Программное обеспечение для Ethernet-коммутаторов серии STK

Руководство администратора

Приведенные в настоящем руководстве администратора названия и программные продукты могут содержать в себе зарегистрированные товарные знаки своих владельцев.

Настоящее руководство администратора содержит всю необходимую информацию для корректного использования программного обеспечения для Ethernet-коммутаторов серии STK (далее по тексту – ПО) производства компании ООО «СТАНДАРТ ТЕЛЕКОМ ИТ».

Настоящее руководство предназначено для специалистов, отвечающих за администрирование системы.

Настоящее руководство предназначено для пользователей, осуществляющих мониторинг и настройку коммутаторов марки STK через графический интерфейс.

Данное руководство администратора предназначено для ознакомления администратора с операциями консоли.

Оглавление

1 Введение

1.1 Назначение руководства администратора

Настоящее руководство администратора представляет полную информацию о характеристиках и функциях использования программного обеспечения для Ethernet-коммутаторов серии STK.

Программное обеспечение предназначено для работы с коммутаторами марки STK и работает только при его наличии.

Администрация программного обеспечения осуществляется через терминал персонального компьютера.

1.2 Начало работы администратора с ПО

Перед началом работы убедитесь в наличии коммутатора марки STK совместимого с программным обеспечением для Ethernet-коммутаторов серии STK и персонального компьютера.

Ознакомьтесь с настоящим руководством администратора и при возникновении вопросов воспользуйтесь содержанием, собранным для удобства навигации по разделам руководства. Убедитесь, что проблема не описана в одном из них.

Если возникшая проблема не описана ни в одном из разделов руководства администратора, то обратитесь за помощью в отдел технической поддержки по электронной почте:

support@st-telecom-it.ru

1.3 Уровень подготовки пользователей

Специалист, осуществляющий работу с программным обеспечением для Ethernet-коммутаторов серии STK, должен обладать знаниями и умениями эксплуатации коммутаторов и их операционной системы, а также обладать углублёнными знаниями работы с терминалом.

2 Требование к вспомогательному оборудованию

2.1 Требования к коммутатору

Перед началом работы убедитесь, что у вас установлен коммутатор марки STK.

2.2 Требования к персональному компьютеру (ПК)

Для доступа к интерфейсу программного обеспечения, к персональному компьютеру предъявляются следующие требования:

2.2.1 Операционная система

- Windows 7 или выше;
- Linux с ядром 4.19 и выше.

2.2.2 Web-браузер

Наличие на персональном компьютере web-браузера.

Специальных требования к web-браузеру не предъявляется.

2.2.3 Технические характеристики ПК

Специальных требований к техническим характеристикам персонального компьютера не предъявляется.

3 Работа с аутентификацией, авторизацией и учетом

3.1 Команда aaa accounting

Команда **aaa accounting** используется для настройки учета **(accounting)** действий пользователей на устройстве. Учет позволяет отслеживать и регистрировать действия, выполняемые пользователями, такие как вход в систему, выполнение команд и т.д. Эта информация может быть отправлена на сервер TACACS+ для дальнейшего анализа и аудита.

aaa accounting — основная команда, которая активирует функцию учета.

{console | telnet | ssh } — указывает тип подключения, для которого будет применяться учет:

- **console** — учет для подключений через консольный порт.
- **telnet** — учет для подключений через Telnet.
- **ssh** — учет для подключений через SSH.
- **tacacs** — указывает, что учет будет отправляться на сервер TACACS+.
- **commands <priv_lvl>** — опция, которая включает учет выполнения команд. <priv_lvl> указывает уровень привилегий, для которого будет вестись учет. Например, уровень привилегий может быть 1 (обычный пользователь) или 15 (администратор).
- **exec** — опция, которая включает учет сессий (например, начало и завершение сессии пользователя).

3.1.1 Пример

```
aaa accounting ssh tacacs commands 15 exec
```

Учет будет вестись для **SSH-подключений**.

Учет будет отправляться на сервер **TACACS+**.

Будут регистрироваться все команды, выполненные с уровнем привилегий **15** (администратор).

Также будет регистрироваться начало и завершение сессии.

3.1.2 Telnet-подключение

Учет для **Telnet-подключений** с регистрацией команд уровня привилегий 1:

```
aaa accounting telnet tacacs commands 1
```

Учет будет вестись для Telnet-подключений.

Будут регистрироваться все команды, выполненные с уровнем привилегий 1.

3.1.3 Регистрация сессии

Учет для консольных подключений с регистрацией сессий:

```
aaa accounting console tacacs exec
```

Учет будет вестись для консольных подключений.

Будут регистрироваться начало и завершение сессий.

3.2 Команда aaa authentication login

Команда **aaa authentication login** используется для настройки аутентификации пользователей при входе в систему промышленного коммутатора.

Она определяет методы аутентификации (**локальные, через RADIUS или TACACS+**), которые будут использоваться для проверки учетных данных пользователей в зависимости от типа подключения **(console, telnet, ssh, http).**

```
aaa authentication login { console | telnet | ssh | http }
{ { local | radius | tacacs } [ { local | radius | tacacs }
[ { local | radius | tacacs } ] ] }
```

3.2.1 Тип подключения:

- **console** — аутентификация для подключений через консольный порт.
- **telnet** — аутентификация для подключений через **Telnet.**
- **ssh** — аутентификация для подключений через **SSH.**
- **http** — аутентификация для подключений через веб-интерфейс **(HTTP).**

3.2.2 Методы аутентификации:

- **local** — аутентификация через локальную базу пользователей на устройстве.
- **radius** — аутентификация через сервер **RADIUS**.
- **tacacs** — аутентификация через сервер **TACACS+**.

Методы аутентификации указываются в порядке приоритета. Устройство будет пытаться использовать первый метод, и если он недоступен, перейдет к следующему.

3.2.3 Примеры

Аутентификация для SSH-подключений с приоритетом **TACACS+**, затем **RADIUS**, и локальная база как резерв.

```
aaa authentication login ssh tacacs radius local
```

Устройство сначала попытается аутентифицировать пользователя через сервер **TACACS+.**

Если **TACACS+** недоступен, будет использован сервер **RADIUS**.

Если **RADIUS** также недоступен, аутентификация будет выполнена через локальную базу пользователей.

Аутентификация для консольных подключений только через локальную базу:

```
aaa authentication login console local
```

Аутентификация для Telnet-подключений с приоритетом RADIUS, затем локальная база:

```
aaa authentication login telnet radius local
```

Устройство сначала попытается аутентифицировать пользователя через сервер RADIUS.

Если RADIUS недоступен, будет использована локальная база пользователей.

Аутентификация для HTTP-подключений с приоритетом TACACS+, затем локальная база:

```
aaa authentication login http tacacs local
```

Устройство сначала попытается аутентифицировать пользователя через сервер **TACACS+**, если **TACACS+** недоступен, будет использована локальная база пользователей.

3.3 Команда aaa authorization

Команда **aaa authorization** используется для настройки авторизации **(разрешения)** действий пользователей на промышленном коммутаторе. Она определяет, какие команды и действия пользователь может выполнять в зависимости от уровня привилегий **(<priv_lvl>)** и типа подключения **(console, telnet, ssh)**. Авторизация выполняется через сервер **TACACS+.**

```
aaa authorization { console | telnet | ssh }
tacacs commands <priv_lvl> [ config-commands ]
```

3.3.1 Тип подключения:

- **console** — авторизация для подключений через консольный порт.

- **telnet** — авторизация для подключений через Telnet.
- **ssh** — авторизация для подключений через SSH.

tacacs

Указывает, что авторизация будет выполняться через сервер **TACACS+.**

commands <priv_lvl>

Включает авторизацию для выполнения команд.

<priv_lvl> — уровень привилегий, для которого будет применяться авторизация. Например:

1 — обычный пользователь.

15 — администратор.

config-commands (опционально)

Включает авторизацию для команд конфигурации (команд, которые изменяют настройки устройства). Если этот параметр не указан, авторизация для команд конфигурации не выполняется.

3.3.2 Примеры

Авторизация для **SSH-подключений** с уровнем привилегий 15:

```
aaa authorization ssh tacacs commands 15
```

Сервер **TACACS+** будет проверять, может ли пользователь выполнять команды с уровнем привилегий 15.

Авторизация для **Telnet**-подключений с уровнем привилегий 1 и проверкой команд конфигурации:

```
aaa authorization telnet tacacs commands 1 config-commands
```

Авторизация будет выполняться для **Telnet-подключений**.

Сервер **TACACS+** будет проверять, может ли пользователь выполнять команды с уровнем привилегий **1**.

Также будет выполняться авторизация для команд конфигурации.

Авторизация для консольных подключений с уровнем привилегий **15:**

```
aaa authorization console tacacs commands 15
```

Сервер **TACACS+** будет проверять, может ли пользователь выполнять команды с уровнем привилегий **15.**

- Команда **aaa authorization** полезна для контроля доступа пользователей к командам на устройстве;
- Уровень привилегий **(<priv_lvl>)** определяет, какие команды может выполнять пользователь. Например, уровень **15** позволяет выполнять все команды, включая команды конфигурации;
- Параметр **config-comm**ands добавляет дополнительный уровень безопасности, проверяя разрешение на выполнение команд, которые изменяют конфигурацию устройства.

3.4 Команды access management

Команды **access management** используются для управления доступом к устройству через различные интерфейсы **(веб-интерфейс, SNMP, Telnet)** на основе **IP-адресов и VLAN**.

Эти команды позволяют ограничить или разрешить доступ к устройству для определенных диапазонов **IP-адресов и VLAN.**

Базовая команда:

Отображает текущие настройки управления доступом.

```
access management
```

Добавление правила доступа:

Создает правило доступа для определенного диапазона IP-адресов и VLAN.

```
access management <access_id> <access_vid>
<start_addr> [ to <end_addr> ] { [ web ] [ snmp ] [ telnet ] | all }
```

Удаление правила доступа:

Удаляет правило доступа по его идентификатору (<access_id>).

```
no access management <access_id>
```

Параметры команд:

<access_id>

Уникальный идентификатор правила доступа (число). Используется для создания, изменения или удаления правила.

<access_vid>

Идентификатор VLAN (VLAN ID), к которой применяется правило доступа.

<start_addr>

Начальный IP-адрес диапазона, для которого настраивается доступ.

[to <end_addr>] (опционально)

Конечный IP-адрес диапазона. Если не указан, правило применяется только к одному IP-адресу (<start_addr>).

```
{ [ web ] [ snmp ] [ telnet ] | all }
```

Указывает, к каким интерфейсам применяется правило:

- **web** — доступ через веб-интерфейс.
- **snmp** — доступ через SNMP.
- **telnet** — доступ через Telnet.
- **all** — доступ через все интерфейсы (веб, SNMP, Telnet).

3.4.1 Примеры

Отображение текущих настроек доступа:

```
access management
```

Выводит список всех настроенных правил доступа.

Создание правила доступа для диапазона IP-адресов и VLAN 10 с доступом через веб-интерфейс и Telnet:

```
access management 1 10 192.168.1.10 to 192.168.1.20 web telnet
```

Создает правило с идентификатором 1.

Применяется к VLAN 10.
Разрешает доступ для IP-адресов от 192.168.1.10 до 192.168.1.20.

Создание правила доступа для одного IP-адреса и VLAN 20 с доступом через все интерфейсы:

```
access management 2 20 192.168.2.50 all
```

Создает правило с идентификатором 2.
Применяется к VLAN 20.
Разрешает доступ для IP-адреса 192.168.2.50.
Доступ разрешен через все интерфейсы (веб, SNMP, Telnet).

Удаление правила доступа с идентификатором 1:

```
no access management 1
```

Примечания:
Правила доступа применяются в порядке их создания.
Если правило не указано для определенного IP-адреса или VLAN, доступ по умолчанию может быть запрещен (в зависимости от конфигурации устройства).
Используйте команду access management без параметров для проверки текущих настроек.

3.5 Команда access-list ace

Команда access-list ace используется для настройки правил контроля доступа **(Access Control Entries, ACE)** на промышленном коммутаторе. Эти

правила позволяют фильтровать трафик на основе множества параметров, таких как **VLAN, MAC-адреса, IP-адреса, протоколы, порты и флаги. П**равила могут применяться для разрешения **(permit),** запрета **(deny)** или перенаправления **(redirect)** трафика, а также для настройки дополнительных функций**,** таких как ограничение скорости **(rate-limiter) и** зеркалирование **(mirror).**

access-list ace [update] <ace_id> [next { <ace_id_next> | last }] [ingress { switch <ingress_switch_id> | switchport { <ingress_switch_port_id> | <ingress_switch_port_list> } | interface { <port_type> <ingress_port_id> | (<port_type> [<ingress_port_list>]) } | any }] [policy <policy> [policy-bitmask <policy_bitmask>]] [tag { tagged | untagged | any }] [vid { <vid> | any }] [tag-priority { <tag_priority> | 0-1 | 2-3 | 4-5 | 6-7 | 0-3 | 4-7 | any }] [dmac-type { unicast | multicast | broadcast | any }] [frame-type { any | etype [etype-value { <etype_value> | any }] [smac { <etype_smac> | any }] [dmac { <etype_dmac> | any }] | arp [sip { <arp_sip> | any }] [dip { <arp_dip> | any }] [smac { <arp_smac> | any }] [arp-opcode { arp | rarp | other | any }] [arp-flag [arp-request { <arp_flag_request> | any }] [arp-smac { <arp_flag_smac> | any }] [arp-tmac { <arp_flag_tmac> | any }] [arp-len { <arp_flag_len> | any }] [arp-ip { <arp_flag_ip> | any }] [arp-ether { <arp_flag_ether> | any }]] | ipv4 [sip { <sipv4> | any }] [dip { <dipv4> | any }] [ip-protocol { <ipv4_protocol> | any }] [ip-flag [ip-ttl { <ip_flag_ttl> | any }] [ip-options { <ip_flag_options> | any }] [ip-fragment { <ip_flag_fragment> | any }]] | ipv4-icmp [sip { <sipv4_icmp> | any }] [dip { <dipv4_icmp> | any }] [icmp-type { <icmpv4_type> | any }] [icmp-code { <icmpv4_code> | any }] [ip-flag [ip-ttl { <ip_flag_icmp_ttl> | any }] [ip-options { <ip_flag_icmp_options> | any }] [ip-fragment { <ip_flag_icmp_fragment> | any }]] | ipv4-udp [sip { <sipv4_udp> | any }] [dip { <dipv4_udp> | any }] [sport { <sportv4_udp_start> [to <sportv4_udp_end>] | any }] [dport { <dportv4_udp_start> [to <dportv4_udp_end>] | any }] [ip-flag [ip-ttl { <ip_flag_udp_ttl> | any }] [ip-options { <ip_flag_udp_options> | any }] [ip-fragment { <ip_flag_udp_fragment> | any }]] | ipv4-tcp [sip { <sipv4_tcp> | any }] [dip { <dipv4_tcp> | any }] [sport { <sportv4_tcp_start> [to <sportv4_tcp_end>] | any }] [dport { <dportv4_tcp_start> [to <dportv4_tcp_end>] | any }] [ip-flag [ip-ttl { <ip_flag_tcp_ttl> | any }] [ip-options { <ip_flag_tcp_options> | any }] [ip-fragment { <ip_flag_tcp_fragment> | any }]] [tcp-flag [tcp-fin { <tcpv4_flag_fin> | any }] [tcp-syn { <tcpv4_flag_syn> | any }] [tcp-rst { <tcpv4_flag_rst> | any }] [tcp-psh { <tcpv4_flag_psh> | any }] [tcp-ack { <tcpv4_flag_ack> | any }] [tcp-urg { <tcpv4_flag_urg> | any }]] | ipv6 [next-header { <next_header> | any }] [sip { <sipv6> [sip-bitmask <sipv6_bitmask>] | any }] [hop-limit { <hop_limit> | any }] | ipv6-icmp [sip { <sipv6_icmp> [sip-bitmask <sipv6_bitmask_icmp>] | any }] [icmp-type { <icmpv6_type> | any }] [icmp-code { <icmpv6_code> | any }] [hop-limit { <hop_limit_icmp> | any }] | ipv6-udp [sip { <sipv6_udp> [sip-bitmask <sipv6_bitmask_udp>] | any }] [sport { <sportv6_udp_start> [to <sportv6_udp_end>] | any }] [dport { <dportv6_udp_start> [to <dportv6_udp_end>] | any }] [hop-limit { <hop_limit_udp> | any }] | ipv6-tcp [sip { <sipv6_tcp> [sip-bitmask <sipv6_bitmask_tcp>] | any }] [sport { <sportv6_tcp_start> [to <sportv6_tcp_end>] | any }] [dport { <dportv6_tcp_start> [to <dportv6_tcp_end>] | any }] [hop-limit { <hop_limit_tcp> | any }] [tcp-flag [tcp-fin { <tcpv6_flag_fin> | any }] [tcp-syn { <tcpv6_flag_syn> | any }] [tcp-rst { <tcpv6_flag_rst> | any }] [tcp-psh { <tcpv6_flag_psh> | any }] [tcp-ack { <tcpv6_flag_ack> | any }] [tcp-urg { <tcpv6_flag_urg> | any }]] }] [action { permit | deny | filter { switchport <filter_switch_port_list> | interface (<port_type> [<fliter_port_list>]) } }] [rate-limiter { <rate_limiter_id> | disable }] [evc-policer { <evc_policer_id> | disable }] [mirror [disable]] [logging [disable]] [shutdown [disable]] [lookup-second [disable]] [redirect { switchport { <redirect_switch_port_id> | <redirect_switch_port_list> } | interface { <port_type> <redirect_port_id> | (<port_type> [<redirect_port_list>]) } | disable }]

Параметры команды:

1. **[update]**
   Обновляет существующее правило ACE с указанным <ace_id>.

2. **<ace_id>**
   Уникальный идентификатор правила ACE (число).

3. **[next { <ace_id_next> | last }]**
   Указывает следующее правило ACE, которое будет проверяться после текущего:

4. **<ace_id_next> —** идентификатор следующего правила**.**

5. **last —** указывает, что текущее правило является последним в цепочке.

6. **[ingress { switch <ingress_switch_id> | switchport { <ingress_switch_port_id> | <ingress_switch_port_list> } | interface { <port_type> <ingress_port_id> | (<port_type> [<ingress_port_list>]) } | any }]**

Определяет входящий интерфейс или порт, к которому применяется правило:

- **switch —** идентификатор коммутатора.
- **switchport —** порт коммутатора или список портов.
- **interface —** тип интерфейса (например, Ethernet) и его идентификатор.
- **any —** правило применяется ко всем входящим интерфейсам.

7. **[policy <policy> [policy-bitmask <policy_bitmask>]]**

Настраивает политику и битовую маску для фильтрации.

8. **[tag { tagged | untagged | any }]**

Фильтрует трафик по тегу VLAN:

- **tagged** — только тегированный трафик.
- **untagged** — только нетэгированный трафик.

- **any** — любой трафик.

9. **[vid {<vid> | any }]**
   Фильтрует трафик по VLAN ID.

10. **[tag-priority {<tag_priority> | 0-1 | 2-3 | 4-5 | 6-7 | 0-3 | 4-7 | any }]**
Фильтрует трафик по приоритету тега (802.1p).

11. **[dmac-type { unicast | multicast | broadcast | any }]**
    Фильтрует трафик по типу MAC-адреса назначения.

12. **[frame-type { any | etype | arp | ipv4 | ipv4-icmp | ipv4-udp | ipv4-tcp | ipv6 | ipv6-icmp | ipv6-udp | ipv6-tcp }]**
    Фильтрует трафик по типу кадра:

    - **etype** — Ethernet-тип.
    - **arp** — ARP-пакеты.
    - **ipv4** — IPv4-пакеты.
    - **ipv6** — IPv6-пакеты.

13. Дополнительные параметры для каждого типа кадра (например, IP-адреса, порты, флаги).

14. **[action { permit | deny | filter }]**
    Определяет действие для трафика:

    - **permit —** разрешить трафик**.**
    - **deny —** запретить трафик**.**
    - **filter —** фильтровать трафик на указанных портах.

15. **[rate-limiter {<rate_limiter_id> | disable }]**
    Включает ограничение скорости для трафика.

16. **[evc-policer {<evc_policer_id> | disable }]**
    Включает полицер для EVC (Ethernet Virtual Connection**).**

17. **[mirror [disable]]**
    Включает или отключает зеркалирование трафика.

18. **[logging [disable]]**
    Включает или отключает логирование для правила.

19. **[shutdown [disable]]**
    Включает или отключает правило.

20. **[lookup-second [disable]]**
    Включает или отключает повторный поиск для правила.

21. **[redirect { switchport | interface | disable }]**
    Перенаправляет трафик на указанные порты или интерфейсы**.**

3.5.1 Примеры

Создание правила для разрешения трафика с **IP-адреса 192.168.1.10** на порт **Ethernet 1:**

```
access-list ace 1 ingress interface Ethernet 1 frame-type ipv4 sip 192.168.1.10 action
```

Запрет ARP-трафика на VLAN 10:

```
access-list ace 2 vid 10 frame-type arp action deny
```

Ограничение скорости для UDP-трафика на портах 3-5:

```
access-list ace 3 ingress switchport 3-5 frame-type ipv4-udp
rate-limiter 1 action permit
```

Перенаправление TCP-трафика на порт 10:

```
access-list ace 4 frame-type ipv4-tcp redirect switchport 10 action permit
```

Примечания:

– Правила ACE применяются в порядке их создания;
– Используйте команду show access-list для проверки текущих настроек;
– Для сложных сценариев можно комбинировать несколько параметров фильтрации.

4 Управление

4.1 Управление доступом и ограничение скорости

```
access-list rate-limiter [<rate_limiter_list>] { pps <pps_rate> | 10pps <pps10_rate>
100pps <pps100_rate> | 25kbps <kpbs25_rate> | 100kbps <kpbs100_rate> }
```

Ограничивает скорость трафика на основе заданных параметров (пакеты в секунду или килобиты в секунду). Используется для предотвращения перегрузки сети.

Переменные:

- **<rate_limiter_list>**: Список идентификаторов ограничителей скорости.
- **<pps_rate>**: Ограничение скорости в пакетах в секунду.
- **<pps10_rate>:** Ограничение скорости в 10 пакетах в секунду.
- **<kpbs25_rate>:** Ограничение скорости в 25 килобитах в секунду.
- **<kpbs100_rate>:** Ограничение скорости в 100 килобитах в секунду.

4.2 Агрегация каналов

```
aggregation mode { [ smac ] [ dmac ] [ ip ] [ port ] }*1
```

Настраивает агрегацию каналов **(LACP**) на основе **MAC-адресов, IP-адресов** или портов. Увеличивает пропускную способность и отказоустойчивость.

Переменные:

- **smac:** Агрегация на основе MAC-адреса источника.
- **dmac:** Агрегация на основе MAC-адреса назначения.
- **ip:** Агрегация на основе IP-адресов.
- **port:** Агрегация на основе портов.

4.3 Управление уведомлениями

```
alarm <alarm_name> <alarm_expression>
```

Создает уведомления на основе заданных условий (например, превышение пороговых значений).

Переменные:

- **<alarm_name>:** Имя уведомления.
- **<alarm_expression>:** Условие, при котором срабатывает уведомление.

4.4 Настройка баннеров

```
banner [ motd | login | exec ] <banner>
```

Настраивает сообщения, отображаемые при входе в систему **(MOTD, login, exec)**.

Переменные:

- **motd:** Сообщение дня (отображается при входе в систему).
- **login:** Сообщение при входе в систему.
- **exec:** Сообщение при входе в привилегированный режим.
- **<banner>:** Текст сообщения.

4.5 Мониторинг трансиверов

```
ddmi
```

Включает мониторинг состояния трансиверов **(DOM)** для отслеживания параметров, таких как температура и мощность сигнала.

4.6 Сброс настроек ограничения скорости

```
default access-list rate-limiter [ <rate_limiter_list> ]
```

Сбрасывает настройки ограничения скорости до значений по умолчанию.

Переменные:

- **<rate_limiter_list>:** Список идентификаторов ограничителей скорости для сброса.

4.7 Выполнение команд

```
do <command>
```

Выполняет команду в привилегированном режиме.

Переменные:

- **<command>:** Команда для выполнения в привилегированном режиме.

4.8 Настройка 802.1X

Управляет таймерами для аутентификации 802.1X (например, время неактивности или повторной аутентификации).

```
dot1x authentication timer inactivity <v_10_to_100000>
```

Переменные:

- **<v_10_to_100000>:** Время неактивности в секундах (от 10 до 100000).

Настраивает гостевую VLAN для неаутентифицированных устройств.

```
dot1x guest-vlan <value>
```

Переменные:

- **<value>:** Идентификатор гостевой VLAN.

4.9 Настройка EAPS

```
eaps <domain_id> hello-time <hello_time>
```

Настраивает Ethernet Automatic Protection Switching **(EAPS)** для защиты кольцевых топологий.

Переменные:

- **<domain_id>**: Идентификатор домена EAPS.
- **<hello_time>:** Интервал hello-пакетов в миллисекундах.

4.10 Управление EIP

```
eip control-flag <level>
```

Включает или настраивает Ethernet Industrial Protocol **(EIP)** для промышленных сетей.

Переменные:

- **<level>:** Уровень контроля (например, 0-7).

4.11 Управление паролями

```
enable password [ level <priv> ] <password>
```

Устанавливает пароль для входа в привилегированный режим.

Переменные:

- **<priv>**: Уровень привилегий (например, 15).
- **<password>**: Пароль для входа в привилегированный режим.

4.12 Настройка EPS

```
eps <inst> holdoff <hold>
```

Настраивает Ethernet Protection Switching **(EPS)** для защиты соединений.

Переменные:

- **<inst>:** Идентификатор экземпляра EPS.
- **<hold>:** Время задержки в миллисекундах.

4.13 Настройка ERPS

```
erps <group> guard <guard_time_ms>
```

Настраивает **Ethernet Ring Protection Switching (ERPS)** для защиты кольцевых топологий.

Переменные:

- **<group>:** Идентификатор группы **ERPS.**
- **<guard_time_ms>:** Время guard в миллисекундах.

4.14 Энергосбережение

```
erps <group> guard <guard_time_ms>
```

Включает режим энергосбережения **Ethernet (EEE).**

4.15 Настройка GVRP

```
gvrp max-vlans <maxvlans>
```

Включает протокол **GVRP (GARP VLAN Registration Protocol)** для автоматического распространения VLAN.

Переменные:

<maxvlans>: Максимальное количество VLAN, которые можно зарегистрировать через GVRP.

4.16 Команда help

Отображает список доступных команд или предоставляет справку по конкретной команде. Это встроенная команда, которая помогает администратору быстро получить информацию о синтаксисе и использовании команд.

4.16.1 Примеры

Ввод **help** отобразит общий список команд.

Ввод **help <command>** (например, **help ip dhcp**) покажет подробную справку по конкретной команде.

5 Настройка

5.1 Настройка STK-Ring

```
hiper-ring enable


hiper-ring mode [ ring-manager | ring-switch | rm | rs ]


hiper-ring port primary interface <port_type>
<primary_list> secondary interface <port_type>
<secondary_list>

hiper-ring recovery-delay { 500ms | 300ms }
```

Эти команды настраивают протокол **STK-Ring** для создания отказоустойчивых кольцевых топологий. Указываются режимы работы, интерфейсы и время восстановления.

Переменные:

- **ring-manager**: Устройство выступает в роли менеджера кольца.
- **ring-switch**: Устройство выступает в роли коммутатора кольца.
- **rm:** Сокращение для ring-manager.
- **rs:** Сокращение для ring-switch.

5.1.1 Назначение интерфейсов для STK-Ring.

```
hiper-ring port primary interface <port_type>
<primary_list> secondary interface <port_type>
<secondary_list>
```

Назначает первичные и вторичные интерфейсы для HiPER-Ring.

Переменные:

- **<port_type>**: Тип интерфейса (например, Ethernet).
- **<primary_list>:** Список первичных интерфейсов.
- **<secondary_list>:** Список вторичных интерфейсов.

5.1.2 Время восстановления

```
hiper-ring recovery-delay { 500ms | 300ms }
```

Устанавливает время восстановления после сбоя.

Переменные:

500ms: Время восстановления 500 миллисекунд.

300ms: Время восстановления 300 миллисекунд.

5.2 Настройка имени устройства

```
hostname <hostname>
```

Устанавливает имя устройства.

Переменные:

- **<hostname>:** Имя устройства (например, "Switch1").

5.3 Настройка интерфейсов

```
interface ( <port_type> [ <plist> ] )

interface llag <llag_id>

interface vlan <vlist>
```

Эти команды позволяют настраивать интерфейсы, включая физические порты, логическую агрегацию каналов **(LLAG)** и интерфейсы **VLAN.**

Переменные:

- **<port_type>:** Тип интерфейса (например, **Ethernet**).
- **<plist>:** Список интерфейсов (например, "1-10").
- **<llag_id>**: Идентификатор LLAG.
- **<vlist>:** Список VLAN (например, "10,20,30").

5.4 Настройка ARP-инспекции

```
ip arp inspection
ip arp inspection entry interface <port_type>
<in_port_type_id> <vlan_var> <mac_var> <ipv4_var>
ip arp inspection vlan <in_vlan_list>
ip arp inspection vlan <in_vlan_list> logging { deny | permit | all }
```

Эти команды настраивают **ARP-инспекцию** для защиты от **ARP-спуфинга.** Можно добавлять статические записи, включать инспекцию для **VLAN** и настраивать логирование.

- **Переменные:**
- **<port_type>:** Тип интерфейса.
- **<in_port_type_id>:** Идентификатор интерфейса.
- **<vlan_var>:** Идентификатор VLAN.
- **<mac_var>:** MAC-адрес.
- **<ipv4_var>:** IPv4-адрес.
- **<in_vlan_list>:** Список VLAN (например, "10,20,30").
- **deny | permit | all:** Уровень логирования.

5.5 Настройка DHCP

```
ip dhcp excluded-address <low_ip> [ <high_ip> ]
ip dhcp pool <pool_name>
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information policy { drop | keep | replace }
ip dhcp server
ip dhcp snooping
```

Эти команды настраивают **DHCP-сервер**, ретрансляцию и защиту от атак через **DHCP**.

Переменные:

- **<low_ip>:** Начальный IP-адрес диапазона.
- **<high_ip>:** Конечный IP-адрес диапазона (опционально).
- **<pool_name>:** Имя DHCP-пула.
- **drop | keep | replace: Политика обработки DHCP-сообщений.**

5.6 Настройка DNS

```
ip dns proxy
ip domain name { <v_domain_name> | dhcp [ ipv4 | ipv6 ]
[ interface vlan <v_vlan_id_dhcp> ] }
```

Эти команды настраивают проксирование DNS и доменное имя устройства.

Переменные:

- **<v_domain_name>:** Доменное имя.
- **dhcp:** Получение доменного имени через DHCP.
- **<v_vlan_id_dhcp>:** Идентификатор VLAN для DHCP.

5.7 Настройка HTTP/HTTPS

```
ip http secure-certificate
{ upload <url_file> [ pass-phrase <pass_phrase> ] | delete | generate }
ip http secure-redirect
ip http secure-server
```

Эти команды управляют сертификатами и настройками HTTPS.

Переменные:

- **<url_file>:** Путь к файлу сертификата.
- **<pass_phrase>:** Парольная фраза для сертификата.

5.8 Настройка IGMP

```
ip igmp host-proxy [ leave-proxy ]
ip igmp snooping
ip igmp snooping vlan <v_vlan_list>
ip igmp ssm-range <v_ipv4_mcast>
<ipv4_prefix_length>
ip igmp unknown-flooding
```

Эти команды настраивают IGMP для управления multicast-трафиком.

Переменные:

- **<v_vlan_list>:** Список VLAN (например, "10,20,30").
- **<v_ipv4_mcast>:** Multicast-адрес.
- **<ipv4_prefix_length>:** Длина префикса.

5.9 Настройка LLDP

```
lldp holdtime <val>
lldp med datum { wgs84 | nad83-navd88 | nad83-mllw }
lldp med fast <v_1_to_10>
lldp med location-tlv altitude { meters | floors } <v_word11>
lldp med location-tlv civic-addr { { country <country> } | { state | county | city | d
lldp med location-tlv elin-addr <v_word25>
lldp med location-tlv latitude { north | south } <v_word8>
lldp med location-tlv longitude { west | east } <v_word9>
lldp med media-vlan-policy <policy_index> { voice | voice-signaling | guest-voice-sign
lldp reinit <val>
lldp timer <val>
lldp transmission-delay <val>
```

Эти команды настраивают LLDP и LLDP-MED для обмена информацией между устройствами.

Переменные:

- **<val>**: Время удержания, таймеры и задержки.
- **<v_word11>, <v_word25>, <v_word8>, <v_word9>:** Различные параметры для LLDP-MED.
- **<v_vlan_id>**: Идентификатор VLAN.

- **<v_0_to_7>:** Приоритет VLAN.
- **<v_0_to_63>:** Значение DSCP.

5.10 Настройка логирования (Logging)

```
logging host { <ipv4_addr> | <domain_name> }
```

Настраивает удаленный сервер для отправки логов.

Переменные:

- **<ipv4_addr>:** IPv4-адрес сервера.
- **<domain_name>:** Доменное имя сервера.

Пример: logging host 192.168.1.100 — отправка логов на сервер с IP 192.168.1.100.)

5.10.1 Уровень логирования

```
logging level { informational | notice | warning | error }
```

Устанавливает уровень логирования.

Переменные:

- **informational:** Информационные сообщения.
- **notice:** Уведомления.

- **warning:** Предупреждения.
- **error:** Ошибки.

Пример: l**ogging level warning** — логирование только предупреждений и ошибок.

5.10.2 Настройка уведомлений

```
logging notification listen <name> level { informational | notice | warning | error }
```

Настраивает уведомления для определенного узла.

Переменные:

- **<name>:** Имя уведомления.
- **<node>:** Узел, для которого настраиваются уведомления.

Пример: logging notification listen syslog level error node1 —включение уведомления об ошибках для узла node1.

5.10.3 Включение логирования

```
logging on
```

Включает логирование.

5.11 Защита от петель (Loop Protection)

5.11.1 Включение защиты

```
loop-protect
```

Включает защиту от петель на уровне портов.

5.11.2 Время отключения

```
loop-protect shutdown-time <t>
```

Устанавливает время отключения порта при обнаружении петли.

Переменные:

- **<t>:** Время в секундах.

Пример: loop-protect shutdown-time 10 — отключение порта через 10 секунд.

5.11.3 Установка интервалов

```
loop-protect transmit-time <t>
```

Устанавливает интервал отправки тестовых пакетов для обнаружения петель.

Переменные:

- **<t>**: Время в секундах.

Пример: loop-protect transmit-time 5 — отправка пакетов каждые 5 секунд.

5.12 Управление таблицей MAC-адресов

5.12.1 Время старения

```
mac address-table aging-time <v_0_10_to_1000000>
```

Устанавливает время старения записей в таблице MAC-адресов.

Переменные:

- **<v_0_10_to_1000000>**: Время в секундах (от 10 до 1 000 000).

Пример: mac address-table aging-time 300 — записи стареют через 300 секунд.

5.12.2 Изучение MAC-адресов

```
mac address-table learning vlan <vlan_list>
```

Включает или отключает изучение MAC-адресов для указанных VLAN.

Переменные:

- **<vlan_list>:** Список VLAN (например, "10,20,30").

Пример: mac address-table learning vlan 10 — изучение MAC-адресов для VLAN 10.

5.12.3 Добавление статической записи

```
mac address-table static <v_mac_addr> vlan <v_vlan_id>
{ [ interface ( <port_type> [ <v_port_type_list> ] ) ]
[ sr <v_uint> ] [ psfp <v_uint_1> ] }
```

Добавляет статическую запись в таблицу MAC-адресов.

Переменные:

- **<v_mac_addr>:** MAC-адрес.
- **<v_vlan_id>:** Идентификатор VLAN.
- **<port_type>:** Тип интерфейса.
- **<v_port_type_list>:** Список портов.
- **<v_uint>, <v_uint_1>:** Дополнительные параметры.

Пример: mac address-table static 00:1A:2B:3C:4D:5E vlan 10 interface Ethernet 1 — статическая запись для MAC-адреса в VLAN 10 на порту Ethernet 1.

5.13 Настройка MEP (Maintenance End Point)

```
mep <inst> [ mip ] { up | down } domain
{ port | evc | vlan | tp-link | tunnel-tp | pw | lsp }
[ vid <vid> ] [ flow <flow> ] level <level>
[ interface <port_type> <port> ]
```

Настраивает конечную точку обслуживания (MEP) для мониторинга сети.

Переменные:

- **<inst>:** Идентификатор экземпляра MEP.
- **<vid>:** Идентификатор VLAN.
- **<flow>:** Идентификатор потока.
- **<level>:** Уровень MEP.
- **<port_type>:** Тип интерфейса.
- **<port>:** Порт.

Пример: mep 1 up domain vlan vid 10 level 5 interface Ethernet 1 — настройка MEP в VLAN 10 на порту Ethernet 1.

5.13.1 Настройка

```
mep <inst> ais [ fr1s | fr1m ] [ protect ]
```

Настраивает Alarm Indication Signal (AIS) для MEP.

Переменные:

- **fr1s:** Частота 1 секунда.
- **fr1m:** Частота 1 минута.
- **protect:** Включение защиты.

  Пример: mep 1 ais fr1s protect — AIS с частотой 1 секунда и защитой.

6 Настройка мониторинга

```
monitor session <session_number> [ destination { interface ( <port_type>
[ <di_list> ] ) | remote vlan <drvid> reflector-port <port_type>
<rportid> } | source { interface ( <port_type> [ <si_list> ] )
[ both | rx | tx ] | remote vlan <srvid> | vlan <source_vlan_list> | cpu
[ both | rx | tx ] } ]
```

Настраивает сессию мониторинга для зеркалирования трафика.

Переменные:

- **<session_number>:** Номер сессии.
- **<port_type>:** Тип интерфейса.
- **<di_list>, <si_list>**: Списки портов.
- **<drvid>, <srvid>:** Идентификаторы VLAN.
- **<rportid>:** Порт-отражатель.

Пример: monitor session 1 destination interface Ethernet 1 source interface Ethernet 2 — зеркалирование трафика с порта Ethernet 2 на порт Ethernet 1.

7 Настройка MRP

```
mrp domain <domain_id> advanced-mode { enabled | disabled }
```

Включает или отключает расширенный режим для домена MRP.

Переменные:

- **<domain_id>:** Идентификатор домена.

Пример: mrp domain 1 advanced-mode enabled — расширенный режим для домена 1.

7.1 Назначение VLAN

```
mrp domain <domain_id> vlan <vlan_id>
```

Назначает VLAN для домена MRP.

Переменные:

- **<vlan_id>:** Идентификатор VLAN**.**

Пример: mrp domain 1 vlan 10 — назначение VLAN 10 для домена 1.

8 Настройка MVR (Multicast VLAN Registration)

```
mvr name <mvr_name> channel <profile_name>
```

Настраивает канал для MVR.

Переменные:

- **<mvr_name>:** Имя MVR.
- **<profile_name>:** Имя профиля.

Пример: mvr name mvr1 channel profile1 — настройка канала profile1 для MVR mvr1.

8.1 Назначение VLAN

```
mvr vlan <v_vlan_list> [ name <mvr_name> ]
```

Назначает VLAN для MVR.

Переменные:

- **<v_vlan_list>:** Список VLAN**.**
- **<mvr_name>:** Имя MVR.

9 Настройка MVRP

```
mvrp managed vlan { all | none | [ add | remove | except ] <vlist> }
```

Управляет VLAN, зарегистрированными через MVRP.

Переменные:

- **all:** Все VLAN.
- **none:** Ни одна VLAN.
- **add:** Добавить VLAN.
- **remove:** Удалить VLAN.
- **except**: Исключить VLAN.
- **<vlist>:** Список VLAN.

Пример: mvrp managed vlan add 10,20 — добавление VLAN 10 и 20 в MVRP.

10 Настройка NTP (Network Time Protocol)

```
ntp
```

Включает **NTP** для синхронизации времени.

10.1 Настройка

```
ntp server <index_var> ip-address { <ipv4_var> | <ipv6_var> | <name_var> }
```

Настраивает NTP-сервер.

Переменные:

- **<index_var>**: Индекс сервера.

- **<ipv4_var>:** IPv4-адрес сервера.
- **<ipv6_var>:** IPv6-адрес сервера.
- **<name_var>:** Имя сервера.

Пример: ntp server 1 ip-address 192.168.1.100 — настройка NTP-сервера с IP 192.168.1.100.

11 Настройка портовой безопасности

```
port-security
```

Включает портовую безопасность.

11.1 Хранение MAC-адресов

```
port-security aging time <aging_time>
```

Влияет на то, сколько хранятся MAC-адреса в коммутаторе.

Переменные:

- **<aging_time>:** Время в секундах.

Пример: port-security aging time 300 — хранение адресов будет осуществляться 300 секунд.

11.2 Время удержания

```
port-security hold time <hold_time>
```

Устанавливает время удержания порта в заблокированном состоянии.

Переменные:

- **<hold_time>:** Время в секундах.

  Пример: **port-security hold time 60** — удержание порта 60 секунд.

12 Настройка QoS

```
qos map cos-dscp <cos> dpl <dpl> dscp { <dscp_num> |
{ be | af11 | af12 | af13 | af21 | af22 | af23 | af31
| af32 | af33 | af41 | af42 | af43 | cs1 | cs2 | cs3
| cs4 | cs5 | cs6 | cs7 | ef | va } }
```

Сопоставляет CoS (Class of Service) с DSCP (Differentiated Services Code Point).

Переменные:

- **<cos>:** то значение CoS (Class of Service), которое находится в диапазоне от 0 до 7.

CoS используется в заголовке **802.1Q (VLAN tagging)** для указания приоритета трафика на уровне кадров Ethernet.

Пример: **cos 5** соответствует высокоприоритетному трафику (например, голосовому).

• **dpl <dpl>:** это значение **Drop Precedence Level** (уровень приоритета отбрасывания пакетов), которое используется для управления сбросом пакетов при перегрузке сети.

Обычно принимает значения 0, 1 или 2, где:

0 — низкий приоритет отбрасывания.

1 — средний приоритет отбрасывания.

2 — высокий приоритет отбрасывания.

- **dscp:** это значение DSCP (Differentiated Services Code Point), которое используется для указания приоритета трафика на уровне IP-пакетов.

Может быть задано либо числом (от 0 до 63), либо предопределенным именем:

- **be (Best Effort)** — стандартный приоритет (обычно 0).

• **af11, af12, af13, af21, af22, af23, af31, af32, af33, af41, af42, af43** — классы Assured Forwarding (AF), где первая цифра указывает класс, а вторая — приоритет отбрасывания.

• **cs1, cs2, cs3, cs4, cs5, cs6, cs7** — классы Class Selector (CS), которые соответствуют старым значениям IP Precedence.

• **ef (Expedited Forwarding)** — используется для трафика с минимальной задержкой (например, VoIP).

• **va (Voice Admit)** — используется для трафика, который должен быть гарантированно доставлен (например, голосовые вызовы).

12.1 Штормовой трафик

```
qos storm { unicast | multicast | broadcast } <rate> [ fps | kfps | kbps | mbps ]
```

Ограничивает штормовой трафик.

Переменные:

- **unicast:** Уникастный трафик.
- **multicast:** Мультикастный трафик.
- **broadcast:** Бродкастный трафик.
- **<rate>:** Ограничение скорости.
- **fps, kfps, kbps, mbps:** Единицы измерения.

Пример: qos storm broadcast 100 kbps — ограничение бродкастного трафика до 100 кбит/с.

13 Настройка RADIUS

```
radius-server attribute 32 <id>
```

Настраивает атрибут 32 (NAS-Identifier) для RADIUS.

Переменные:

- **<id>:** Идентификатор.

Пример: radius-server attribute 32 NAS1 — настройка атрибута 32 как "NAS1".

13.1 Настройка

```
radius-server host <host_name> [ auth-port <auth_port> ]
[ acct-port <acct_port> ] [ timeout <seconds> ] [ retransmit <retries> ]
[ key { [ unencrypted ] <unencrypted_key> | encrypted <encrypted_key> } ]
```

Настраивает RADIUS-сервер.

Переменные:

- **<host_name>:** Имя или IP-адрес сервера.
- **<auth_port>:** Порт для аутентификации (по умолчанию 1812).
- **<acct_port>:** Порт для учета (по умолчанию 1813).
- **<seconds>:** Таймаут.
- **<retries>:** Количество попыток.
- **<unencrypted_key>:** Ключ в открытом виде.
- **<encrypted_key>:** Зашифрованный ключ.

Пример: radius-server host 192.168.1.100 auth-port 1812 key secret123 — настройка RADIUS-сервера с ключом "secret123".

14 Настройка RMON

```
rmon alarm <id> { ifInOctets | ifInUcastPkts | ifInNUcastPkts | ifInDiscards |
ifInErrors | ifInUnknownProtos | ifOutOctets | ifOutUcastPkts | ifOutNUcastPkts
| ifOutDiscards | ifOutErrors } <ifIndex> <interval> { absolute | delta } rising-thres
<rising_threshold> [ <rising_event_id> ] falling-threshold <falling_threshold>
[ <falling_event_id> ] { [ rising | falling | both ] }
```

Настраивает алармы RMON.

Переменные:

- **<id>:** Идентификатор аларма.
- **<ifIndex>:** Индекс интерфейса.
- **<interval>:** Интервал проверки.
- **<rising_threshold>:** Верхний порог.
- **<falling_threshold>:** Нижний порог.
- **<rising_event_id>, <falling_event_id>:** Идентификаторы событий.

Пример: rmon alarm 1 ifInOctets 1 60 absolute rising-threshold 1000 1 falling-threshold 500 2 — аларм для входящих октетов на интерфейсе 1.

15 Настройка STP

```
spanning-tree mode { stp | rstp | mstp }
```

Устанавливает режим STP.

Переменные:

- **stp:** Стандартный STP.

- **rstp:** Rapid STP.
- **mstp:** Multiple STP.

Пример: spanning-tree mode rstp — включение Rapid STP.

15.1 Установка приоритета

```
spanning-tree mst <instance> priority <prio>
```

Устанавливает приоритет для экземпляра MSTP.

Переменные:

- **<instance>:** Идентификатор экземпляра.
- **<prio>:** Приоритет (0-61440, с шагом 4096)**.**

Пример: spanning-tree mst 1 priority 8192 — приоритет 8192 для экземпляра 1.

16 Настройка SVL

```
svl fid <fid> vlan <vlan_list>
```

Настраивает общее изучение VLAN (Shared VLAN Learning) для указанного FID (Forwarding Identifier).

Переменные:

- **<fid>:** Идентификатор FID.

- **<vlan_list>:** Список VLAN.

Пример: svl fid 1 vlan 10,20 — общее изучение **VLAN 10 и 20 для FID 1.**

17 Настройка VLAN

```
vlan <vlist>
```

Создает или настраивает **VLAN.**

Переменные:

<vlist>: Список VLAN (например, "10,20,30").

Пример: vlan 10 — создание VLAN 10.

17.1 Настройка EtherType

```
vlan ethertype s-custom-port <etype>
```

Настраивает EtherType для пользовательских портов.

Переменные:

- **<etype>:** Значение EtherType (например, 0x8100 для VLAN).

Пример: vlan ethertype s-custom-port 0x8100 — настройка EtherType 0x8100.

17.2 Настройка протоколов

```
vlan protocol { { eth2 { <etype> | arp | ip | ipx | at } }
| { snap { <oui> | rfc-1042 | snap-8021h } <pid> }
| { llc <dsap> <ssap> } } group <grp_id>
```

Настраивает протоколы для VLAN.

Переменные:

- **<etype>:** EtherType.
- **<oui>:** OUI (Organizationally Unique Identifier).
- **<pid>:** Идентификатор протокола.
- **<dsap>, <ssap>:** Значения DSAP и SSAP для LLC.
- **<grp_id>:** Идентификатор группы.

Пример: vlan protocol eth2 ip group 1 — настройка протокола IP для группы 1.

18 Настройка Voice VLAN

```
voice vlan
```

Включает Voice VLAN для VoIP-трафика.

18.1 Время старения записей

```
voice vlan aging-time <aging_time>
```

Устанавливает время старения записей Voice VLAN.

Переменные:

- **<aging_time>:** Время в секундах.

Пример: voice vlan aging-time 300 — старение записей через 300 секунд.

18.2 Настройка OUI

```
voice vlan oui <oui> [ description <description> ]
```

Настраивает OUI для Voice VLAN.

Переменные:

- **<oui>:** OUI (например, 0001E3 для Cisco).
- **<description>:** Описание.

Пример: voice vlan oui 0001E3 description Cisco — настройка OUI для Cisco.

19 Настройка привелегий для Web-интерфейса

```
web privilege group <group_name> level
{ [ configRoPriv <configRoPriv> ] [ configRwPriv <configRwPriv> ]
[ statusRoPriv <statusRoPriv> ] [ statusRwPriv <statusRwPriv> ] }*1
```

Настраивает привилегии для групп в веб-интерфейсе.

Переменные:

- **<group_name>:** Имя группы.
- **<configRoPriv>, <configRwPriv>:** Привилегии для чтения/записи конфигурации.
- **<statusRoPriv>, <statusRwPriv>**: Привилегии для чтения/записи статуса.

Пример: web privilege group admin level configRwPriv 15 statusRwPriv 15 — полные права для группы "admin".

20 Настройка UDLD

```
udld { aggressive | enable | message time-interval <v_interval> }
```

Настраивает UDLD для обнаружения односторонних соединений.

Переменные:

- **aggressive:** Агрессивный режим. (Расширенный режим работы протокола UDLD, который обеспечивает более быстрое и строгое реагирование на обнаружение однонаправленных соединений.)
- **enable:** Включение UDLD.
- **<v_interval>:** Интервал отправки сообщений.

Пример: udld aggressive — включение агрессивного режима UDLD.

21 Настройка UPnP (Universal Plug and Play)

```
upnp
```

Включает UPnP.

21.1 Длительность рекламы

```
upnp advertising-duration <v_100_to_86400>
```

Устанавливает длительность рекламы UPnP.

Переменные:

- **<v_100_to_86400>:** Время в секундах (от 100 до 86400).

Пример: upnp advertising-duration 3600 — длительность рекламы 1 час.

21.2 Режим IP-адресации

```
upnp ip-addressing-mode { dynamic | static }
```

Настраивает режим IP-адресации для UPnP.

Переменные:

- **dynamic:** Динамический режим.
- **static:** Статический режим.

Пример: upnp ip-addressing-mode dynamic — динамический режим IP-адресации

22 Настройка пользователей

```
username { default-administrator | <input_username> }
privilege <priv> password { unencrypted <unencry_password>
| encrypted <encry_password> | none }
```

Создает или настраивает пользователя.

Переменные:

- **<input_username>:** Имя пользователя.
- **<priv>:** Уровень привилегий (0-15).
- **<unencry_password>:** Пароль в открытом виде.
- **<encry_password>:** Зашифрованный пароль.

Пример: username admin privilege 15 password admin123 — создание пользователя "**admin**" с паролем "**admin123**" и максимальными привилегиями.

23 Настройка тепловой защиты (Thermal Protection)

```
thermal-protect grp <grp_list> temperature <new_temp>
```

Настраивает тепловую защиту для группы устройств.

Переменные:

- **<grp_list>:** Список групп.
- **<new_temp>:** Новая температура.

Пример: thermal-protect grp 1 temperature 50 — установка температуры 50°C для группы 1.

24 Настройка TACACS+

```
tacacs-server deadtime <minutes>
```

Устанавливает время ожидания перед повторным использованием недоступного сервера.

Переменные:

- **<minutes>**: Время в минутах.

  Пример: tacacs-server deadtime 5 — время ожидания 5 минут.

24.1 Настройка

```
tacacs-server host <host_name> [ port <port> ] [ timeout <seconds> ]
[ key { [ unencrypted ] <unencrypted_key> | encrypted <encrypted_key> } ]
```

Настраивает TACACS+-сервер.

Переменные:

- **<host_name>:** Имя или IP-адрес сервера.
- **<port>**: Порт (по умолчанию 49).
- **<seconds>:** Таймаут.
- **<unencrypted_key>:** Ключ в открытом виде.

- **<encrypted_key>:** Зашифрованный ключ.

Пример: tacacs-server host 192.168.1.100 key secret123 — настройка TACACS+-сервера с ключом **"secret123".**

25 Настройка sFlow

```
sflow agent-ip { ipv4 <v_ipv4_addr> | ipv6 <v_ipv6_addr> }
```

Настраивает IP-адрес агента sFlow.

Переменные:

- **<v_ipv4_addr>:** IPv4-адрес.
- **<v_ipv6_addr>:** IPv6-адрес.

Пример: sflow agent-ip ipv4 192.168.1.1 — настройка IPv4-адреса агента.

25.1 Настройка адреса

```
sflow collector-address [ receiver <rcvr_idx_list> ]
[ <ipv4_var> | <ipv6_var> | <domain_name> ]
```

Настраивает адрес коллектора sFlow.

Переменные:

- **<rcvr_idx_list>:** Список индексов приемников.
- **<ipv4_var>, <ipv6_var>:** IP-адрес коллектора.
- **<domain_name>:** Доменное имя коллектора.

Пример: sflow collector-address 192.168.1.100 — настройка коллектора с IP 192.168.1.100.

26 Настройка SNMP

```
snmp-server access <group_name> model { v1 | v2c | v3 | any }
level { auth | noauth | priv } [ read <view_name> ] [ write <write_name> ]
```

Настраивает доступ к SNMP для группы.

Переменные:

- **<group_name>:** Имя группы.
- **v1, v2c, v3, any:** Версия SNMP.
- **auth, noauth, priv:** Уровень безопасности**.**
- **<view_name>:** Имя представления для чтения.
- **<write_name>:** Имя представления для записи**.**

Пример: snmp-server access group1 v3 auth read view1 write view2 — настройка доступа для группы "group1".

26.1 Настройка SNMP

```
snmp-server community <v3_comm> [ { ip-range <v_ipv4_addr>
<v_ipv4_netmask> | ipv6-range <v_ipv6_subnet> } ]
{ <v3_sec> | encrypted <v3_sec_enc> }
```

Настраивает SNMP-сообщество.

Переменные:

- **<v3_comm>:** Имя сообщества.
- **<v_ipv4_addr>, <v_ipv4_netmask>:** Диапазон IPv4.
- **<v_ipv6_subnet>:** Диапазон IPv6.
- **<v3_sec>:** Уровень безопасности.
- **<v3_sec_enc>:** Зашифрованный уровень безопасности.

snmp-server community public ip-range 192.168.1.0 255.255.255.0 — настройка сообщества "public" для сети 192.168.1.0/24.

26.2 Установка информации

```
snmp-server contact <v_line255>
```

Устанавливает контактную информацию для SNMP.

Переменные:

- **<v_line255>: Контактная информация (до 255 символов).**

Пример: snmp-server contact admin@example.com — настройка контактной информации.

26.3 Местоположение

```
snmp-server location <v_line255>
```

Устанавливает местоположение для SNMP.

Переменные:

- **<v_line255>:** Местоположение (до 255 символов).

Пример: snmp-server location "Data Center 1" — настройка местоположения.

26.4 Настройка SNMP-трапов

```
snmp-server trap <source_name> { [ id <filter_id> ]
| [ <oid_subtree> { include | exclude } ] }
```

Настраивает SNMP-трапы.

Переменные:

- **<source_name>**: Имя источника.
- **<filter_id>:** Идентификатор фильтра.
- **<oid_subtree>:** Поддерево OID.
- **include, exclude**: Включение или исключение поддерева.

Пример: snmp-server trap trap1 id 1 — настройка трапа с идентификатором 1.

26.5 Настройка пользователя

```
snmp-server user <username> engine-id <engineID>
[ { md5 { <md5_passwd> | { encrypted <md5_passwd_encrypt> } } | sha
{ <sha_passwd> | { encrypted <sha_passwd_encrypt> } } }
[ priv { des | aes } { <priv_passwd> | { encrypted <priv_passwd_encrypt> } } ] ]
```

Настраивает пользователя SNMP.

Переменные:

- **<username>:** Имя пользователя.
- **<engineID>**: Идентификатор Engine.
- **<md5_passwd>, <sha_passwd>**: Пароль для MD5 или SHA.
- **<priv_passwd>:** Пароль для шифрования.

Пример: snmp-server user admin engine-id 8000000001020304 md5 password123 — настройка пользователя "admin" с MD5-аутентификацией.

26.6 Настройка представления

```
snmp-server view <view_name> <oid_subtree> { include | exclude }
```

Настраивает представление SNMP.

Переменные:

- **<view_name>:** Имя представления.
- **<oid_subtree>:** Поддерево OID.
- **include, exclude**: Включение или исключение поддерева.

Пример: snmp-server view view1 1.3.6.1.2.1 include — настройка представления для OID 1.3.6.1.2.1.

27 Настройка STP

```
spanning-tree aggregation
```

Включает агрегацию портов для STP.

27.1 Фильтрация BPDU

```
spanning-tree edge bpdu-filter
```

Включает фильтрацию BPDU на edge-портах.

27.2 Защита BPDU

```
spanning-tree edge bpdu-guard
```

Включает защиту BPDU на edge-портах.

27.3 Режим STP

```
spanning-tree mode { stp | rstp | mstp }
```

Устанавливает режим STP.

Переменные:

- **stp:** Стандартный STP.
- **rstp:** Rapid STP.
- **mstp:** Multiple STP.

Пример: spanning-tree mode rstp — включение Rapid STP.

27.4 Установка приоритета

```
spanning-tree mst <instance> priority <prio>
```

Устанавливает приоритет для экземпляра MSTP.

Переменные:

- **<instance>:** Идентификатор экземпляра.
- **<prio>:** Приоритет (0-61440, с шагом 4096).

Пример: spanning-tree mst 1 priority 8192 — приоритет 8192 для экземпляра 1.

27.5 Назначение VLAN

```
spanning-tree mst <instance> vlan <v_vlan_list>
```

Назначает VLAN для экземпляра MSTP.

Переменные:

- **<v_vlan_list>:** Список VLAN.

Пример: spanning-tree mst 1 vlan 10,20 — назначение VLAN 10 и 20 для экземпляра 1.

28 Alarm/Аварийные сигналы

```
alarm
```

Команда **alarm** используется для управления аварийными сигналами **(alarms)** на промышленном коммутаторе. Аварийные сигналы — это уведомления о критических событиях.

Переменные:

- **alarm suppress**

Описание: подавляет (отключает) указанную тревогу. Это полезно, если вы хотите временно отключить уведомления о конкретной проблеме.

Синтаксис:

- **alarm suppress <alarm_name>**
- **<alarm_name>:** Имя тревоги, которую нужно подавить**.**

Пример:

Если ввести alarm suppress, коммутатор покажет список доступных тревог (в нашем случае это <kword127> — это пример имени тревоги).

29 Clear/Очистка

```
Clear
```

Из вывода команды **clear ?** (если он доступен) можно увидеть список доступных опций.

29.1 Очистка статистики

```
clear access management statistics
```

Эта команда используется для очистки статистики, связанной с управлением доступом (access management) на коммутаторе. Управление доступом может включать в себя данные о попытках входа, аутентификации, авторизации и других событиях, связанных с доступом к устройству.

29.2 Обнуление статистики

```
clear access management statistics
```

После выполнения этой команды все счетчики статистики управления доступом будут обнулены.

- **<cr> (Carriage Return)**
- Команда **clear access management statistics** не принимает дополнительных параметров после **<cr> (Carriage Return,** т.е. нажатие **Enter).**

Ввод лишних символов или параметров приводит к синтаксической ошибке.

29.3 Очистка статистики ACE и ACL

```
clear access-list ace statistics
```

Эта команда используется для очистки статистики, связанной с записями ACE (Access Control Entry) в ACL (Access Control List). ACE — это отдельные правила в ACL, которые определяют, какой трафик разрешен или запрещен.

После выполнения этой команды все счетчики трафика для **ACE** будут обнулены.

Очистка статистики **ACE:**

Команда сбрасывает счетчики трафика, связанные с каждым правилом **ACE в ACL**. Это может быть полезно, если вы хотите начать мониторинг трафика "с чистого листа" или устранить проблемы, связанные с накоплением данных.

Статистика ACE:

Включает данные о количестве пакетов и байтов, которые соответствуют каждому правилу ACE.

После очистки:

Все счетчики будут обнулены, и сбор статистики начнется заново.

Введите clear access-list ?, чтобы увидеть доступные подкоманды:

test01# clear access-list ?
ace Access list entry

Уточнение подкоманды:

Введите **clear access-list ace ?**, чтобы увидеть доступные опции для **ACE:**

test01# clear access-list ace ?
statistics Traffic statistics

Выполнение команды:

Введите команду для очистки статистики:

clear access-list ace statistics

29.4 Очистка статистики IEEE 802.1X

```
clear dot1x statistics
```

Эта команда используется для очистки статистики, связанной с протоколом **IEEE 802.1X (Port-Based Network Access Control)**. Статистика может включать данные о попытках аутентификации, успешных и неудачных подключениях, а также другие события, связанные с **802.1X.**

Пример:

Очищает статистику 802.1X для конкретного интерфейса или группы интерфейсов.

clear dot1x statistics
clear dot1x statistics interface <тип_интерфейса>

Доступные типы интерфейсов:

1. *: Все интерфейсы.
2. FastEthernet: Интерфейсы Fast Ethernet.
3. GigabitEthernet: Интерфейсы Gigabit Ethernet.
4. 10GigabitEthernet: Интерфейсы 10 Gigabit Ethernet.

Пример:

1. Очищает статистику **802.1X** для интерфейса **GigabitEthernet 0/1.**

clear dot1x statistics interface GigabitEthernet 0/1

2. Очищает статистику **802.1X** для всех интерфейсов.

clear dot1x statistics interface *

Просмотр доступных опций

Введите **clear dot1x ?,** чтобы увидеть доступные подкоманды:

test01# clear dot1x ?
statistics Clears the statistics counters

Уточнение подкоманды

Введите clear **dot1x statistics ?,** чтобы увидеть доступные опции:

test01# clear dot1x statistics ?
interface Interface
<cr>

Для очистки статистики на всех интерфейсах:

- **clear dot1x statistics**

Для очистки статистики на конкретном интерфейсе:

- **clear dot1x statistics interface GigabitEthernet 0/1**

29.5 Очистка данных ERPS

```
clear erps
```

Очищает данные, связанные с протоколом **ERPS (Ethernet Ring Protection Switching). ERPS** используется для защиты кольцевых топологий **Ethernet**.

Синтаксис

```
clear erps <группа> [statistics]
```

- **<группа>:** Номер группы ERPS (от 1 до 64).
- **statistics:** Очищает статистику для указанной группы ERPS.

Примеры:

Очистка данных для группы ERPS 1:

- **clear erps 1**

Очистка статистики для группы ERPS 1:

- **clear erps 1 statistics**

29.6 Очистка данных IPv6

```
clear ip/ipv6
```

Очищает данные, связанные с **IPv6.** Включает подкоманды для работы с **MLD (Multicast Listener Discovery)** и таблицей соседей **(neighbors).**

Подкоманды

```
clear ipv6 mld snooping
```

Очищает данные, связанные с **MLD snooping** (протокол для управления multicast-трафиком в IPv6).

Синтаксис

```
clear ipv6 mld snooping [statistics | vlan <vlan_id>]
```

- **statistics:** Очищает статистику **MLD snooping.**
- **vlan <vlan_id>:** Очищает данные **MLD snooping** для указанного **VLAN**.

Примеры:

- Очистка статистики **MLD snooping:**

clear ipv6 mld snooping statistics

- Очистка данных **MLD snooping для VLAN 10:**

clear ipv6 mld snooping vlan 10

clear ipv6 neighbors

- Очистка таблицы соседей **IPv6** (аналог ARP-таблицы для IPv6).

clear ipv6 neighbors

29.7 Очистка данных EPS

```
clear eps
```

Очищает данные, связанные с EPS (Ethernet Protection Switching). EPS используется для защиты Ethernet-соединений.

Синтаксис

```
clear eps <номер_экземпляра>
```

<номер_экземпляра>: Номер экземпляра EPS (целое число).

Пример

Очистка данных для экземпляра **EPS 1:**

clear eps 1

Просмотр доступных опций

Для ERPS:

```
clear erps ?
```

Для IPv6:

```
clear ipv6 ?
```

Выполнение команды

Очистка статистики **ERPS для группы 1:**
clear erps 1 statistics

Очистка таблицы **соседей IPv6:**
clear ipv6 neighbors

29.8 Просмотр опций

```
clear ip
```

Просмотр опций

clear ip ?

Опции:

- **acd:** Очистка данных **Address Conflict Detection (обнаружение конфликтов IP-адресов).**
- **arp:** Очистка **ARP-таблицы (соответствие IP и MAC-адресов).**
- **dhcp:** Очистка данных **DHCP (статистика, snooping, relay и т.д.).**
- **igmp:** Очистка данных **IGMP (управление multicast-трафиком).**

- **ospf:** Очистка данных **OSPF (протокол маршрутизации).**
- **statistics:** Очистка общей статистики **IP-трафика.**

29.8.1 Очистка данных ACD

```
clear ip acd
```

Просмотр опций:

clear ip acd ?

- **<cr>:** очищает все данные ACD.

29.8.2 Очистка данных ARP

```
clear ip arp
```

Просмотр опций:

clear ip arp ?

- **<cr>:** очищает всю ARP-таблицу.

29.8.3 Очистка DHCP

```
clear ip dhcp
```

Просмотр опций:

clear ip dhcp ?

- **detailed:** Очистка подробной статистики **DHCP.**
- **relay:** Очистка данных **DHCP relay.**
- **server:** Очистка данных **DHCP сервера.**
- **snooping:** Очистка данных **DHCP snooping.**

29.8.4 Очистка статистика

```
clear ip dhcp detailed statistics
```

Просмотр опций:

clear ip dhcp detailed statistics ?

- **all:** очищает всю статистику **DHCP.**
- **client:** очищает статистику **DHCP клиента.**
- **helper:** очищает статистику **DHCP helper.**
- **relay:** очищает статистику **DHCP relay.**
- **server:** очищает статистику **DHCP сервера.**
- **snooping:** очищает статистику **DHCP snooping.**

29.8.5 Очистка IGMP

```
clear ip igmp
```

Просмотр опций:

clear ip igmp ?

- **snooping:** Очистка данных **IGMP snooping.**

Просмотр опций:

clear ip igmp snooping ?

- **statistics:** очищает статистику **IGMP snooping.**
- **vlan:** очищает данные **IGMP snooping** для указанного **VLAN.**

29.8.6 Очистка OSPF

```
clear ip ospf
```

Просмотр опций:

clear ip ospf ?

- **process:** очищает данные **OSPF** процесса.

clear ip ospf process

Просмотр опций:

clear ip ospf process ?

- **<cr>:** Очищает все данные **OSPF.**

clear ip statistics

Просмотр опций:

clear ip statistics ?

- **<cr>:** очищает общую статистику **IP-трафика.**

Итог:

- **clear ip acd:** очистка данных ACD.
- **clear ip arp:** очистка ARP-таблицы.
- **clear ip dhcp:** очистка данных DHCP (статистика, snooping, relay и т.д.).
- **clear ip igmp:** очистка данных IGMP (multicast-трафик).
- **clear ip ospf:** очистка данных OSPF (маршрутизация).
- **clear ip statistics:** очистка общей статистики IP-трафика.

30 Configure/Режим конфигурации

Переход в режим конфигурации для изменения настроек коммутатора.

Просмотр опций:

configure ?

- **terminal:** переход в терминальный режим конфигурации.

30.1 Переход в глобальный режим

```
configure terminal
```

Переход в глобальный режим конфигурации, где можно изменять настройки коммутатора.

Просмотр опций:

configure terminal ?

- **<cr>:** нажмите Enter для перехода в режим конфигурации.

Пример

Переход в режим конфигурации:

configure terminal

Пример:

interface GigabitEthernet 0/1ip address 192.168.1.1 255.255.255.0

31 Copy/Копирование файлов

Используется для копирования файлов или конфигураций между источниками и назначениями (ф**лэш-память, TFTP-сервер и др.).**

31.1 Работа с конфигурационными файлами.

```
copy conf
```

Работа с конфигурационными файлами.

Синтаксис

<flash:filename | tftp://server/path-and-filename>

Доп. Опции:

- **default-config:** копирование/проверка конфигурации по умолчанию.

- **syntax-check:** Проверка синтаксиса конфигурации.

Фильтрация вывода:

- **| begin <строка>:** показать вывод, начиная с указанной строки.
- **| exclude <строка>:** исключить строки, содержащие указанный текст.
- **| include <строка>:** показать только строки, содержащие указанный текст.

Пример:

copy conf default-config syntax-check | include "vlan"

31.1.1 Копирование конфигурации

```
copy default-config
```

Копирование конфигурации по умолчанию.

- **logo:** Работа с файлом логотипа.

Пример:

copy default-config logo tftp://192.168.1.100/logo.bmp

31.1.2 Работа с текущей конфигурацией.

```
copy running-config
```

Пример:

startup-config: Сохранение текущей конфигурации в стартовую.

copy running-config startup-config

31.1.3 Работа со стартовой конфигурацией.

```
copy startup-config
```

running-config: Загрузка стартовой конфигурации в текущую.

Пример:

copy startup-config running-config

Общие параметры:

- **<url_file>:** путь к файлу. Формат:
- **flash:**<имя_файла> (например, flash:backup.cfg).
- tftp://<сервер>/<путь> (например, tftp://192.168.1.100/config.cfg).

Правила именования файлов:

- Допустимые символы: A-Za-z, 0-9, ., -, _.
- Максимальная длина: 63 символа.
- Запрещено начинать с (-) и использовать только (.)

32 Delete/Удаление

Удаляет файлы из флэш-памяти коммутатора. Используется для освобождения места или удаления ненужных файлов (например, старых конфигураций или прошивок).

```
delete <url_file>
```

<url_file>: Путь к файлу в формате **flash:<имя_файла>**.

Правила именования файлов:

- Допустимые символы: **A-Za-z, 0-9, ., -, _.**
- Максимальная длина: **63 символа.**

Запрещено:

- Начинать имя файла с (-).
- Использовать только (.) в имени файла.

Пример удаления файла конфигурации:

delete flash:old_config.cfg

Удаление файла прошивки:

delete flash:firmware_v1.bin

33 DIR/Отображение файловой системы

Отображает содержимое файловой системы (например, флэш-памяти). Позволяет просматривать список файлов, их размеры и даты создания.

Синтаксис:

dir [<путь>] | <фильтр>

- **<путь>:** Указывает директорию или устройство (например, flash:).
- **| <фильтр>:** Фильтрация вывода.

Фильтры вывода:

- **begin <строка>**
  Показывает вывод, начиная с первой строки, содержащей указанный текст.

- **dir flash: | begin "config"**

- **exclude <строка>**

Исключает строки, содержащие указанный текст.

Пример:

dir flash: | exclude "backup"

include <строка>

Показывает только строки, содержащие указанный текст.

Пример:

dir flash: | include "firmware"

34 DISABLE/Пользовательский режим

Переводит пользователя из привилегированного режима (режима **enable)** в пользовательский режим. В пользовательском режиме доступны только базовые команды.

Синтаксис:

disable [<уровень_привилегий>]

- **<уровень_привилегий>**: Опциональный параметр, указывающий уровень привилегий (от 0 до 15). По умолчанию — уровень 1 (пользовательский режим).

Примеры использования:

Переход в пользовательский режим с указанием уровня привилегий (например, уровень 0):

- **disable 0**

35 DO/Команды режима EXEC

Позволяет выполнять команды режима EXEC
(например, show, ping, traceroute) из режима конфигурации.

Синтаксис:

do <команда_EXEC>

- **<команда_EXEC>:** Любая команда, которая доступна в режиме **EXEC** (например, **show running-config, ping, traceroute**).

Примеры использования

Проверка текущей конфигурации из режима конфигурации:

- **do show running-config**

Проверка доступности удаленного узла из режима конфигурации:

- **do ping 192.168.1.1**

Просмотр таблицы MAC-адресов из режима конфигурации:

- **do show mac address-table**

36 DOT1X INITIALIZE/Инициация аутентификации

Принудительно инициирует повторную аутентификацию устройства через протокол 802.1X.

Синтаксис:

- **dot1x initialize [interface <тип_интерфейса>]**
- **interface <тип_интерфейса>**: указывает интерфейс для повторной аутентификации. Если не указан, команда применяется ко всем интерфейсам.

Опции интерфейса

- ***:** Все интерфейсы.
- **FastEthernet:** Интерфейсы Fast Ethernet.
- **GigabitEthernet:** Интерфейсы Gigabit Ethernet.
- **10GigabitEthernet:** Интерфейсы 10 Gigabit Ethernet.

Примеры использования

Повторная аутентификация на всех интерфейсах:

- **dot1x initialize**

Повторная аутентификация на конкретном интерфейсе:

- **dot1x initialize interface GigabitEthernet 0/1**

Повторная аутентификация на всех интерфейсах Fast Ethernet:

- **dot1x initialize interface FastEthernet ***

37 ENABLE/Привилегированный режим

Переводит пользователя в привилегированный режим (режим **enable**), где доступны расширенные команды для настройки и управления устройством.

Синтаксис:

- **enable [<уровень_привилегий>]**

- **<уровень_привилегий>:** уровень привилегий **(от 0 до 15).** По умолчанию — уровень **15** (максимальные привилегии).

Примеры использования:

Переход в привилегированный режим (уровень 15):

- **enable**

Переход на уровень привилегий 7:

- **enable 7**

38 ERPS/Защита кольцевых топологий

Управление протоколом **ERPS (Ethernet Ring Protection Switching)**, используемым для защиты кольцевых топологий **Ethernet.**

Синтаксис:

erps <группа> command <опция> <порт>

- **<группа>:** номер группы ERPS (от 1 до 64).

- **command:** выполнение административных команд.

- **<опция>:** тип команды **(clear, force, manual)**.

- **<порт>:** порт ERPS (port0, port1).

Выбор группы ERPS

erps ?

1-64: номер группы ERPS.

Выбор команды

erps 1 command ?

clear: Очистка состояния порта.

force: Принудительное изменение состояния порта.

manual: Ручное изменение состояния порта.

Выбор порта

erps 1 command clear ?

port0: Порт 0.

port1: Порт 1.

39 EXIT/Выход из режима

Выход из текущего режима **EXEC.**

```
exit
```

40 FIRMWARE/Управление прошивкой коммутатора

Управление прошивкой коммутатора, включая обновление и переключение между активной и резервной версиями.

Синтаксис

firmware <опция> [<url_file>]

- **<опция>:** Действие с прошивкой (swap, upgrade).
- **<url_file>:** Путь к файлу прошивки (для команды upgrade).

Выбор опций через ?

- **firmware ?**
- **swap:** Переключение между активной и резервной прошивкой.
- **upgrade:** Обновление прошивки.

Формат url_file для upgrade:

firmware upgrade ?

Синтаксис: **<protocol>://[<username>[:<password>]@]<host>[:<port>][/<path>]/<file_name>.**

Допустимые символы: A-Za-z, 0-9, ., -, _.

Максимальная длина: 63 символа.

Запрещено: Начинать с (-) или использовать только (.).

Пример использования:

Переключение на резервную прошивку:

firmware swap

Обновление прошивки с TFTP-сервера:

firmware upgrade tftp://192.168.1.100/firmware.bin

41 HELP/Справка

```
help
```

Отображает справку по использованию команд.

42 Настройка и управление IP

Основная команда для настройки и управления IP-протоколом (IPv4). Включает подкоманды для работы с DHCP, маршрутизацией, интерфейсами и другими функциями.

Основные опции

ip dhcp: управление DHCP (Dynamic Host Configuration Protocol).

retry: перезапуск DHCP-запроса на указанном интерфейсе.

Синтаксис

ip dhcp retry interface vlan <vlan_id>

<vlan_id>: Идентификатор **VLAN**, на котором нужно перезапустить **DHCP-**запрос.

Пример использования:

Перезапуск **DHCP-запроса на VLAN 10**:

ip dhcp retry interface vlan 10

43 Настройка и управления IPv6

Основная команда для настройки и управления **IPv6.**

Основные опции:

- **ipv6 dhcp-client:** Управление DHCPv6-клиентом.
- **restart:** Перезапуск службы DHCPv6-клиента.

Синтаксис

ipv6 dhcp-client restart interface vlan <vlan_list>

<vlan_list>: список VLAN, на которых нужно перезапустить DHCPv6-клиент.

Пример использования:

Перезапуск DHCPv6-клиента на VLAN 10:

ipv6 dhcp-client restart interface vlan 10

44 LINK-OAM/Управление функциями

Управление функциями **OAM (Operation, Administration, and Maintenance)** для диагностики и мониторинга Ethernet-соединений.

Основные опции

remote-loopback: управление удаленным loopback-тестированием.

- **start:** запуск loopback-теста.
- **stop:** остановка loopback-теста.

Синтаксис:

link-oam remote-loopback <start|stop> interface <тип_интерфейса>

- **<тип_интерфейса>:** Тип интерфейса **(FastEthernet, GigabitEthernet, 10GigabitEthernet).**

Пример использования:

Запуск loopback-теста на интерфейсе GigabitEthernet 0/1:

- **link-oam remote-loopback start interface GigabitEthernet 0/1**

Остановка loopback-теста на интерфейсе GigabitEthernet 0/1:

- **link-oam remote-loopback stop interface GigabitEthernet 0/1**

45 Logout/Завершение текущей сессии

Завершает текущую сессию и выходит из режима **EXEC.**

```
logout
```

46 MORE/Просмотр файла

Просмотр содержимого файла постранично. Используется для чтения текстовых файлов (например, конфигураций или логов).

Синтаксис:

more <url_file>

<url_file>: Путь к файлу. Формат:

flash: <имя_файла> (например, flash: config.txt).

fttp://<сервер>/<путь> (например, tftp://192.168.1.100/log.txt).

Правила именования файлов

Допустимые символы: A-Za-z, 0-9, ., -, _.

Максимальная длина: 63 символа.

Запрещено:

Начинать имя файла с (-).

Использовать только (.) в имени файла.

47 NO/Отмена или удаление настроек

Используется для отмены или удаления настроек. Применяется к различным командам и параметрам.

Основные опции:

no alarm suppress: отменяет подавление указанной тревоги.

Синтаксис:

no alarm suppress <alarm_name>

Пример:

no alarm suppress overheat-alarm

no debug: отключает функции отладки.

Опции:

- **gdbserver:** отключает отладку через GDB-сервер.
- **interrupt:** отключает отладку прерываний.

- **ipv6:** отключает отладку IPv6.
- **trace:** отключает трассировку.

Пример:

no debug gdbserver

no terminal: отменяет настройки командной строки.

Пример:

no terminal length

47.1 Отмена настрое командной строки

Отменяет настройки командной строки.

Основные опции:

- **no terminal editing**

Отключает редактирование командной строки (например, использование стрелок для перемещения курсора).

Пример:

- **no terminal editing**
- **no terminal exec-timeout:**

Отключает таймаут **EXEC-сессии** (время бездействия, после которого сессия завершается).

Пример:

no terminal exec-timeout

- **no terminal history**

Отключает сохранение истории команд.

Пример:

no terminal history

- **no terminal length:**

Отменяет ограничение на количество строк, отображаемых на экране.

Пример:

no terminal length

- **no terminal width**

Отменяет заданную ширину терминала (количество символов в строке).

Пример:

no terminal width

Фильтрация вывода:

Используйте | для фильтрации вывода:

- **begin <строка>:** показывает вывод, начиная с указанной строки.

- **exclude <строка>:** исключает строки, содержащие указанный текст.

- **include <строка>:** показывает только строки, содержащие указанный текст.

48 PING/Проверка доступности узла

ping ip — проверка доступности узла по IPv4.

ping ipv6 — проверка доступности узла по IPv6.

Используйте IP-адрес или доменное имя в качестве цели.

49 PLATFORM/ Управление платформой устройства

Управление платформой устройства, включая отладочные функции и другие низкоуровневые настройки.

Основные опции

platform debug: Управление разрешением или запретом выполнения отладочных команд.

- **allow:** разрешает выполнение отладочных команд.
- **deny:** запрещает выполнение отладочных команд.

Примеры использования:

Разрешение отладочных команд:

platform debug allow

Запрет отладочных команд:

platform debug deny

50 RELOAD/Перезагрузка

Перезагрузка устройства. Поддерживает различные режимы перезагрузки.

Основные опции

- **reload cold:** Полная перезагрузка устройства (холодная перезагрузка).

Пример

- **reload cold**
- **reload defaults:** Перезагрузка с восстановлением настроек по умолчанию.

Опции:

- **keep-ip: сохраняет IP-адрес на VLAN 1**.

Пример:

reload defaults keep-ip

Примеры использования

Полная перезагрузка:

- **reload cold**

Перезагрузка с восстановлением настроек по умолчанию:

- **reload defaults**

Перезагрузка с сохранением IP-адреса на VLAN 1:

- **reload defaults keep-ip**

51 SEND/Отправка сообщений

Отправка сообщений на терминальные линии **(TTY, консоль, VTY)**. Используется для уведомления других пользователей, подключенных к устройству.

- **send *:** Отправка сообщения на все терминальные линии.

Синтаксис

- **send * <сообщение>**

Пример

send * "Система будет перезагружена через 5 минут."

- **send <0~16>:** отправка сообщения на указанные терминальные линии.

Синтаксис

send <номер_линии> <сообщение>

Пример:

send 1 "Проверьте подключение."

send console: Отправка сообщения на консольную линию.

Синтаксис:

send console <сообщение>

Пример:

send console "Внимание: обновление конфигурации."

send vty: Отправка сообщения на виртуальные терминальные линии (VTY).

Синтаксис:

send vty <номер_линии> <сообщение>

Пример:

send vty 0 "Сессия будет завершена через 10 минут."

52 SHOW/Показ дополнительных команд

Показывает дополнительные команды.

С AAA (Authentication, Authorization and Accounting methods) до WEB (настройка WEB-привелегий).

Данные команды описаны в первой части документа, до спункта 29 – Alarm.

53 TERMINAL/Настройка параметров терминала

Команда terminal используется для настройки параметров терминала.

Опции команды terminal:

- editing
  Включает/отключает редактирование командной строки.

Синтаксис:

- terminal editing

Позволяет использовать клавиши (стрелки, Backspace) для редактирования команд.

- exec-timeout
  Устанавливает тайм-аут неактивной сессии (в секундах).

Синтаксис:

terminal exec-timeout <секунды>

Пример:

terminal exec-timeout 10

Сессия завершится, если пользователь неактивен более 10 секунд.

- help
  Описание интерактивной справки.

Синтаксис:

terminal help

Показывает информацию о доступных командах и их использовании.

- history
  Управляет размером истории команд.

Синтаксис:

- terminal history size <количество команд>

Пример:

- terminal history size 20

Сохраняет последние 20 команд для быстрого доступа.

- length
  Устанавливает количество строк на экране.

Синтаксис:

terminal length <количество_строк>

Пример:

terminal length 24

Ограничивает вывод 24 строками перед постраничным отображением.

- width: Устанавливает ширину терминала (количество символов в строке).

Синтаксис:

terminal width <количество_символов>

Пример:

terminal width 80

Устанавливает ширину терминала в 80 символов.

54 VERIPHY/Проверка состояния

Используется для проверки состояния и настроек интерфейсов на коммутаторе. Она позволяет получить информацию о конкретных портах или всех портах сразу. Команда поддерживает различные типы интерфейсов, такие как **Fast Ethernet, Gigabit Ethernet и 10 Gigabit Ethernet.**

Опции команды veriphy:

- interface
  Позволяет выбрать интерфейс для проверки.

Синтаксис:

veriphy interface <тип_интерфейса> <список_портов>

Параметры:

- — проверка всех портов на всех коммутаторах.
- FastEthernet — проверка портов Fast Ethernet.
- GigabitEthernet — проверка портов Gigabit Ethernet.
- 10GigabitEthernet — проверка портов 10 Gigabit Ethernet.

<список_портов> — список портов для проверки (например, 1/1, 1/1-32).

Примеры использования:

1. Проверка всех портов на всех коммутаторах:

veriphy interface *

2. Проверка конкретного порта Fast Ethernet:

veriphy interface FastEthernet 1/1

3. Проверка диапазона портов Gigabit Ethernet:

veriphy interface GigabitEthernet 1/1-32

4. Проверка всех портов 10 Gigabit Ethernet:

veriphy interface 10GigabitEthernet 1/1-4