1.30 source-address 192.168.1.3
  
```

**Команда SNMP:**

```

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.2.1.1.1 i 1 \
1.3.6.1.4.1.35265.32.1.2.1.1.2.1 i 1 \
1.3.6.1.4.1.35265.32.1.2.1.1.7.1 i 4 \
1.3.6.1.4.1.35265.32.1.2.2.1.1.1 i 1 \
1.3.6.1.4.1.35265.32.1.2.2.1.2.1 a 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.2.2.1.3.1 a 192.168.1.3 \
1.3.6.1.4.1.35265.32.1.2.2.1.8.1 i 4
  
```

##### Редактирование параметров операции ICMP Echo

**MIB:** ELTEX-IPSLA-MIB

**Используемые таблицы:** eltexIpSlaAdminCtrlTable — 1.3.6.1.4.1.35265.32.1.2.1,  
 eltexIpSlaAdminIcmpEchoTable - 1.3.6.1.4.1.35265.32.1.2.2

```

snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.32.1.2.1.1.1.{index of operation} i {index of operation}
\
1.3.6.1.4.1.35265.32.1.2.2.1.2.{index of operation} a {source-IP} \
1.3.6.1.4.1.35265.32.1.2.2.1.3.{index of operation} a {destination-IP} \
1.3.6.1.4.1.35265.32.1.2.1.1.4.{index of operation} i {value frequency} \
1.3.6.1.4.1.35265.32.1.2.2.1.4.{index of operation} i {IfIndex} \
1.3.6.1.4.1.35265.32.1.2.2.1.5.{index of operation} i {value timeout} \
  
```

```
1.3.6.1.4.1.35265.32.1.2.2.1.6.{index of operation} i {payload} \
1.3.6.1.4.1.35265.32.1.2.2.1.7.{index of operation} i {value byte ToS} \
1.3.6.1.4.1.35265.32.1.2.2.1.8.{index of operation} i
{createandGo(4),destroy(6)}
```

### Пример

#### Команда CLI:

```
ip sla operation 1
 ip sla operation 1
 icmp-echo 192.168.1.3 source-address 192.168.1.30 source-interface
 GigabitEthernet1/0/1
 timeout 500
 request-data-size 20
 tos 3
 frequency 50
```

#### Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.2.1.1.1.1 i 1 \
1.3.6.1.4.1.35265.32.1.2.2.1.2.1 a 192.168.1.3 \
1.3.6.1.4.1.35265.32.1.2.2.1.3.1 a 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.2.1.1.4.1 i 50 \
1.3.6.1.4.1.35265.32.1.2.2.1.4.1 i 49 \
1.3.6.1.4.1.35265.32.1.2.2.1.5.1 i 500 \
1.3.6.1.4.1.35265.32.1.2.2.1.6.1 i 20 \
1.3.6.1.4.1.35265.32.1.2.2.1.7.1 i 3 \
1.3.6.1.4.1.35265.32.1.2.2.1.8.1 i 4
```

### Просмотр статистики IP SLA операции ICMP Echo

**MIB:** eltlpSlaV2.mib

**Используемые таблицы:** eltexIpSlaStatsIcmpEchoTable - 1.3.6.1.4.1.35265.32.1.3.2

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.32.1.3.2
```

### **Пример: просмотра статистики сервера с IP-адресом 192.168.10.2**

#### Команда CLI:

```
show ip sla statistics 1
```

#### Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.3.2
```

### Создание операции UDP Jitter операции

**MIB:** ELTEX-IPSLA-MIB

**Используемые таблицы:** eltexIpSlaAdminCtrlTable — 1.3.6.1.4.1.35265.32.1.2.1,  
eltexIpSlaAdminIcmpEchoTable - 1.3.6.1.4.1.35265.32.1.2.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.32.1.2.1.1.1.{index of operation} i {index of operation} \
1.3.6.1.4.1.35265.32.1.2.1.1.2.{index of operation} i {UDP Jitter(2)} \
1.3.6.1.4.1.35265.32.1.2.1.1.7.{index of operation} i {createandGo(4),
delete(6)} \
1.3.6.1.4.1.35265.32.1.2.3.1.1.{index of operation} i {index of operation} \
1.3.6.1.4.1.35265.32.1.2.3.1.2.{index of operation} a {source-IP} \
```

```

1.3.6.1.4.1.35265.32.1.2.3.1.3.{index of operation} i {destination port
UDP} \
1.3.6.1.4.1.35265.32.1.2.3.1.4.{index of operation} a {destination-IP} \
1.3.6.1.4.1.35265.32.1.2.3.1.5.{index of operation} i {source port UDP} \
1.3.6.1.4.1.35265.32.1.2.3.1.6.{index of operation} i {ifindex} \
1.3.6.1.4.1.35265.32.1.2.3.1.7.{index of operation} i {interval} \
1.3.6.1.4.1.35265.32.1.2.3.1.8.{index of operation} i {num-packets} \
1.3.6.1.4.1.35265.32.1.2.3.1.8.{index of operation} i {createandGo(4),
destroy(6)}

```

### Пример создания операции UDP Jitter с индексом 2

#### Команда CLI:

```

ip sla operation 2
  udp-jitter 192.168.1.3 35000 source-address 192.168.1.30 source-port 35000
  num-packets 4 interval 200

```

#### Команда SNMP:

```

snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.2.1.1.1.2 i 2 \
1.3.6.1.4.1.35265.32.1.2.1.1.2.2 i 2 \
1.3.6.1.4.1.35265.32.1.2.1.1.7.2 i 4 \
1.3.6.1.4.1.35265.32.1.2.3.1.1.2 i 2 \
1.3.6.1.4.1.35265.32.1.2.3.1.2.2 a 192.168.1.3 \
1.3.6.1.4.1.35265.32.1.2.3.1.3.2 i 35000 \
1.3.6.1.4.1.35265.32.1.2.3.1.4.2 a 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.2.3.1.5.2 i 35000 \
1.3.6.1.4.1.35265.32.1.2.3.1.6.2 i 49 \
1.3.6.1.4.1.35265.32.1.2.3.1.7.2 i 200 \
1.3.6.1.4.1.35265.32.1.2.3.1.8.2 i 4 \
1.3.6.1.4.1.35265.32.1.2.3.1.12.2 i 4

```

### Редактирование параметров операции UDP Jitter

**MIB:** ELTEX-IPSLA-MIB

**Используемые таблицы:** eltexIpSlaAdminCtrlTable — 1.3.6.1.4.1.35265.32.1.2.1,  
 eltexIpSlaAdminUdpJitterTable - 1.3.6.1.4.1.35265.32.1.2.3

```

snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.32.1.2.1.1.4.{index of operation} i {seconds between
operations} \
1.3.6.1.4.1.35265.32.1.2.1.1.7.{index of operation} i 1 \
1.3.6.1.4.1.35265.32.1.2.3.1.1.{index of operation} i {index of operation}
\
1.3.6.1.4.1.35265.32.1.2.3.1.9.{index of operation} i {value timeout} \
1.3.6.1.4.1.35265.32.1.2.3.1.10.{index of operation} i {size in bytes} \
1.3.6.1.4.1.35265.32.1.2.3.1.11.{index of operation} i {value byte ToS} \
1.3.6.1.4.1.35265.32.1.2.3.1.12.{index of operation} i 1

```

### Пример редактирования операции UDP Jitter с индексом 2

#### Команда CLI:

```

ip sla operation 2
  udp-jitter 192.168.1.3 35000 source-address 192.168.1.30 source-port 35000
  num-packets 4 interval 200
  timeout 22
  request-data-size 22
  tos 3
  frequency 10
exit

```

#### Команда SNMP:

```

snmpset -v2c -c private 192.168.1.30 \

```

```
1.3.6.1.4.1.35265.32.1.2.1.1.7.2 i 1 \
1.3.6.1.4.1.35265.32.1.2.1.1.4.2 i 10 \
1.3.6.1.4.1.35265.32.1.2.3.1.1.2 i 2 \
1.3.6.1.4.1.35265.32.1.2.3.1.9.2 i 22 \
1.3.6.1.4.1.35265.32.1.2.3.1.10.2 i 22 \
1.3.6.1.4.1.35265.32.1.2.3.1.11.2 i 3 \
1.3.6.1.4.1.35265.32.1.2.3.1.12.2 i 1
```

### Просмотр статистики IP SLA операции UDP Jitter

**MIB:** eltlpSlaV2.mib

**Используемые таблицы:** eltexIpSlaStatsUdpJitterTable - 1.3.6.1.4.1.35265.32.1.3.3

```
snmpwalk -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.32.1.3.3
```

#### **Пример**

**Команда CLI:**

```
show ip sla statistics 1
```

**Команда SNMP:**

```
snmpwalk -v2c -c public 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.3.3
```

### Удаление операции ICMP Echo, UDP Jitter

**MIB:** eltlpSlaV2.mib

**Используемые таблицы:** eltexIpSlaAdminIcmpEchoTable – 1.3.6.1.4.1.35265.32.1.2.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.32.1.2.{ICMP-Echo(2), UDP Jitter(3)}.1.{ICMP-Echo(8), UDP
Jitter(12)}.1 i {destroy(6)} \
1.3.6.1.4.1.35265.32.1.2.1.1.7.1 i {destroy(6)}
```

#### **Пример удаления операции ICMP Echo с индексом 1**

**Команда CLI:**

```
no ip sla statistics 1
```

**Команда SNMP:**

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.32.1.2.2.1.8.1 i 6 \
1.3.6.1.4.1.35265.32.1.2.1.1.7.1 i 6
```

## 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 SNMP-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
```

## 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
```

#### 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 1/0/2

Команда CLI:

```
interface GigabitEthernet 1/0/2
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
```



### 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: swlfduplexAdminMode - 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 GigabitEthernet 1/0/1

```
Команда CLI:  
interface GigabitEthernet1/0/1  
duplex half  
  
Команда SNMP:  
snmpwalk -v2c -c public 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

#### 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 GigabitEthernet 1/0/1

#### Команда CLI:

```
show interfaces status GigabitEthernet 1/0/1
```

#### Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
  1.3.6.1.2.1.2.2.1.8.49
```



#### 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 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 — 1.3.6.1.2.1.31.1.1.3

```
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

```
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
```



### 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)

### Очистка счетчиков интерфейсов

**MIB:** rllInterfaces.mib

**Используемые таблицы:** rllfClearPortMibCounters - 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 2124M**

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  
000000000000FFFFFFFF0000000000000000000000000000000000000000000000000000000000000000000  
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000  
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000  
00000000000000000000000001FFFE
```



**1) Для MES1000:**

```
snmpset -v2c -c private 192.168.1.30 1.3.6.1.4.1.89.54.4.0 x  
fffff00000f000000000000000000000000000000000000000000000000000000000000000000000000  
00000000000000000000000000000000000000000000000000000000000000000000000000000000000
```





**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
```



Выведенные значения представляют из себя битовую маску, методика расчета которой приведена в [Приложении А](#).

#### 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
```

#### 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> \
1.3.6.1.4.1.89.48.45.1.3.168.249.75.51.41.192.32 i {mac-group number} \
1.3.6.1.4.1.89.48.45.1.4.168.249.75.51.41.192.32 i {createAndGo(4),
destroy(6)}
```

#### **Пример**

##### **Команда CLI:**

```
vlan database
vlan 10,20
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:** rErrdisableRecoveryEnable - 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 отключения порта

**MIB:** rErrdisableRecoveryIfReason

**Используемые таблицы:** rErrdisableRecoveryIfReason - 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:

- oopback-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:** rErrdisableRecoveryEnable - 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 a 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:** rErrdisableRecoveryInterval - 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**

**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: включение таблицы для использованного телефона**

**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

**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
```

## **6.5 Настройка LLDP**

#### Глобальное включение/отключение lldp

**MIB:** rLldp.mib

**Используемые таблицы:** rLldpEnabled - 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:** rslpAddrEntry - 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
```

### Удаление IP-адреса на interface vlan:

**MIB:** rlip.mib

**Используемые таблицы:** rslpAddrEntry - 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 во 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:** rIDhcpClActionStatus - 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
```



## 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
```

### Просмотр параметра 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
```

## 9 НАСТРОЙКА КОЛЬЦЕВЫХ ПРОТОКОЛОВ

### 9.1 Настройка протокола 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:**

- Forwarding (1)**
- Blocking (2)**
- Signal-fail (3)**
- Manual-switch (4)**
- 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:

- Forwarding (1)
- Blocking (2)
- Signal-fail (3)
- Manual-switch (4)
- 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 порта:

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

Pending (6)

## 9.2 Настройка протокола Spanning-tree

### Включение/выключение 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 выключения протокола 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 GigabitEthernet1/0/2  
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
```

### 9.3 Протокол EAPS

#### Просмотр состояния кольца

**MIB:** RADLAN-BRIDGEMIBOBJECTS-MIB.mib

**Используемые таблицы:** REapsRingState - 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:** rIBrgStaticInetMulticastEntry - 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 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 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 \
1.3.6.1.4.1.89.116.5.1.6.622.1.4.239.200.200.17.1.4.0.0.0.0 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
00000000000008000 \
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 приложении А.

#### Запрещение прохождения незарегистрированного Multicast трафика

**MIB:** rlbrgmulticast.mib

**Используемые таблицы:** rIMacMulticastUnregFilterEnable - 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 "00000000000000000018"
```



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

2) Методику расчета битовой маски можно посмотреть в приложении А.

### Фильтрация многоадресного трафика

**MIB:** rlbrgmulticast.mib

**Используемые таблицы:** rIMacMulticastEnable - 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:** rIIgmpSnoopEnable - 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:** rIIgmpMldSnoopVlanEnable - 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:** rIIgmpMldSnoopMembershipTable - 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.{ifindex} 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.

## 10.2 Функции ограничения multicast-трафика

### Создание multicast snooping profile

**MIB:** eltlpMulticast.mib

**Используемые таблицы:** eltlgmpMldSnoopProfileTable - 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:** eltlpMulticast.mib

**Используемые таблицы:** eltlgmpMldSnoopFilterTable - 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:** eltlpMulticast.mib

**Используемые таблицы:** eltlgmpMldSnoopIfProfileExtEntry - 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 порту

**MIB:** eltlpMulticast.mib

**Используемые таблицы:** eltlgmpMldSnooIfMaxGroupsEntry - 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
```

## 11 ФУНКЦИИ УПРАВЛЕНИЯ

### 11.1 Механизм AAA

#### Добавление нового пользователя

**MIB:** rlaaa.mib

**Используемые таблицы:** rIAAALocalUserTable - 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 {privilege 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 зашифрованном виде, пишется обязательно в кавычках, перед паролем добавляется #.**

#### Настройка методов авторизации для enable-пользователя

**MIB:** RADLAN-AAA

**Используемые таблицы:** rIAAAMethodListEntry - 1.3.6.1.4.1.89.79.15.1

```
snmpset -v2c -c <community> <ip address>
  1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.99.95.100.{"enable_c_
default" in DEC, каждая буква логина отделяется от следующей точкой} i
  {enable(2),radius(4),tacacs(5)} \
  1.3.6.1.4.1.89.79.15.1.3.16.101.110.97.98.108.101.95.99.95.100.{"enable_c_
default" in DEC, каждая буква логина отделяется от следующей точкой} i
  {enable(2),radius(4),tacacs(5)} \
  1.3.6.1.4.1.89.79.15.1.4.16.101.110.97.98.108.101.95.99.95.100.{"enable_c_
default" in DEC, каждая буква логина отделяется от следующей точкой} i
  {enable(2),radius(4),tacacs(5)}
```

## Пример

### Команда CLI:

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

### Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.99.95.100.101.102.97.1
17.108.116 i 5 \
1.3.6.1.4.1.89.79.15.1.3.16.101.110.97.98.108.101.95.99.95.100.101.102.97.1
17.108.116 i 4 \
1.3.6.1.4.1.89.79.15.1.4.16.101.110.97.98.108.101.95.99.95.100.101.102.97.1
17.108.116 i 2
```

## Удаление настройки методов авторизации для enable-пользователя

**MIB:** RADLAN-AAA

**Используемые таблицы:** rIAAAMethodListEntry – 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.{method1(2)}.16.101.110.97.98.108.101.95.99.95.100.{
"enable_c_default" in DEC, каждая буква логина отделяется от следующей
точкой} i {enable(0),radius(6),tacacs(2)} \
1.3.6.1.4.1.89.79.15.1.{method2(3)}.16.101.110.97.98.108.101.95.99.95.100.{
"enable_c_default" in DEC, каждая буква логина отделяется от следующей
точкой} i {enable(0),radius(6),tacacs(2)} \
1.3.6.1.4.1.89.79.15.1.{method3(4)}.16.101.110.97.98.108.101.95.99.95.100.{
"enable_c_default" in DEC, каждая буква логина отделяется от следующей
точкой} i {enable(0),radius(6),tacacs(2)}
```

## Пример удаления методов авторизации для enable пользователя

### Команда CLI:

```
no aaa authentication enable default
```

### Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.79.15.1.2.16.101.110.97.98.108.101.95.99.95.100.101.102.97.1
17.108.116 i 2 \
1.3.6.1.4.1.89.79.15.1.3.16.101.110.97.98.108.101.95.99.95.100.101.102.97.1
17.108.116 i 6 \
1.3.6.1.4.1.89.79.15.1.4.16.101.110.97.98.108.101.95.99.95.100.101.102.97.1
17.108.116 i 0
```

## Включение telnet сервера

**MIB:** RADLAN-TELNET-MIB

**Используемые таблицы:** rITelnetEnable - 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:** RADLAN-SSH-MIB

**Используемые таблицы:** rISshServerEnable - 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:** rIAAA.mib

**Используемые таблицы:** rIAAAUserInetName - 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**

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
```



## 12 ЗЕРКАЛИРОВАНИЕ 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 лимитов для зеркалирования на интерфейсе

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

**Используемые таблицы:** eltQosTailDropMirrorTxDescriptorsLimit - 1.3.6.1.4.1.35265.1.23.12.1.1.6  
 eltQosTailDropMirrorRxDescriptorsLimit - 1.3.6.1.4.1.35265.1.23.12.1.1.5

```
snmpset -v2c -c <community> <IP address> \
  1.3.6.1.4.1.35265.1.23.12.1.1
```

### **Пример**

**Команда CLI:**  

```
interface gigabitethernet1/0/1
  qos tail-drop mirror-limit rx 5902
  qos tail-drop mirror-limit tx 5902
```

**Команда SNMP:**  

```
snmpset -v2c -c private 192.168.1.30 \
  1.3.6.1.4.1.35265.1.23.12.1.1.5.0 i 5902
```

```
snmpset -v2c -c private 192.168.1.30 \
  1.3.6.1.4.1.35265.1.23.12.1.1.6.0 i 5902
```

### Настройка зеркалирования 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
```



**Для серий MES1000/MES2000 функция зеркалирования Vlan не поддерживается.**

## 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 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 GigabitEthernet1/0/12
```

**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}.4
```

#### **Пример измерения длины кабеля на интерфейсе GigabitEthernet1/0/5**

Команда CLI:

```
show cable-diagnostics cable-length interface GigabitEthernet1/0/5
```

Команда SNMP:

```
snmpwalk -v2c -c public 192.168.1.30 \
  1.3.6.1.4.1.89.90.1.2.1.3.53.4
```

## **13.2 Диагностика оптического трансивера**

### Снятие показаний DDM

**MIB:** rlphy.mib

**Используемые таблицы:** rlPhyTestGetResult - 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
```



#### **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
```

## 14 ФУНКЦИИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

### 14.1 Функции обеспечения защиты портов

#### Ограничение количества MAC-адресов, изучаемых на Ethernet-портах

**MIB:** rlinterfaces.mib

**Используемые таблицы:** swlfTable - 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:** swlfPortLockIfRangeTable - 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 "00000000000000C0"
```



Методика расчета битовой маски приведена в [Приложении А](#).

#### Установка режима работы port security

**MIB:** rlinterfaces.mib

**Используемые таблицы: swlftable - 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: swlfllockAdminStatus- 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: swlflAdminLockAction- 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: swlfllockMaxMacAddresses- 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 (по умолчанию 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:** dot1qTrFdbTable - 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

**Используемые таблицы:** ipNetToMediaPhysAddress — 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
```

```
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:

```
snmpwalkset -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.4.22.1.2
snmpwalkset -v2c -c public 192.168.1.30 \
1.3.6.1.2.1.17.7.1.2.2.1
```



1. Значение таблицы ipNetToMediaPhysAddress отображает IP-адрес и MAC-адрес vlan.
2. Значение таблицы dot1qTpFdbEntry - отображает статус и идентификационный номер порта, с которого доступно устройство.

### Перевод порта в режим изоляции и внутри группы портов

MIB: rIprotectedport.mib

**Используемые таблицы:** rIProtectedPortsTable - 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
```

## **14.2 Контроль протокола DHCP и опция 82**

### Включение/выключение DHCP сервера на коммутаторе

**MIB:** rldhcp.mib

**Используемые таблицы:** rIDhcpRelayInterfaceListTable - 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
```

### Включение/выключение dhcp snooping глобально

**MIB:** rlbridge-security.mib

**Используемые таблицы:** rIIPDhcpSnoopEnable - 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
```

**Просмотр записей таблицы dhcp snooping****MIB:** rIBridgeSecurity.mib**Используемые таблицы:** rIIPDhcpSnoopEntry - 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 private 192.168.1.30 \  
1.3.6.1.4.1.89.112.1.11.1
```

**Настройка DHCP relay в vlan****MIB:** rldhcp.mib**Используемые таблицы:** rIDhcpRelayInterfaceListVlanId1To1024 - 1.3.6.1.4.1.89.38.29.1.3  
rIDhcpRelayInterfaceListVlanId1025To2048 - 1.3.6.1.4.1.89.38.29.1.4  
rIDhcpRelayInterfaceListVlanId2049To3072 - 1.3.6.1.4.1.89.38.29.1.5  
rIDhcpRelayInterfaceListVlanId3073To4094 - 1.3.6.1.4.1.89.38.29.1.6

```
snmpwalk -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 приложении А.**

### Настройка ip DHCP information option

**MIB:** rlbriDgesecurity.mib

**Используемые таблицы:** rllpDhcpOpt82InsertionEnable - 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 private 192.168.1.30 \  
1.3.6.1.4.1.89.112.1.8.0 i 1

### Включение/выключение dhcp snooping во vlan

**MIB:** rlbridge-security.mib

**Используемые таблицы:** rllpDhcpSnoopEnableVlanTable - 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:** rllpDhcpSnoopTrustedPortTable - 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:** rbridge-security.mib

**Используемые таблицы:** rllpSourceGuardEnable - 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:** rbridge-security.mib

**Используемые таблицы:** rllpDhcpSnoopStaticTable - 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:** rbridge-security.mib

**Используемые таблицы:** rllpSourceGuardPortTable - 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:** rllpArpInspectEnable - 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:** rllpArpInspectEnableVlanTable - 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:** rllpArpInspectTrustedPortRowStatus - 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:** rbridge-security.mib

**Используемые таблицы:** rllpArpInspectAssignedListName - 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
```

## **14.5 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 1
```

#### **Пример**

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:** rldot1xExtAuthSessionStatsTable - 1.0.8802.1.1.1.2.1.1.13

```
snmpset -v2c -c <community> <IP address> \  
1.0.8802.1.1.1.2.1.1.13.{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
```

#### Настройка режимов аутентификации 802.1X на интерфейсе.

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

**Используемые таблицы:** dot1xAuthConfigTable - 1.0.8802.1.1.1.2.1.1.6

```
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
```

#### Установка периода между повторными проверками подлинности

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

**Используемые таблицы:** dot1xAuthConfigTable - 1.0.8802.1.1.1.2.1.1.12

```
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
```

#### Разрешение наличия одного/нескольких клиентов на авторизованном порту 802.1X

**MIB:** rllnterfaces.mib

**Используемые таблицы:** swlftable - 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 обработки опции Tunnel-Private-Group-ID (81) в сообщениях RADIUS-сервера (Назначение vlan через radius server)

**MIB:** radlan-dot1x-mib.mib

**Используемые таблицы:** rldot1xAuthenticationPortTable - 1.3.6.1.4.1.89.95.10.1.2

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

### Пример назначения vlan через radius server на интерфейсе GigabitEthernet 1/0/3

#### Команда CLI:

```
interface gigabitEthernet 1/0/3  
dot1x radius-attributes vlan
```

#### Команда SNMP:

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

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

**MIB:** rlaaa.mib

**Используемые таблицы:** rIAAAEapMethodListTable - 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:** rIRadiusServerInetTable - 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
```

### 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> \
1.3.6.1.4.1.89.95.10.1.1.1.{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 mac-authentication mac-only
```

#### 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 неавторизованным пользователям интерфейса доступ в гостевой VLAN

**MIB:** radlan-dot1x-mib.mib

**Используемые таблицы:** rldot1xExtAuthSessionStatsTable — 1.3.6.1.4.1.89.95.1

```
snmpset -v2c -c <community> <IP address> \  
1.3.6.1.4.1.89.95.5.0 x {ifindex в виде битовой маски}
```

### **Пример разрешения неавторизованным пользователям интерфейса GigabitEthernet 1/0/2 доступ в гостевой VLAN**

**Команда CLI:**

```
interface gigabitethernet1/0/2  
dot1x guest-vlan enable
```

**Команда SNMP:**

```
snmpset -v2c -c private 192.168.1.30 \  
1.3.6.1.4.1.89.95.5.0 x 00000000000040
```

## **14.6 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**

**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)}
```

### Включение/отключение loopback-detection на интерфейсах

**MIB:** rllbd.mib

**Используемые таблицы:** rllbdIfAdminStatus - 1.3.6.1.4.1.89.127.4

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.127.4.0 x {Битовая маска для интерфейса}
```

### **Пример включения loopback-detection на интерфейсе GigabitEthernet1/0/2**

**Команда CLI:**

```
interface gigabitethernet1/0/2
loopback-detection enable
```

**Команда SNMP:**

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.127.4.0 x 00000000000004000
```



**Методику расчета битовой маски можно посмотреть в приложении А в конце документа**

### 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.

## 14.7 Контроль широковежательного шторма (storm-control)

### Включение Storm-control для broadcast трафика

**MIB:** radlan-stormctrl-mib.mib

**Используемые таблицы:** rlStormCtrlBroadcastEnable - 1.3.6.1.4.1.89.77.8.1.6

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

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

Команда CLI:  

```
interface GigabitEthernet1/0/1
storm-control broadcast enable
```

Команда SNMP:  

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

### Настройка storm-control для unknown-unicast трафика (до версии 2.5.46/1.1.46 включительно)

**MIB:** RADLAN-STORMCTRL-MIB

**Используемые таблицы:** rlStormCtrlMulticastEnable – 1.3.6.1.4.1.89.77.8.1.8  
rlStormCtrlUnknownUnicastEnable - 1.3.6.1.4.1.89.77.8.1.2

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

### Пример установки unknown-unicast трафика на интерфейсе GigabitEthernet1/0/1

Команда CLI:  

```
interface GigabitEthernet1/0/1
storm-control include-unknown-unicast
```

Команда SNMP:  

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

### Настройка перевода порта в errdisable состояние при срабатывании storm-control

**MIB:** ELTEX-MES-STORMCTRL-MIB

### Используемые таблицы: eltStormCtrlConfigTable - 1.3.6.1.4.1.35265.1.23.77.1.1.2

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.35265.1.23.77.1.1.2.1.2.{ifindex}.1 i {createandGo(2),
delete(1)}
```

#### Пример

##### Команда CLI:

```
interface GigabitEthernet1/0/1
storm-control broadcast shutdown
```

##### Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.35265.1.23.77.1.1.2.1.2.49.1 i 2
```



**Работает только для broadcast трафика**

### Настройка уровня ограничения трафика (начиная с версий 2.5.47/1.1.47)

**MIB:** radlan-stormctrl-mib.mib

### Используемые таблицы: rlStormCtrlBroadcastRate - 1.3.6.1.4.1.89.77.8.1.7

```
snmpset -v2c -c <community> <IP address> \
1.3.6.1.4.1.89.77.8.1.{unknown-unicast(3), broadcast(7),
multicast(9)}.{ifIndex} u {kbps}
```

#### Пример установки уровня kbps для broadcast трафика на интерфейсе GigabitEthernet1/0/1

##### Команда CLI:

```
interface GigabitEthernet1/0/1
storm-control broadcast level kbps 1500
```

##### Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
1.3.6.1.4.1.89.77.8.1.7.49 u 1500
```



**До версии 2.5.46/1.1.46 включительно лимиты для unknown-unicast, broadcast, multicast трафика были общими**

### Настройка Storm-control для multicast трафика на интерфейсе (начиная с версий 2.5.47/1.1.47)

**MIB:** radlan-stormctrl-mib.mib

### Используемые таблицы: rlStormCtrlMulticastEnable - 1.3.6.1.4.1.89.77.8.1.8

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

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

##### Команда CLI:

```
interface GigabitEthernet1/0/1
storm-control multicast enable
```

##### Команда SNMP:

```
snmpset -v2c -c private 192.168.1.30 \
```

```
1.3.6.1.4.1.89.77.8.1.8.49 i 1
```

Настройка Storm-control unknown-unicast на интерфейсе (начиная с версии 2.5.47/1.1.47)

**MIB:** radlan-stormctrl-mib.mib

**Используемые таблицы:** rlStormCtrlUnknownUnicastEnable - 1.3.6.1.4.1.89.77.8.1.2

```
snmpset -v2c -c <community> <IP address> \  
1.3.6.1.4.1.89.77.8.1.2.{ifIndex} i 1
```

**Пример включения storm-control для multicast на интерфейсе GigabitEthernet1/0/1**

Команда CLI:

```
interface GigabitEthernet 1/0/1  
storm-control unknown-unicast enable
```

Команда SNMP:

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



## 15 КОНФИГУРИРОВАНИЕ IP И MAC ACL (СПИСКИ КОНТРОЛЯ ДОСТУПА)

### Создание ip access-list (ACL)

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosAcITable - 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 [Приложении Б](#).

### Привязка IP или MAC ACL к порту

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosIfAcIIn - 1.3.6.1.4.1.89.88.13.1.14, rIQosIfPolicyMapStatus – 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**

**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
```



Для удаления ACL с порта достаточно индекс 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
1.3.6.1.4.1.89.88.13.1.13.50.2 i 1
```

### Привязка IP и MAC ACL к порту

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosIfAcIIn - 1.3.6.1.4.1.89.88.13.1.14, rIQosIfIpv6AcIIn - 1.3.6.1.4.1.89.88.13.1.201.3.6.1.4.1.89.88.13.1.20, rIQosIfPolicyMapStatus - 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**

**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 -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:** rIQosClassMapTable - 1.3.6.1.4.1.89.88.9, rIQosPolicyMapTable - 1.3.6.1.4.1.89.88.11, rIQosPolicyClassPriorityRefTable - 1.3.6.1.4.1.89.88.39, rIQosPolicerTable - 1.3.6.1.4.1.89.88.10

**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 {create and
go(4),destroy(6)}
```

### 3. Привязываем class-мар к policy-мар

```
snmpset -v2c -c <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}
```

### 4. Создаем ограничение скорости для class-мар

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

### 5. Привязываем ограничение скорости к class-мар

```
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-мар с именем test и выставляется метка DSCP=36(DEC) трафика, подпавшего под IP ACL. Class test привязывается к policy-мар 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 cos 4
   set queue 3
   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 \
1.3.6.1.4.1.35265.1.23.88.5.1.1.20.5 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

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 3 \
1.3.6.1.4.1.35265.1.23.88.5.1.3.20.4 i 4

```

### Назначение Policy-map на порт

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosIfPolicyMapPointerIn - 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 (Ifindex 51)**

**Команда CLI:**

```

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

```

**Команда SNMP:**

```

snmpset -c private -v2c 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/mas acl с пометкой log-input

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosDenyAceStatisticsTable - 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
```

### Создание mac access-list (ACL)

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosAcITable - 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:

```
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 правила в MAC ACL на основе EtherType

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosTuplETable - 1.3.6.1.4.1.89.88.5, rIQosAceTidxTable - 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-мас (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-портах

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosIfPolicyEntry - 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:** rIQosIfPolicyEntry - 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}.2 i {disable(1), enable(2)} \
1.3.6.1.4.1.89.88.13.1.18.{ifIndex}.2 i {rate-limit}
\1.3.6.1.4.1.89.88.13.1.19.{ifIndex}.2 i {Burst}
```

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

**Команда CLI:**  
interface GigabitEthernet 1/0/1  
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

#### 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
```



**1) Чтобы вернуться к настройкам по умолчанию достаточно установить значение равным 12 для коммутаторов серии MES2000.**

**Для коммутаторов серии MES3000 и MES5000 значение по умолчанию равно 18.**

**2) Для коммутаторов серий MES3000, MES5000 номер очереди изменяется от 1 до 8.**

**3) Для коммутаторов серий MES1000, MES2000 номер очереди изменяется от 1 до 4.**

### 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
```



**Чтобы вернуться к настройкам по умолчанию достаточно установить значение равным 44**

```
snmpset -v2c -c private 192.168.1.30 \  
1.3.6.1.4.1.35265.1.23.12.1.1.4.1.2 i 44
```

### 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 назначения созданного профиля на интерфейс GigabitEthernet 1/0/1

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:** ELTEX-MES-QOS-TAIL-DROP-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
```

## 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**

Команда CLI:

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

Количество tail drop пакетов в 8 очереди интерфейса GigabitEthernet 1/0/1:

```
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:

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**

Команда CLI:

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

Количество переданных пакетов в 8 очереди интерфейса GigabitEthernet 1/0/1:

```
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:

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 МАРШРУТИЗАЦИЯ

### 17.1 Статическая маршрутизация

#### 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:** rIip.mib

**Используемые таблицы:** rIipStaticRouteTable - 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 для MES3124**

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

### 17.2 Динамическая маршрутизация

#### Просмотр соседства OSPF

**MIB:** rIip.mib

**Используемые таблицы:** rIOspfNbrTable - 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 для MES3124**

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
```



**На коммутаторах серий MES1000/MES2000 OSPF аппаратно не поддерживается.**



## ПРИЛОЖЕНИЕ Б. ПРИМЕР СОЗДАНИЯ ТИПОВОГО IP ACL

В данном приложении рассмотрен пример заполнения 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:** rIQoSTupleTable - 1.3.6.1.4.1.89.88.5, rIQoSAceTidxTable - 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:** rIQoSTupleTable - 1.3.6.1.4.1.89.88.5, rIQoSAceTidxTable - 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:** rIQoSTupleTable - 1.3.6.1.4.1.89.88.5, rIQoSAceTidxTable - 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:** rIQoSTupleTable - 1.3.6.1.4.1.89.88.5, rIQoSAceTidxTable - 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:** rIQosTupleTable - 1.3.6.1.4.1.89.88.5, rIQosAceTidxTable - 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:** rIQoSTupleTable - 1.3.6.1.4.1.89.88.5, rIQoSAceTidxTable - 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:** rIQoSTupleTable - 1.3.6.1.4.1.89.88.5, rIQoSAceTidxTable - 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
"0x0019161514160000000000000000"
```

```
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:** rIQosTupleTable - 1.3.6.1.4.1.89.88.5, rIQosAceTidxTable - 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
"0x01000c0000000000000000ffffff"

```

```

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 I2 12 00 88 I2 13 00 63 I2 15 00 07
deny any any offset-list PADO
```

### Создание mac access-list

**MIB:** qosclimib.mib

**Используемые таблицы:** rIQosAcITable – 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:** rIQosOffsetTable - 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 I2 12 00 88 I2 13 00 63 I2 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 12 12 00 88 12 13 00 63 12 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:** rIQosTupleTable - 1.3.6.1.4.1.89.88.5, rIQosAceTidxTable - 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
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
```

## ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Для получения технической консультации по вопросам эксплуатации оборудования ТОО «ЭлтексАлатау» Вы можете обратиться в Сервисный центр компании:

050032, Республика Казахстан, г. Алматы, мкр-н. Алатау, ул. Ибрагимова 9

Телефон:

+7(727) 220-76-10, +7 (727) 220-76-07

E-mail: [post@eltexalatau.kz](mailto:post@eltexalatau.kz)

На официальном сайте компании Вы можете найти техническую документацию и программное обеспечение для продукции ТОО «ЭлтексАлатау», обратиться к базе знаний, проконсультироваться у инженеров Сервисного центра на техническом форуме.

Официальный сайт компании: <http://eltexalatau.kz>