



Зелакс ММ

Справочник команд
Настройка сетевых и системных параметров
ММ-22х, ММ-52х

Система сертификации в области связи
Сертификат соответствия
Регистрационный номер: ОС-1-СПД-0018

© 1998 — 2016 Zelax. Все права защищены.

Редакция 06 от 14.07.2016 г.
ПО 1.23.1.2 (ММ-22х, ММ-52х)

Россия, 124681 Москва, г. Зеленоград, ул. Заводская, дом 1Б, строение 2
Телефон: +7 (495) 748-71-78 (многоканальный) • <http://www.zelax.ru>
Отдел технической поддержки: tech@zelax.ru • Отдел продаж: sales@zelax.ru

Оглавление

1	Интерфейс и режимы настройки сетевых и системных параметров.....	3
1.1	Синтаксис команд	3
1.2	Контекстная справка	4
1.3	Сообщения об ошибках	5
2	Работа с файловой системой	6
3	Системные параметры	21
4	Сетевые настройки и утилиты	34
5	Режим моста (bridge), протокол Spanning Tree (STP)	42
6	Виртуальные локальные сети (VLAN).....	52
7	Режим объединения каналов (bond)	55
8	Пример настройки объединения каналов (bond).....	58
9	Режим Multilink Point-to-Point (MLP).....	61
10	Пример настройки Multilink Point-to-Point.....	65
11	Статическая маршрутизация	68
12	Настройка синхронизации (NTP)	69
13	Протоколирование событий (Syslog).....	70
13.1	Фильтрация данных	70
13.2	Фильтр кадров Ethernet (ebtables)	70
13.3	Фильтр пакетов IPv4 и NAT (iptables)	72
14	Настройка протокола DHCP	76
15	Настройка протокола SNMP	79
16	Настройка качества обслуживания (QoS).....	80
16.1.1	Элементы управления исходящим трафиком.....	80
16.1.2	Механизмы управления исходящим трафиком.....	80
16.2	Бесклассовые дисциплины обработки очереди.....	82
16.2.1	Дисциплина pfifo_fast	82
16.2.2	Дисциплина Token Bucket Filter (TBF)	84
16.2.3	Дисциплина Stochastic Fairness Queueing (SFQ).....	85
16.2.4	Дисциплина Traffic Equalizer (TEQL).....	86
16.3	Классовые дисциплины обработки очереди	88
16.3.1	Создание классов	88
16.3.2	Задание классовых дисциплин обработки очереди	90
16.3.2.1	Дисциплина PRIO	90
16.3.2.2	Дисциплина Class Based Queueing (CBQ)	91
16.3.2.3	Дисциплина Hierarchical Token Bucket (HTB).....	95
16.3.3	Создание фильтров	98
16.3.3.1	Общее описание фильтров.....	99
16.3.3.2	Фильтр fw	99
16.3.3.3	Фильтр U32	101
16.4	Ограничение входящего трафика (policing).....	104
17	Настройка протокола OSPF	107
18	Пример настройки протокола OSPF	114
19	Настройка протокола RIP	116
20	Настройка протокола VRRP	120
21	Пример настройки протокола VRRP	125

1 Интерфейс и режимы настройки сетевых и системных параметров

1.1 Синтаксис команд

Синтаксис команд, вводимых в командной строке:

команда {параметр | параметр} [параметр | параметр]

где:

Команда — строго заданная последовательность символов, определяющая дальнейшие параметры.

Параметр — ключевое слово, IP-адрес, маска сети, IP-адрес, MAC-адрес, число, слово, строка.

Команда и параметры отделяются друг от друга пробелами.

При описании синтаксиса команд используются следующие обозначения:

- в фигурных скобках {} указываются обязательные параметры;
- в квадратных скобках [] указываются необязательные параметры;
- символ "|" обозначает логическое "или" — выбор между различными параметрами;
- ключевые слова выделяются жирным шрифтом.

Типы параметров команд:

Ключевое слово — слово несущее определенную смысловую нагрузку, например, название протокола, имя интерфейса и т. д.

IP-адрес — A.B.C.D — задается в виде четырех десятичных чисел, разделенных точками.

Маска сети — A.B.C.D — задается в виде четырех десятичных чисел, разделенных точками.

MAC-адрес — A1:A2:A3:A4:A5:A6 — задается в виде шести групп чисел, разделенных символом ":". Каждая группа состоит из двух шестнадцатеричных чисел.

Число <Num1 ... Num2> — задается десятичным числом, которое больше или равно Num1 и меньше или равно Num2.

Слово — WORD — задается в виде набора символов без пробелов.

Строка — LINE — задается в виде набора символов. Допустимо использование символа "Пробел".

Для исполнения набранной команды необходимо нажать клавишу "Enter".

Для получения контекстной справки используется символ "?" и параметр "--help".

При нажатии клавиши табуляции "Tab" происходит автоматическое доопределение сокращенных названий команд и некоторых типов параметров до их полного вида, или, в случае, когда несколько команд начинаются с одинаковых символов, до их общей части.

Последние десять введенных команд хранятся в буфере. Чтобы воспользоваться ранее введенной командой, необходимо нажать клавишу "↑" (вверх) или "↓" (вниз).

1.2 Контекстная справка

Для получения контекстной справки используется символ “?”.

При вводе символа “?” выводится список команд, доступных в данном режиме.

Пример. Использование контекстной справки для получения списка команд.

```
router#?
@                head                netstat          switchstat
arp              hostname            nslookup         syslogd
arping           hwclock            ntp              tail
ash              id                  ntpdate         tar
brctl            ifconfig           ospfd            tc
busybox          ip                  passwd           tcpstat
cat              ipaddr             passwd-user      telnet
chmod            iplink             ping             tftp
cp               iprule             pkill            timezone
crontab          iptables           preset-config    top
date             iptables-restore   ps               traceroute
df               iptables-save      pwd              udhcpd
dumpleases       iptunnel           reboot           udhcpd
ebtables         keepalived         rm               update-muxdcfg
echo             kill                route            upgrade-software
env              less                sed              uptime
ethtool          login               sethctl          vconfig
ftpget           ls                  sh               version
ftpput           more                shell             vtysh
grep             mv                  snmpd            wget
gzip             nano                stty             zebra
```

При вводе параметра “--help” через пробел после команды выводится список параметров данной команды.

Пример. Использование контекстной справки для получения списка параметров команды tftp.

```
router#tftp --help
BusyBox v1.8.2 (2009-12-16 21:32:21 MSK) multi-call binary

Usage: tftp [OPTION]... HOST [PORT]

Transfer a file from/to tftp server using "octet" mode

Options:
  -l FILE Local FILE
  -r FILE Remote FILE
  -g      Get file
  -p      Put file
  -b SIZE Transfer blocks of SIZE octets
```

При вводе символа “?” без пробела после частично введенной команды выводится список команд, начинающихся с данных символов.

Пример. Использование контекстной справки для получения списка команд, начинающихся с символов “ar”.

```
router#ar?
arp      arping
```

1.3 Сообщения об ошибках

Ниже приведены сообщения об ошибках, которые могут выводиться во время работы с командной строкой.

Сообщение об ошибке	Описание ошибки	Рекомендуемые действия
-sh: <команда>: not found	Неизвестная команда. Неполная команда или неполный синтаксис. Команда не была идентифицирована. Введена ошибочная команда.	С помощью контекстной справки "?" следует проверить корректность вводимой команды.
-sh: cd: can't cd to /<директория>	Отсутствуют права на вход в директорию.	Выполнить команду, используя учетную запись с соответствующими правами.
-sh: ./<имя файла>: Permission denied	Отсутствуют права на доступ к файлу.	Выполнить команду, используя учетную запись с соответствующими правами.
<команда>: SIOCSIFFLAGS: Permission denied	Отсутствуют права на конфигурирование.	Выполнить команду, используя учетную запись с соответствующими правами.

2 Работа с файловой системой

cat

Назначение:

Считывает последовательно файлы и выдает их содержимое в стандартный выходной поток.

Синтаксис:

`cat [-u] {file, ...}`

Параметр	Описание
-u	Не буферизовать выдаваемый результат
File	Имя файла, содержимое которого будет выведено в выходной поток

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **cat** позволяет выполнить вывод содержимого одного или нескольких файлов в стандартный выходной поток. А также объединить несколько файлов в один.

Пример:

Вывод содержимого файла muxd.conf на терминал:

```
router#cat muxd.conf
! Timestamp: Sat Jan  1 00:12:00 UTC 2000
! -----
!
! -----
! Hash: dfc4efaca24e24ffe6b9aef250312c51
```

chmod

Назначение:

Изменение прав доступа к объекту файловой системы.

Синтаксис:

`chmod [-R] mode [mode, ...] {file, ...}`

Параметр	Описание
-R	Рекурсивный вызов команды
mode	Права доступа
file	Имя файла или директории

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **chmod** позволяет изменить права доступа к директории, одному или нескольким файлам.

Существует три уровня доступа к объекту – чтение, запись, исполнение. Права определяются для трёх типов пользователей - владельца файла, группы, в которую он входит и для остальных пользователей. Параметр **mode** может быть записан как в числовом так и в символьном виде (см. пример).

Использование параметра **-R** позволяет определить права доступа к директории и всем её файлам и поддиректориям.

Пример:

Предоставление полных прав доступа для пользователя владельца, его группы и права на чтение и запись остальным пользователям.

Запись в числовом формате:

```
router#chmod 776 resolv.conf
```

Запись в символьном формате:

```
router#chmod +rwxrwxrw- resolv.conf
```

ср

Назначение:

Копирование файлов и директорий.

Синтаксис:

ср [-a] [-d,-P] [-H,-L] [-p] [-f] [-i] [-R,-r] [-l,-s] {source} {dest}

Параметр	Описание
-a	Сохраняется структура и атрибуты исходных файлов при копировании, но не сохраняется структура директорий
-d,-P	Символьные ссылки копируются как ссылки, а не как файлы
-H,-L	Разыменовываются символьные ссылки, копируются те объекты, на которые они указывают
-p	Сохраняются оригинальные атрибуты файла
-f	Производится перезапись существующих файлов
-i	Запрос, перед перезаписью существующих файлов
-R,-r	Рекурсивное копирование директорий
-l,-s	Создаются жесткие ссылки (symlinks) вместо копирования обычных файлов
Source	Имя объекта источника копирования
Dest	Имя объекта назначения копирования

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ср** позволяет выполнить копирование файлов и директорий, а также создать или скопировать жесткие ссылки.

Пример:

```
router#cp 1.txt 2.txt
```

crontab

Назначение:

Планировщик заданий.

Синтаксис:

crontab [-c dir] {file | -} [-u | -l | -e | -d user]

Параметр	Описание
-c	Указание директории файла crontab
dir	Директория файла crontab
file	Имя файла crontab
-u	Указание пользователя, чей файл crontab подвергается обработке
-l	Вывод на дисплей текущего содержимого файла crontab
-e	Редактирование файла crontab
-d	Удаление файла crontab
user	Имя пользователя, чей файл crontab подвергается обработке

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **crontab** позволяет управлять запланированными заданиями. Для каждого пользователя ведется свой файл crontab.

Пример:

Редактирование файла crontab для текущего пользователя:

```
router#crontab -e
```

df

Назначение:

Отображает статистику использования файловой системы.

Синтаксис:

df [-h | -m | -k] [filesystem, ...]

Параметр	Описание
-h	К каждому значению добавляется его размерность
-m	Устанавливает размер блока в двоичный мегабайт (1024 кбайт)
-k	Устанавливает размер блока равный 1024 байт
filesystem	Файловая система

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **df** позволяет отобразить статистику использования файловой системы. Введенная без параметров команда отображает статистику для всех файловых систем.

Пример:

```
router#df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/mtdblock2  7.6M      4.6M      3.0M    60% /
tmpfs           14.5M     16.0k     14.5M    0% /tmp
```

echo

Назначение:

Вывод одной строки текста на стандартный выходной поток.

Синтаксис:

echo

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **echo** позволяет вывести одну строку текста на стандартный выходной поток.

Пример:

```
router#echo ---Test string---  
---Test string---
```

env

Назначение:

Работа с переменными окружения.

Синтаксис:

env [-i] [-u] [-] [name=value] [command]

Параметр	Описание
-i	Выполнить команду без переменных окружения
-u	Удалить переменную из текущего окружения
name	Имя переменной
value	Значение переменной
command	Команда

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **env** позволяет выполнить команду с указанием определенных переменных окружения. Введенная без параметров – выводит текущие переменные окружения.

Пример:

```
router#env  
USER=admin  
OLDPWD=/home/admin  
HOME=/home/admin  
PS1=router#  
LOGNAME=admin  
TERM=vt102  
PATH=/usr/sbin:/bin  
SHELL=/bin/sh  
MUX_CMD_HISTORY=30  
PWD=/etc/config  
EDITOR=/usr/sbin/nano
```

ftpget

Назначение:

Загрузка файла с удаленного устройства по протоколу FTP.

Синтаксис:

ftpget [-c] [-v] [-u] [-p] [-P] {remote-host} {local-file} {remote-file}

Параметр	Описание
-c	Продолжить предыдущую загрузку
-v	Выводит подробную информацию
-u	Имя пользователя для регистрации на FTP-сервере
-p	Пароль для регистрации на FTP-сервере
-P, --port	Номер порта
remote-host	Имя или IP-адрес FTP-сервера
local-file	Имя файла, под которым он будет сохранен на устройстве
remote-file	Имя файла, под которым он хранится на FTP-сервере

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ftpget** позволяет загрузить файл с удаленного устройства по протоколу FTP.

Пример:

Загрузка файла muxd.conf с сервера с IP-адресом 172.16.3.2 с указанием имени пользователя и пароля, а также выводом подробной статистики:

```
router#ftpget -v -u Zelax -p 123456 172.16.3.2 /tmp/muxd_load.conf muxd.conf
Connecting to 172.16.3.2 (172.16.3.2:21)
ftpget: cmd (null) (null)
ftpget: cmd USER Zelax
ftpget: cmd PASS 123456
ftpget: cmd TYPE I (null)
ftpget: cmd PASV (null)
ftpget: cmd SIZE muxd.conf
ftpget: cmd RETR muxd.conf
ftpget: cmd (null) (null)
ftpget: cmd QUIT (null)
```

ftpput**Назначение:**

Загрузка файла на удаленное устройство по протоколу FTP.

Синтаксис:

ftpget [-v] [-u] [-p] [-P] {remote-host} {local-file} {remote-file}

Параметр	Описание
-v	Выводит подробную информацию
-u	Имя пользователя для регистрации на FTP-сервере
-p	Пароль для регистрации на FTP-сервере
-P, --port	Номер порта
remote-host	Имя или IP-адрес FTP-сервера
local-file	Имя файла, под которым он хранится на устройстве
remote-file	Имя файла, под которым он будет сохранен на FTP-сервере

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ftpput** позволяет загрузить файл на удаленное устройство по протоколу FTP.

Пример:

Загрузка файла muxd.conf на сервер с IP-адресом 172.16.3.2 с указанием имени пользователя и пароля, а также выводом подробной статистики:

```
router#ftpput -v -u Zelax -p 123456 172.16.3.2 muxd.conf muxd.conf
Connecting to 172.16.3.2 (172.16.3.2:21)
ftpput: cmd (null) (null)
ftpput: cmd USER Zelax
ftpput: cmd PASS 123456
ftpput: cmd TYPE I (null)
ftpput: cmd PASV (null)
ftpput: cmd ALLO 171 (null)
ftpput: cmd STOR muxd.conf
ftpput: cmd (null) (null)
ftpput: cmd QUIT (null)
```

grep

Назначение:

Поиск образца в файлах.

Синтаксис:

```
grep [-HhriLnqvsoeFABC] [PATTERN] [filename1, filename2, ...]
```

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **grep** выполняет поиск образца в текстовых файлах и выдает все строки, содержащие этот образец.

Пример:

Поиск в файле muxd.conf строки содержащей Jan:

```
router#grep Jan muxd.conf
! Timestamp: Sat Jan 1 00:12:00 UTC 2000
```

gzip

Назначение:

Сжатие или распаковывание файлов.

Синтаксис:

```
gzip [-c] [-d] [-s] [filename, ...]
```

Параметр	Описание
-c	Вывод результата архивирования в стандартный поток вывода

-d	Распаковка
-s	Проводить сжатие или распаковку, даже если на файл есть ссылки либо такой архив уже существует, а также, если сжатые данные читаются или пишутся на терминал
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **gzip** позволяет осуществить сжатие или распаковывание файлов.

Пример:

Создание архива файла muxd_load.conf. В результате оригинальный файл будет удален и создан архив с именем muxd_load.conf.gz:

```
router#gzip muxd_load.conf
```

head

Назначение:

Частичный вывод содержимого файла в стандартный поток вывода.

Синтаксис:

```
head [-c] [-n] [-q] [-v] [filename]...
```

Параметр	Описание
-c, --bytes=[-]N	Вывод первых N байт содержимого файла. Если указан символ [-] вывод последних N байт
-n	Вывод первых N строк содержимого файла. Если указан символ [-] вывод последних N строк
-q	Не печатать заголовки, содержащие имена файлов
-v	Печатать заголовки, содержащие имена файлов
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **head** позволяет осуществить частичный просмотр содержимого файла. Для того, чтобы просмотреть содержимое нескольких файлов необходимо указать их через пробел.

Пример:

Вывод первых девяти строк из файлов muxd.conf netconfig.sh:

```
router#head -n 90 -v muxd.conf netconfig.sh
==> muxd.conf <==
! Timestamp: Sat Jan  1 01:28:11 UTC 2000
! -----
!
interface HDLC 0 E1 1/0
!
! -----
```

```

==> netconfig.sh <==
## device name
hostname router

## interface Ethernet 0 configuration
ifconfig eth0 up

## interface HDLC 0 configuration
sethdlc hdlc0 hdlc-eth
ifconfig hdlc0 up

## bridge configuration
brctl addbr br1
brctl addif br1 eth0
brctl addif br1 hdlc0

## interface br1 configuration
ifconfig br1 up 192.168.1.1

```

kill

Назначение:

Посылка определённого сигнала процессу по имени и другим атрибутам.

Синтаксис:

kill [-l {signal | number}]

kill [-signal] PID

Параметр	Описание
-l	Вывод имен сигналов, соответствующих номеру, и наоборот
PID	Совпадение введённой команды

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда kill посылает сигнал процессу или выводит список допустимых сигналов.

Пример:

Посыл сигнала завершения процессу syslogd:

```
router#kill 236
```

less

Назначение:

Просмотр содержимого файла или нескольких файлов.

Синтаксис:

less [-E] [-M | -m] [-N]~?] [filename1, filename2, ...]

Параметр	Описание
-E	Выход по достижении конца файла
-M -m	Отображения строки состояния, которая содержит номер строки и процент отображенного содержимого файла

-N	Предварять каждую строку номером
--	Подавить вывод ~ по достижении окончания файла
filename1, filename2	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **less** позволяет отобразить содержимое одного или нескольких файлов. Данная команда может применяться для просмотра больших файлов, т.к. не требует прочтения всего содержимого файла, перед его открытием.

Пример:

```
router#less -E -- muxd.conf
! Timestamp: Sat Jan 1 00:12:00 UTC 2000
! -----
!
! -----
! Hash: dfc4efaca24e24ffe6b9aef250312c51
```

ls

Назначение:

Вывод содержимого каталога.

Синтаксис:

ls [-1AacCdeFilnP LRrSsTtuvwxXhk] [filename1, filename2, ...]

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ls** позволяет отобразить содержимое каталога.

Пример:

```
router#ls
muxd.conf      netconfig.sh  syslog.conf
```

more

Назначение:

Вывод содержимого файла по частям.

Синтаксис:

more {filename}

Параметр	Описание
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **more** позволяет отобразить содержимое файла постранично. Прежде чем продолжить вывод содержимого файла команда ожидает реакции пользователя.

Пример:

```

router#more muxd.conf
! Timestamp: Sat Jan  1 00:12:00 UTC 2000
! -----
!
! -----
! Hash: dfc4efaca24e24ffe6b9aef250312c51
router#

```

mv**Назначение:**

Перемещение и переименование файлов.

Синтаксис:

mv [-f] [-i] {source} {dest}

Параметр	Описание
-f	Перезапись без запроса подтверждения
-i	запроса подтверждения перед перезаписью
source	Исходное имя
dest	Результирующее имя

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **mv** позволяет переименовать или переместить файлы. Если последний аргумент является именем существующего каталога, то **mv** перемещает все остальные файлы в этот каталог. В противном случае, если задано только два файла, то имя первого файла будет изменено на имя второго. Если последний аргумент не является каталогом и задано более чем два файла, то будет выдано сообщение об ошибке.

Пример:

Переименование файла:

```

router#mv muxd1.conf muxd2.conf

```

nano**Назначение:**

Текстовый редактор.

Синтаксис:

nano {filename}

Параметр	Описание
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **nano** позволяет запустить текстовый редактор. Помимо функции редактирования, **nano** позволяет осуществлять поиск и замену, переход к определенной строке файла и многие другие необходимые операции.

Пример:

```
router#nano muxd2.conf
```

```
GNU nano 2.0.9                               File: muxd2.conf

! Timestamp: Sat Jan  1 00:12:00 UTC 2000
! -----
!
! -----
! Hash: dfc4efaca24e24ffe6b9aef250312c51

[ Read 5 lines ]
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text
^C Cur Pos      ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut
^X Exit Text    ^T To Spell
```

pkill

Назначение:

Поиск процесса и посылка определённого сигнала процессу по имени и другим атрибутам.

Синтаксис:

```
pkill [-l] | [-f] | [-n] | [-o] | [-v] | [-x] [-signal] pattern
```

Параметр	Описание
-l	Список всех сигналов
-f	Совпадение введённой команды
-n	Сигнал только новым процессам
-o	Сигнал только старым процессам
-v	Не совпадает
-x	Совпадение по имени

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **pkill** производит поиск процесса и посылку определённого сигнала процессу по имени и другим атрибутам.

Пример:

Вывод списка всех доступных сигналов:

```
router#pkill -l
HUP
```

INT
QUIT
...
PROF
WINCH
POLL
PWR
SYS

ps

Назначение:

Отображение запущенных процессов.

Синтаксис:

ps

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ps** позволяет отобразить список запущенных процессов.

Пример:

```
router#ps
  PID  Uid          VSZ  Stat  Command
   1  root          1116  S    init
   2  root           SW<  [kthreadd]
   3  root           SW<  [ksoftirqd/0]
   4  root           SW<  [watchdog/0]
```

pwd

Назначение:

Отображение текущей директории.

Синтаксис:

pwd

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **pwd** позволяет отобразить текущую директорию.

Пример:

```
router#pwd
/etc/config
```

rm

Назначение:

Удаление файлов или каталогов.

Синтаксис:

`rm [-i] [-f] [-r | -R] {filename}`

Параметр	Описание
-i	Запрос на подтверждение удаления
-f	Не запрашивать подтверждения удаления
-r, -R	Рекурсивное удаление дерева каталогов
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда `rm` позволяет удалить файл или каталог.

Пример:

```
router#rm -i muxd_test.conf
rm: remove 'muxd_test.conf'? y
```

tail**Назначение:**

Вывод содержимого последних строк файла.

Синтаксис:

`tail [--retry] [-c] [-f] [-F] [-n] [-n] [-q] [-s] [filename]...`

Параметр	Описание
--retry	Продолжать попытки открытия файла, даже если он недоступен
-c, --bytes=N	Вывод последних N байт содержимого файла
-f, --follow	Выводить поступающие данные по мере роста файла
-F	Тоже, что и --follow
-n, --lines=N	Вывод последних N строк, а не последние 10
-q, --quiet, --silent	Не печатать заголовки, содержащие имена файлов
-s, --sleep-interval=S	С ключом -f, бездействовать примерно S секунд (по умолчанию 1.0) перед итерацией
-v	Печатать заголовки, содержащие имена файлов
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда `tail` позволяет осуществить просмотр указанного количества последних строк файла. Для того, чтобы просмотреть содержимое нескольких файлов необходимо указать их через пробел.

Пример:

```
router#tail --lines=4 muxd.conf
! -----
```

```
!  
!  
! Hash: dcc46d4dcbdb83ca8a769674f49c5a5c1
```

tar

Назначение:

Архиватор.

Синтаксис:

```
tar {-c | -x | -t} [v] [O] {-z | -j | -a | -Z} [-X FILE] [-f TARFILE] [-C DIR] [filename]...
```

Параметр	Описание
-c	Создать новый архив
-z	Фильтрация архива с помощью gzip
-j	Фильтрация архива с помощью bzip2
-a	Фильтрация архива с помощью lzma
-Z	Фильтрация архива с помощью compress
-x	Извлечь файлы из архива
-t	Вывести содержание архива
-v	Предоставление списка обрабатываемых файлов
-O	Направление извлекаемых из архива файлов в стандартный вывод
-X	Исключить из обработки файлы, перечисленные в файле FILE
-f	Использовать архивный файл или устройство
-C	Изменить каталог на указанный
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **tar** позволяет давить или извлечь файлы и каталоги из архива.

Пример:

```
router#tar -cz -f muxd_archive muxd_test.conf
```

tftp

Назначение:

Передача файла по протоколу TFTP.

Синтаксис:

```
tftp {-l | -r filename} {-g | -p} [-b SIZE]
```

Параметр	Описание
-l	Имя локального файла
-r	Имя удаленного файла
-g	Загрузка файла
-p	Выгрузка файла
-b	Передача блока данных размером SIZE октетов
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **tftp** позволяет осуществить передачу файлов по протоколу TFTP.

Пример:

Загрузка файла с TFTP-сервера с IP-адресом 192.168.0.105:

```
router#tftp -l zelaxmm.1.1.0.1.tgz -r zelaxmm.1.1.0.1.tgz -g 192.168.0.105
```

top

Назначение:

Вывод статистику активности запущенных процессов.

Синтаксис:

top [-b] [-n count] [-d seconds]

Параметр	Описание
-b	Вывод статистики на дисплей с сохранением предыдущего вывода
-n	Выводит статистику n раз
-d	Интервал вывода статистики

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **top** позволяет в реальном времени отобразить активность запущенных процессов. Выход осуществляется с помощью сочетания клавиш "Ctrl+C".

Пример:

```
router#rm -i muxd_test.conf
rm: remove 'muxd_test.conf'? y
```

wget

Назначение:

Загрузка файлов по протоколам HTTP и FTP.

Синтаксис:

wget [-c|--continue] [-s|--spider] [-q|--quiet] [-O|--output-document file] [--header 'header: value'] [-Y|--proxy on/off] [-P DIR] [-U|--user-agent agent] url

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Утилита **wget** позволяет загрузить файлы в устройство используя протоколы HTTP и FTP.

3 Системные параметры

@

Назначение:

Команда позволяет выполнить команду mux shell находясь в Linux shell и наоборот выполнить команду Linux shell находясь в mux shell.

Синтаксис:

@ {command}

Параметр	Описание
command	Команда другого режима конфигурирования

Режим конфигурации:

Команда доступна в linux shell и mux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда @ позволяет выполнять команду доступную в другом режиме конфигурирования. При использовании команды @ недоступно использование автодополнения команд и контекстной справки, но возможно использование сокращённых команд.

Пример:

Выполнение команды привилегированного режима из режима linux shell:

```
router#@ show system mims
```

```
Module 0 is MIME-RE100T-4 (4 serial interface, 1 ethernet port)
  Controllers: IMUX 0/0..0/1, BACKUP 0/10..0/19
```

```
Module 1 is MIME-2xG703L (2 ports G.703/E1, sensitivity -12 dB)
  Controllers: E1 1/0..1/1
```

Выполнение команды режима Linux shell из режима конфигурирования контроллеров и кросс-коннектора:

```
router(shell-config)#@ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:08:0F:22:CA:FE
          inet addr:172.16.1.112  Bcast:172.16.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1600  Metric:1
          RX packets:36195 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17211 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10589606 (10.0 MiB)  TX bytes:1030286 (1006.1 KiB)
          Base address:0xe00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

date

Назначение:

Устанавливает и отображает дату и время.

Синтаксис:

date [-s] [MMDDhhmmYYYY]

Параметр	Описание
-s	Устанавливает дату и время
MM	Месяц, в числовом формате
DD	Число
hh	Час
mm	Минута
YYYY	Год

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **date** позволяет установить дату и время. Введенная без параметров команда отображает текущие значения даты и времени.

Пример:

Установка даты 10 октября 2009 года и времени 15:22:

```
router#date -s 102015222009
Tue Oct 20 15:22:00 UTC 2009
```

hostname

Назначение:

Вывод или установка имени устройства.

Синтаксис:

hostname [-s] [-i] [-d] [-f] [hostname | -F filename]

Параметр	Описание
-s	Вывод имени устройства без указания доменного имени
-i	IP-адрес устройства
-d	Имя домена
-f	Полное доменное имя устройства
-F	Использовать содержимое файла для задания имени устройства
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **hostname** позволяет задать или отобразить имя устройства. Параметры команды позволяют отобразить как IP-адрес устройства, так и полное доменное имя, а также задать имя устройства на основе содержимого файла.

Пример:

Установка имени устройства в router:

123#hostname router

hwclock

Назначение:

Устанавливает и отображает аппаратные часы устройства.

Синтаксис:

hwclock [-r|--show] [-s|--hctosys] [-w|--systohc] [-l|--localtime] [-u|--utc] [-f filename]

Параметр	Описание
-r, --show	Отображает текущее время и дату в соответствии с аппаратными часами
-s, --hctosys	Устанавливает системные часы в соответствии с аппаратными
-w, --systohc	Устанавливает аппаратные часы в соответствии с системными
-l, --localtime	Аппаратные часы соответствуют локальному времени
-u, --utc	Аппаратные часы соответствуют универсальному координированному времени (UTC)
-f	Устанавливает аппаратные часы на основе данных файла
filename	Имя файла

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **hwclock** позволяет установить аппаратные часы устройства. Данная команда позволяет установить соответствие между аппаратными и системными часами и задать способ ведения времени – локальное время или UTC.

Пример:

Устанавливает время аппаратных часов равным UTC на основании системного времени:

```
router#hwclock --systohc --utc
```

id

Назначение:

Вывод информации о пользователе.

Синтаксис:

id [-g] [-u] [-n] [-r] [username]

Параметр	Описание
-g	Вывод только идентификатора группы, в которую входит пользователь
-u	Вывод только идентификатора пользователя
-n	Вывод действующих имен пользователя
-r	Вывод подлинных числовых идентификаторов пользователя
username	Имя пользователя

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **id** позволяет отобразить имя пользователя. Параметры команды позволяют отобразить идентификаторы группы и самого пользователя. Без указания параметра `username` команда выводит информацию для текущего пользователя.

Пример:

```
router#id
uid=1000(admin) gid=1000(admin)
```

passwd

Назначение:

Изменения пароля пользователя.

Синтаксис:

passwd [-a] [-d] [-l] [-u] [username]

Параметр	Описание
-a	Определяет алгоритм для хэширования пароля. Доступны des, md5
-d	Удаление пароля для учетной записи
-l	Блокировка учетной записи
-u	Разблокировка учетной записи
username	Имя учетной записи

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **passwd** позволяет изменить пароль для учетной записи, если параметр `username` не задан, будет изменен пароль для текущего пользователя.

Пример:

Смена пароля для текущей учетной записи (admin):

```
router#passwd
Changing password for admin
Old password:
New password:
Retype password:
Password for admin changed by admin
```

passwd-user

Назначение:

Изменения пароля пользователя.

Синтаксис:

passwd-user

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **passwd-user** позволяет изменить пароль для учетной записи пользователя. Данная команда доступна только администратору.

preset-config

Назначение:

Команда позволяет установить в устройство заранее подготовленные конфигурации.

Синтаксис:

preset-config {-l | --list [filename]} {-s | --set [N] [filename]} {-d | --default } {-b | --backup [filename]} {-r | --restore [filename]} {-v | --view [N]}

Параметр	Описание
-h --help	Выводит справочную информацию по команде
-l [filename] --list [filename]	Выводит список доступных конфигураций из указанного файла
-s [N] [filename] --set [N] [filename]	Устанавливает конфигурацию с номером N из указанного файла
-d --default	Возвращает устройство в заводские настройки и осуществляет сброс пароля
-b [filename] --backup [filename]	Создает архив текущей конфигурации устройства в указанный файл
-r [filename] --restore [filename]	Восстанавливает конфигурацию устройства из указанного файла
-v [N] --view [N]	Выводит конфигурацию с номером N из списка заранее подготовленных

Режим конфигурации:

Команда доступна в linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **preset-config** с ключом **-l** позволяет вывести список доступных для установки заранее подготовленных конфигураций. Конфигурации хранятся в архиве `preset-config.tar.gz`, расположенном в директории `/usr/conf`.

Команда **preset-config** с ключом **-s** позволяет установить в устройство одну из заранее подготовленных конфигураций. Конфигурации хранятся в архиве `preset-config.tar.gz`, расположенном в директории `/usr/conf`.

Команда **preset-config** с ключом **-d** возвращает устройство в заводские настройки и осуществляет сброс пароля. Перед сбросом настроек, система предлагает сделать резервную копию текущих настроек, запрашивая имя файла. Если имя файла не указано, то настройки сохраняются в файл `/home/<username>/backup-config.tar.gz`. Заводские настройки вступают в силу после перезагрузки устройства.

Команда **preset-config** с ключом **-b** позволяет создать резервную копию текущей конфигурации. Если параметр `filename` не задан, то система запрашивает имя файла. Если имя файла не введено, то используется имя файла по-умолчанию - `/home/<username>/backup-config.tar.gz`.

Команда **preset-config** с ключом **-r** позволяет восстановить конфигурацию устройства из заранее созданной резервной копии. Если имя файла не указано, то конфигурация восстанавливается из файла по-умолчанию `/home/<username>/backup-config.tar.gz`.

Команда **preset-config** с ключом **-v** позволяет просмотреть конфигурацию с номером N из списка заранее подготовленных конфигураций.

Пример:

Вывод списка доступных конфигураций:

```
router#preset-config -l
List of available configurations from preset-config.tar.gz
No#      Description
-----
0:       bridge_eth_over_g703/
1:       imux_bridge_eth_over_4xg703/
-----
```

Установка заранее подготовленной конфигурации:

```
router#preset-config -s 0
Do you want to set configuration Ethernet over G.703 (y/n)? y
Management IP address will be 192.168.1.1/24 after reboot
To change management IP address edit configuration file netconfig.sh and then reboot
device
```

Type `cd .` that in the folder `/etc/config` show up installed configuration files

New configuration will be applied after reboot

Восстановление заводских настроек:

```
router#preset-config -d
Current configuration is not default
Do you want to backup current configuration (y/n)? n
New configuration will be applied after reboot
```

Сохранение конфигурации в файл по-умолчанию:

```
router#preset-config -b
Enter filename to save configuration:
Backup configuration is saved in /home/admin/backup-config.tar.gz
```

Восстановление конфигурации из файла по-умолчанию:

```
router#preset-config -r
Do you want to restore current configuration (y/n)? y
Configuration is not default
Do you want to backup current configuration (y/n)? y
Enter filename to save configuration: backup-config-1
Backup configuration is saved in /home/admin/backup-config-1.tar.gz
New configuration will be applied after reboot
```

Просмотр заранее подготовленной конфигурации:

```
router#preset-config -v 26
-----
description.txt:

Ethernet over G.703

-----
netconfig.sh:

#!/bin/sh

## device name
hostname router
```

```
## interface Ethernet 0 configuration
ifconfig eth0 up mtu 1600

## interface HDLC 0 configuration
sethdlc hdlc0 hdlc-eth
ifconfig hdlc0 up

## bridge configuration
brctl addbr br1
brctl addif br1 eth0
brctl addif br1 hdlc0

## interface br1 configuration
ifconfig br1 up 192.168.1.1
```

muxd.conf:

```
interface HDLC 0 E1 2/0
```

post-config.txt:

```
### Post-config recommendations ###
```

```
Current managment IP address is 192.168.1.1/24
To change managment IP address use ifconfig command in Linux shell
```

```
Current synchronization source is internal clock
To change synchronization source use clock source or network-clock-select command in
mux shell
```

Do not forget to save configuration after any changes

execute-config.sh:

```
cp muxd.conf /etc/config/muxd.conf
cp netconfig.sh /etc/config/netconfig.sh
```

requirements.txt:

```
[software]
minPackageVersion=1.7.0.0
```

```
[hardware]
device=MM-221;MM-222;MM-225
slot1=*
slot2=MIM-G703;MIME-2xG703;MIME-2xG703L;MIM-E1A;MIM-2xE1A;MIM-4xE1A
```

reboot

Назначение:

Перезагрузка устройства.

Синтаксис:

reboot [-d delay] [-n] [-f]

Параметр	Описание
-d delay	Отложить перезагрузку на delay секунд
-n	Не выполнять синхронизацию
-f	Принудительная перезагрузка, без вызова init

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **reboot** позволяет перезагрузить устройство.

Пример:

```
router#reboot -d 5
The system is going down NOW!
Sending SIGTERM to all processes
Terminated
Sending SIGKILL to all processes
Requesting system reboot
Restarting system
```

shell

Назначение:

Команда переводит устройство в режим конфигурирования контроллеров и кросс-коннектора.

Синтаксис:

shell [-v]

Параметр	Описание
-v	Вывод версии оболочки конфигурирования контроллеров и кросс-коннектора

Режим конфигурации:

Команда доступна в режиме конфигурирования linux shell:

```
router#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **shell** позволяет перейти в режим конфигурирования контроллеров и кросс-коннектора. В этом режиме помимо конфигурирования доступен просмотр текущего состояния контроллеров и кросс-коннектора.

Для выхода из режима конфигурирования контроллеров и кросс-коннектора необходимо выполнить команду **exit**.

Пример:

```
router#shell
router(shell)#
router(shell)#exit
router#
```

stty

Синтаксис:**update-muxdcfg** {filename}

Параметр	Описание
filename	Имя конфигурационного файла

Режим конфигурации:

Команда доступна в режиме конфигурирования linux shell:

router#

Установка по умолчанию:

отсутствует

Описание:

Команда **update-muxdcfg** позволяет установить конфигурационный файл в качестве загрузочного. В процессе выполнения, команда предлагает применить конфигурацию сразу же или после перезагрузки устройства.

Пример:

Загрузка настроек из файла с именем config-to-update и установка этих настроек в качестве загрузочной конфигурации:

```
router#update-muxdcfg config-to-update
Replace muxd config with new file (y/n)? y
Apply new muxd config immediately (y/n)? n
New muxd config will be applied after reboot
```

См. также:

Команда	Описание
copy running-config ramdisk	Сохранение настроек контроллеров и кросс-коннектора в файл
copy startup-config ramdisk	Сохранение настроек контроллеров и кросс-коннектора, хранящихся в загрузочной конфигурации, в файл

upgrade-software**Назначение:**

Команда позволяет обновить программное обеспечение устройства.

Синтаксис:**upgrade-software** {filename | --check}

Параметр	Описание
filename	Имя файла с архивом программного обеспечения
--check	Проверка целостности установленного программного обеспечения

Режим конфигурации:

Команда доступна в режиме конфигурирования linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **upgrade-software** выполняет обновление ПО из указанного файла. Файл с архивом ПО должен быть предварительно загружен в устройство любым доступным способом — с помощью протокола TFTP или FTP.

Команда **upgrade-software** с ключом **--check** проверяет целостность установленного программного обеспечения.

Пример:

Обновление ПО из файла с именем `zelaamm.0.2.1.3.tgz` и последующая его проверка:

```
router#upgrade-software zelaamm.0.2.1.3.tgz
Current software package version is 0.2.1.3
New software package version is 0.2.1.3
Do you really want upgrade software to version 0.2.1.3 (y/n)? y
Unpacking, checking and installing software package... please wait
Execute post-upgrade script
Update script complete
Upgrade complete
New software will run after reboot

router#upgrade-software --check
Checking current software package...
linuxrc: OK
uImage: OK
bin/[]: OK
bin/[]: OK
bin/@: OK
bin/arp: OK
bin/arping: OK
.
.
.
sr/sbin/update-muxdcfg: OK
usr/sbin/upgrade-software: OK
usr/sbin/vconfig: OK
usr/sbin/vi: OK
Check complete, no errors
```

См. также:

Команда	Описание
tftp	Загрузка файла по протоколу TFTP
ftpget	Загрузка файла по протоколу FTP

version

Назначение:

Команда отображает текущую версию программного обеспечения и информацию об аппаратном оснащении устройства.

Синтаксис:

version

Режим конфигурации:

Команда доступна в режиме конфигурирования linux shell:

```
router#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **version** отображает текущую версию программного обеспечения и информацию об аппаратном оснащении устройства. Также данная команда выводит контактную информацию компании

Zelax. Адрес электронной почты отдела технической поддержки tech@zelax.ru, куда следует обращаться в случае возникновения проблем с оборудованием, и адрес сайта www.zelax.ru.

Параметр	Описание
Software package version	Версия установленного программного обеспечения (ПО)
Built at	Дата создания текущей версии ПО
Shell version	Версия оболочки конфигурирования контроллеров и кросс-коннектора
Hardware control daemon (muxd) version	Версия модуля управления контроллерами и кросс-коннектором
Firmware version	Версия ПО аппаратной части
Linux kernel version	Версия ядра Linux
Hardware	Тип устройства
CPU board	Тип системной платы
X MHz system clock, X MB DRAM, X MB Flash	Частота процессора, объем оперативной памяти, объем Flash-памяти
X Serial (HDLC) interfaces	Количество интерфейсов HDLC
X FastEthernet interface	Количество интерфейсов FastEthernet
Base board	Версия системной платы
Device MAC address	MAC-адрес
Device serial number	Серийный номер
Online technical support: tech@zelax.ru, http://www.zelax.ru Do not forget to report the device serial number mentioned above	Адрес отдела технической поддержки и адрес Web-сайта компании Zelax

Пример:

```
router#version
```

```
-----  
Software package version 0.2.1.3  
Built at Fri Oct 2 15:31:36 MSD 2009
```

```
Shell version 1.7.3 build 1021  
Hardware control daemon (muxd) version 1.7.3 build 1021  
Firmware version 1.4  
Linux kernel version 2.6.26.8-svn968-dirty3
```

```
Hardware: Tested device  
CPU board: MIME-RE100T-4 (4 serial interface, 1 ethernet port)  
100 MHz system clock, 32 MB DRAM, 8 MB Flash  
4 Serial (HDLC) interfaces  
1 FastEthernet interface  
Base board: MUX3-240  
Device MAC address: 00:08:0F:22:CA:FE  
Device serial number: 01234567890
```

```
Online technical support: tech@zelax.ru, http://www.zelax.ru  
Do not forget to report the device serial number mentioned above  
-----
```

uptime

Назначение:

Время работы устройства с момента последней перезагрузки.

Синтаксис:

```
uptime
```

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **uptime** позволяет отобразить время работы устройства с момента последней перезагрузки.

Пример:

```
router#uptime  
18:16:27 up 23:15, load average: 0.00, 0.00, 0.00
```

4 Сетевые настройки и утилиты

arp

Назначение:

Вывод текущего состояния и управление таблицей ARP.

Синтаксис:

arp [-a | -s | -d] [-v] [-n] [-i if] [-D] [-A] [-H hwtype] [hostname]

Параметр	Описание
-a	Отображает все записи таблицы arp
-s	Добавляет запись в таблицу arp
-d	Удаляет запись из таблицы arp
-v	Выводит подробную информацию
-n	Отображать адреса устройств в числовом формате, не производить сопоставление именам устройств
-i if	Отображает записи только для указанного интерфейса
-D	Отображает аппаратный адрес указанного устройства
-A, -p	Протокол
-H hwtype	Тип аппаратного адреса
hostname	Имя устройства

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **arp** позволяет управлять таблицей ARP. Использование команды с ключем **-a** позволяет отобразить текущие записи.

Пример:

```
router#arp -a
? (172.16.1.71) at 00:1e:58:48:44:f4 [ether] on eth0
? (172.16.1.111) at 00:90:27:ae:c2:16 [ether] on eth0
```

arping

Назначение:

Отправка ARP-запроса.

Синтаксис:

arping [-f] [-q] [-b] [-D] [-U] [-A] [-c count] [-w timeout] [-I device] [-s sender] {target}

Параметр	Описание
-f	Прекратить работу после приема первого ответа
-q	Не отображать вывод команды
-b	Отправлять только широковещательные пакеты
-D	Режим дублированного обнаружения адреса, в соответствии с RFC2131
-U	Предоставление таблицы ARP соседним устройствам
-A	Работает аналогично -U, но используются пакеты ARP REPLY вместо ARP REQUEST
-c count	Отправка прерывается после отсылки count пакетов ARP REQUEST
-w timeout	Таймаут завершения работы утилиты, в секундах
-I device	Интерфейс, через который производится отправка пакетов

-s sender	IP-адрес отправителя, использующийся в пакетах
target	IP-адрес, на который производится отправка пакетов

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **arping** позволяет отправить ARP-запрос. С помощью этой команды можно отправлять ARP-запросы явно выбирая адрес источника и интерфейс, через который будет проводиться отправка.

Пример:

Отправка пакетов с адреса 172.16.1.112 на адрес 172.16.1.71 с остановкой работы утилиты после отправки пяти пакетов:

```
router#arping -c 5 -s 172.16.1.112 172.16.1.71
arping: warning: interface eth0 is ignored
ARPING to 72.16.1.112 from 172.16.1.112 via eth0
Unicast reply from 172.16.1.71 [0:1e:58:48:44:f4] 171017.295ms
Unicast reply from 172.16.1.71 [0:1e:58:48:44:f4] 172022.551ms
Unicast reply from 172.16.1.71 [0:1e:58:48:44:f4] 173030.484ms
Unicast reply from 172.16.1.71 [0:1e:58:48:44:f4] 174038.481ms
Unicast reply from 172.16.1.71 [0:1e:58:48:44:f4] 175046.474ms
Sent 5 probe(s) (1 broadcast(s))
Received 5 reply (0 request(s), 0 broadcast(s))
```

ethtool

Назначение:

Настройка и отображение параметров интерфейса Ethernet.

Синтаксис:

```
ethtool [-s] {DEVNAME} [speed 10|100|1000] [duplex half|full] [autoneg on|off]
```

Параметр	Описание
DEVNAME	Имя интерфейса.
speed	Скорость работы интерфейса – 10/100/1000 Мбит/с.
duplex	Режим работы интерфейса – полудуплексный или дуплексный.
autoneg	Автоматическое согласование параметров соединения в соответствии со стандартами 802.3/802.3u.

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ethtool**, введенная без ключа, отображает текущие параметры работы указанного интерфейса. С указанием ключа **s** позволяет изменить параметры работы интерфейса Ethernet.

Изменять какие-либо настройки интерфейса Ethernet, не указанные выше, не рекомендуется, это может привести к нестабильной работе устройства.

Пример:

Установка скорости работы интерфейса eth0.

```
router#ethtool -s eth0 speed 10
```

ifconfig

Назначение:

Настройка и отображения текущего состояния сетевых интерфейсов.

Синтаксис:

```
ifconfig [-a] [interface] [address] [[-]broadcast [ADDRESS]] [[-]pointopoint [ADDRESS]] [netmask ADDRESS] [dstaddr ADDRESS] [keepalive NN] [hw ether ADDRESS] [metric NN] [mtu NN] [[-]arp] [txqueuelen NN] [up|down] [clear]
```

Параметр	Описание
-a	Позволяет применить параметры команды <code>ifconfig</code> ко всем интерфейсам в системе. Без указания интерфейса выводит информацию обо всех интерфейсах в системе.
interface	Имя интерфейса.
address	IP-адреса интерфейса.
broadcast	Устанавливает широковещательный адрес. Если IP-адрес не указан, то устанавливает или сбрасывает флаг IFF_BROADCAST .
pointopoint	Включает режим работы "точка-точка" на интерфейсе и указывает IP-адрес удаленного устройства. Если IP-адрес не указан, то устанавливает или сбрасывает флаг IFF_POINTOPOINT .
netmask	Сетевая маска интерфейса.
dstaddr	IP-адрес удаленного устройства при работе в режиме "точка-точка". Аналогичен параметру <code>pointopoint</code> .
keepalive	Интервал отправки сообщений <code>keepalive</code> , в секундах.
hw ether	MAC-адрес интерфейса.
metric	Метрика интерфейса. Значение по-умолчанию 0.
mtu	Максимальный размер пакета (Maximum Transfer Unit — MTU) для интерфейса.
arp	Включение/отключение протокола ARP на интерфейсе.
txqueuelen	Длина очереди передатчика на интерфейсе.
up	Включение интерфейса.
down	Выключение интерфейса.
clear	Сброс статистики интерфейса.

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда `ifconfig` используется для настройки и отображения текущего состояния сетевых интерфейсов. В качестве сетевых интерфейсов могут быть указаны `eth`, `br`, `hdlc`.

Введенная без параметров команда `ifconfig` выводит текущие параметры всех активных интерфейсов.

Перед выполнением команды `ifconfig` для интерфейса `hdlc` следует установить инкапсуляцию командой `sethdlc`.

Пример:

Вывод информации об интерфейсе `br0`:

```
Router# ifconfig br0
br0      Link encap:Ethernet  HWaddr 00:08:0F:22:CA:FE
         inet addr:172.16.1.112  Bcast:172.16.255.255  Mask:255.255.0.0
```

```

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:136361 errors:0 dropped:0 overruns:0 frame:0
TX packets:132327 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:13407008 (12.7 MiB) TX bytes:15079602 (14.3 MiB)

```

ip

Назначение:

Просмотр и управление маршрутизацией, сетевыми интерфейсами и туннелями.

Синтаксис:

```

ip [-f[amily]] {inet | inet6 | link} | -o[ne]line]] {address | route | link | tunnel | rule} {set | add | delete |
show | list | flush}

```

Параметр	Описание
-f -family	Определяет идентификатор протокола. Если опция не указана, семейство протоколов будет определено из других опций
inet	Протокол IPv4
inet6	Протокол IPv6
link	Идентификатор, обозначающий, что сетевой протокол не задействован
-o -online	Вывод каждой записи на отдельной строке
address	Адрес на интерфейсе в соответствии с протоколом (IPv4 или IPv6)
route	Запись таблицы маршрутизации
link	Сетевой интерфейс
tunnel	Туннель IP
rule	Правило маршрутизации
set	Изменения параметров
add	Добавить значения
delete	Удалить значения
show	Отобразить параметры
list	Аналогично show в некоторых объектах
flush	Очистить параметры

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ip** позволяет отображать и изменять параметры объектов системы, таких как интерфейс, маршрутизация, туннелирование и т.д.

Пример:

Отображение параметров только активных интерфейсов:

```

router#ip link show up
1: lo: <LOOPBACK,UP,10000> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
8: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 250
   link/ether 00:ee:10:02:21:02 brd ff:ff:ff:ff:ff:ff

```

ipaddr

Назначение:

Просмотр и управление IP-адресом.

Синтаксис:

```
ipaddr { {add | del} IFADDR dev STRING | [show | flush] [dev STRING] [to PREFIX] }
```

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ipaddr** позволяет отображать и изменять IP-адреса на интерфейсах устройства.

Пример:

```
router#ipaddr show eth0
8: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 250
   link/ether 00:ee:10:02:21:02 brd ff:ff:ff:ff:ff:ff
   inet 192.168.0.24/24 brd 192.168.0.255 scope global eth0
```

iptunnel**Назначение:**

Работа с IP-туннелями.

Синтаксис:

```
iptunnel {add | change | del | show} [name] [mode {pip | gre | sit}] [remote ADDR] [local ADDR] [ttl TTL]
```

Параметр	Описание
add	Добавление туннеля
change	Изменение туннеля
del	Удаление туннеля
show	Просмотр туннеля
name	Название туннеля
mode	Тип туннеля
remote	Удаленный IP-адрес
local	Локальный IP-адрес
ttl	Время жизни пакета

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **iptunnel** позволяет выполнять действия над туннелями.

Пример:

```
router#iptunnel show
tunl0: ip/ip remote any local any ttl inherit nopmtudisc
gre0: gre/ip remote any local any ttl inherit nopmtudisc
```

nslookup**Назначение:**

Запрос к DNS-серверу.

Синтаксис:**nslookup** {host} [server]

Параметр	Описание
host	Имя устройства, IP-адрес которого запрашивается у DNS-сервера
server	DNS-сервер

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **nslookup** позволяет отправить запросить DNS-серверу на соответствие имени устройства IP-адресу.

Пример:

```
router#nslookup examplehost
Server:      192.168.0.1
Address 1:  192.168.0.18
```

ping**Назначение:**

Отправка ICMP-запроса.

Синтаксис:

ping [-LRUbdnqrVvAa] [-c count] [-i interval] [-w deadline] [-p pattern] [-s packetsize] [-t ttl] [-I interface | address] [-M mtu discovery hint] [-S sndbuf] [-T timestamp option] [-Q tos] [hop1 ...] {destination}

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ping** отправляет пакет ICMP ECHO_REQUEST к узлу сети указанному параметром destination.

Пример:

```
router#ping 192.168.0.105
PING 192.168.0.105 (192.168.0.105) 56(84) bytes of data.
64 bytes from 192.168.0.105: icmp_seq=0 ttl=128 time=3.29 ms
64 bytes from 192.168.0.105: icmp_seq=1 ttl=128 time=0.670 ms
64 bytes from 192.168.0.105: icmp_seq=2 ttl=128 time=1.04 ms
64 bytes from 192.168.0.105: icmp_seq=3 ttl=128 time=0.407 ms
64 bytes from 192.168.0.105: icmp_seq=4 ttl=128 time=0.439 ms

--- 192.168.0.105 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4014ms
rtt min/avg/max/mdev = 0.407/1.171/3.295/1.086 ms, pipe 2
```

sethdlc**Назначение:**

Настройка интерфейса hdlc.

Синтаксис:

sethdlc {interface} {hdlc | hdlc-eth | cisco [interval] | ppp [interval] | ppp-link}

Параметр	Описание
interface	Имя интерфейса hdlc
hdlc	Установка типа инкапсуляции hdlc
hdlc-eth	Установка типа инкапсуляции hdlc-eth
cisco	Установка типа инкапсуляции cisco
ppp	Установка типа инкапсуляции ppp
interval	Установка периода отправки сообщений keeralive, в секундах. Доступный диапазон значений 0 – 100. Установка значения interval = 0 отключает отправку сообщений keeralive. Значение по умолчанию – 10
ppp-link	Установка типа инкапсуляции ppp-link

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **sethdlc** используется для настройки интерфейса hdlc. В текущей версии ПО доступны пять типов инкапсуляции – hdlc, hdlc-eth, cisco, ppp, ppp-link.

Пример:

Установка типа инкапсуляции hdlc-eth на интерфейсе hdlc0:

```
router#sethdlc hdlc0 cisco interval 15
```

telnet

Назначение:

Подключение к удаленному устройству по протоколу Telnet.

Синтаксис:

telnet [-a] [-l USER] HOST [PORT]

Параметр	Описание
-a	Регистрация в удаленной системе с использованием текущего имени пользователя
-l	Регистрация в удаленной системе с использованием имени пользователя, заданного параметром USER
HOST	Имя удаленного узла, к которому будет производиться подключение
PORT	Порт для подключения, если не указан используется порт 23

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **telnet** позволяет осуществить подключение к удаленному узлу по протоколу Telnet.

Пример:

```
router#telnet 192.168.0.105
```

traceroute

Назначение:

Трассировка маршрута к узлу сети.

Синтаксис:

```
traceroute [-F]ldnrv] [-f 1st_ttl] [-m max_ttl] [-p port#] [-q nqueries] [-s src_addr] [-t tos] [-w wait] [-g gateway] [-i iface] [-z pausemsecs] {host} [data size]
```

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **traceroute** позволяет маршрут прохождения пакетов до указанного узла сети.

Пример:

```
router#traceroute 192.168.0.105
traceroute to 192.168.0.105 (192.168.0.105), 30 hops max, 38 byte packets
 1  192.168.0.105 (192.168.0.105)  1.130 ms  0.653 ms  0.821 ms
```

5 Режим моста (bridge), протокол Spanning Tree (STP)

brctl addbr

Назначение:

Создание Ethernet-моста (bridge).

Синтаксис:

brctl addbr {bridge}

Параметр	Описание
bridge	Имя Ethernet-моста.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **brctl addbr** создает Ethernet-мост. Имя моста рекомендуется задавать в форме br с последующим цифровым индексом. Такая форма позволит однозначно идентифицировать интерфейс.

Пример:

Создание Ethernet-моста с именем br1:

```
router#brctl addbr br1
```

brctl delbr

Назначение:

Удаление Ethernet-моста (bridge).

Синтаксис:

brctl delbr {bridge}

Параметр	Описание
bridge	Имя Ethernet-моста.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **brctl delbr** удаляет Ethernet-мост.

Пример:

Удаление Ethernet-моста с именем br1:

```
router#brctl delbr br1
```

brctl addif

Назначение:

Включение интерфейса в Ethernet-мост.

Синтаксис:

brctl addif {bridge} {device}

Параметр	Описание
bridge	Имя Ethernet-моста.
device	Интерфейс, который будет включен в мост.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **brctl addif** включает интерфейс в Ethernet-мост. В качестве интерфейсов могут быть указаны ethernet, hdlc.

Пример:

Включение интерфейса hdlc1 в Ethernet-мост br1:

```
router#brctl addif br1 hdlc1
```

brctl delif

Назначение:

Исключение интерфейса из Ethernet-моста.

Синтаксис:

brctl delif {bridge} {device}

Параметр	Описание
bridge	Имя Ethernet-моста.
device	Интерфейс, который будет исключен из моста.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **brctl delif** исключает интерфейс из Ethernet-моста. В качестве интерфейсов могут быть указаны ethernet, hdlc.

Пример:

```
router#brctl delif br1 hdlc1
```

brctl setageing

Назначение:

Задаёт “время жизни” MAC-адресов для Ethernet-моста.

Синтаксис:

brctl setageing {bridge} {time}

Параметр	Описание
bridge	Имя Ethernet-моста.
time	“Время жизни” MAC-адресов, в секундах.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

300 сек.

Описание:

Команда **brctl setageing** указывает “время жизни” MAC-адресов, если в течение указанного времени на мост не приходят кадры с определённого MAC-адреса, этот адрес удаляется из базы данных Forwarding DataBase (fdb).

Пример:

```
router#brctl setageing br1 300
```

brctl setbridgeprio**Назначение:**

Задаёт приоритет Ethernet-моста.

Синтаксис:

brctl setbridgeprio {bridge} {prio}

Параметр	Описание
bridge	Имя Ethernet-моста.
prio	Приоритет Ethernet-моста. Параметр может принимать значения от 0 до 65535.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

32768.

Описание:

Команда **brctl setbridgeprio** устанавливает приоритет Ethernet-моста. Данный параметр используется протоколом STP.

Пример:

```
router#brctl setageing br1 300
```

brctl setfd**Назначение:**

Задаёт интервал forward delay Ethernet-моста.

Синтаксис:**brctl setfd** {bridge} {time}

Параметр	Описание
bridge	Имя Ethernet-моста.
time	Интервал forward delay Ethernet-моста, в секундах.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

15 сек.

Описание:

Команда **brctl setfd** устанавливает интервал forward delay Ethernet-моста. Данный параметр используется протоколом STP. Указывает время нахождения портов моста в состояниях Listening и Learning.

Пример:

router#brctl setfd br1 10

brctl sethello**Назначение:**

Задаёт интервал между рассылками сообщений BPDU.

Синтаксис:**brctl sethello** {bridge} {time}

Параметр	Описание
bridge	Имя Ethernet-моста.
time	Интервал hello time, в секундах.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

2 сек.

Описание:

Команда **brctl sethello** устанавливает интервал hello time. Данный параметр используется протоколом STP. Указывает интервал между рассылками Configuration BPDU и TCN BPDU.

Пример:

router#brctl sethello br1 1

brctl setmaxage**Назначение:**

Задаёт время хранения сообщений BPDU.

Синтаксис:

brctl setmaxage {bridge} {time}

Параметр	Описание
bridge	Имя Ethernet-моста.
time	Интервал хранения сообщений BPDU, в секундах.

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

20 сек.

Описание:

Команда **brctl setmaxage** устанавливает время хранения сообщений BPDU. Данный параметр используется протоколом STP. Если в течении указанного интервала устройство не получило сообщений BPDU, то считается, что произошли изменения в топологии сети.

Пример:

```
router#brctl setmaxage br1 1
```

brctl setpathcost

Назначение:

Задаёт значение стоимости порта, включенного в Ethernet-мост.

Синтаксис:

brctl setpathcost {bridge} {port} {cost}

Параметр	Описание
bridge	Имя Ethernet-моста.
port	Порт, включенный в мост.
cost	Стоимость порта.

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Интерфейс Ethernet - 19.

Интерфейс hdlc – 100.

Описание:

Команда **brctl setpathcost** устанавливает значение стоимости порта. Данный параметр используется протоколом STP. Используется при вычислении маршрута до корневого (root) устройства.

Для того, чтобы привести значения параметра к стандартным величинам необходимо выполнить команду `brctl setpathcost br1 hdlc1 500`.

Пример:

Присвоение стоимости 100 интерфейсу eth0.

```
router#brctl setpathcost br1 eth0 100
```

brctl setportprio

Назначение:

Задает приоритет порта, включенного в Ethernet-мост.

Синтаксис:

brctl setportprio {bridge} {port} {prio}

Параметр	Описание
bridge	Имя Ethernet-моста.
port	Порт, включенный в мост.
prio	Приоритет порта. Параметр может принимать значения от 0 до 255.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

128.

Описание:

Команда **brctl setportprio** устанавливает приоритет порта. Данный параметр используется протоколом STP. Используется при выборе корневого порта (root port).

Пример:

```
router#brctl setportprio br1 hdlc1 1
```

brctl stp

Назначение:

Включение/выключение протокола STP на Ethernet-мосту.

Синтаксис:

brctl stp {bridge} {on|off}

Параметр	Описание
bridge	Имя Ethernet-моста.
on	Включение протокола STP.
off	Выключение протокола STP.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **brctl stp** производит включение или выключение протокола STP. Протокол STP используется для приведения сети к древовидной топологии.

Пример:

```
router#brctl stp br1 on
```

brctl setstpver

Назначение:

Выбор версии протокола STP.

Синтаксис:

brctl setstpver {bridge} {stp | rapidstp}

Параметр	Описание
bridge	Имя Ethernet-моста.
stp	Протокол STP (802.1d).
rapidstp	Протокол RSTP (802.1w).

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

RSTP.

Описание:

Команда **brctl setstpver** позволяет выбрать версию протокола STP. В текущей версии ПО поддерживаются две версии протокола - 802.1d и 802.1w.

Пример:

```
router#brctl setstpver br1 stp
```

brctl setportedge

Назначение:

Включение/отключение пограничного режима работы порта.

Синтаксис:

brctl setportedge {bridge} {if} {off | on | auto}

Параметр	Описание
bridge	Имя Ethernet-моста.
if	Интерфейс.
off	Установить порт в обычный режим работы.
on	Установить порт в пограничный режим работы.
auto	Автоматический выбор режима.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

auto.

Описание:

Команда **brctl setportedge** позволяет выбрать режим работы для пограничного порта. Порт, работающий в пограничном режиме, сразу же после включения переходит в режим передачи данных. Изменение состояния (up/down) такого порта не приводит к перестроению топологии сети.

Данный режим работы поддерживается только протоколом RSTP.

Пример:

```
router# brctl setportedge br1 eth0 on
```

brctl setportp2p

Назначение:

Выбор режима работы порта.

Синтаксис:

brctl setportp2p {bridge} {if} {off | on | auto}

Параметр	Описание
bridge	Имя Ethernet-моста.
if	Интерфейс.
off	Установить порт в разделяемый режим работы.
on	Установить порт в режим работы "точка-точка".
auto	Автоматический выбор режима.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

auto.

Описание:

Команда **brctl setportp2p** позволяет выбрать режим работы порта, подключенного к другому мосту. Существует два типа соединений между устройствами, поддерживающими протокол RSTP – разделяемый и "точка-точка". В протоколе RSTP предусмотрены механизмы для сокращения времени перестроения топологии на портах, работающих в режимах "точка-точка".

Данный режим работы поддерживается только протоколом RSTP.

Пример:

```
router# brctl setportp2p br1 hd1c0 on
```

brctl show**Назначение:**

Отображает информацию о созданных Ethernet-мостах.

Синтаксис:

brctl show

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **brctl show** отображает информацию о созданных Ethernet-мостах и включенных в них интерфейсах.

Пример:

```
router#brctl show
bridge name      bridge id          STP enabled      interfaces
br1              8000.00080f22cafe yes               hd1c1
                hd1c2
```

brctl showmacs

Назначение:

Отображает таблицу MAC-адресов для указанного Ethernet-моста.

Синтаксис:

brctl showmacs {bridge}

Параметр	Описание
bridge	Имя Ethernet-моста.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **brctl showmacs** отображает таблицу MAC-адресов для указанного Ethernet-моста. Таблица MAC-адресов обновляется при прохождении кадра через мост. Столбец ageing timer показывает сколько секунд назад был получен кадр, с данным MAC-адресом в поле Source MAC address. По истечении "времени жизни" MAC-адрес удаляется из таблицы.

Пример:

```
router#brctl showmacs br1
port no mac addr          is local?    ageing timer
 1      00:08:0f:22:ca:fe    yes          0.00
```

См. также:

Команда	Описание
brctl setageing	"Время жизни" MAC-адресов, в секундах.

brctl showstp

Назначение:

Отображает состояние протокола STP для указанного Ethernet-моста.

Синтаксис:

brctl showstp {bridge} [if]

Параметр	Описание
bridge	Имя Ethernet-моста.
if	Интерфейс.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **brctl showstp** состояние протокола STP для указанного Ethernet-моста.

Пример:

router#brctl showstp br1

br1

bridge id	8000.00080f22cafe		
designated root	8000.00080f22cafe		
root port	0	path cost	0
max age	20.00	bridge max age	20.00
hello time	2.00	bridge hello time	2.00
forward delay	15.00	bridge forward delay	15.00
ageing time	300.00		
hello timer	0.00	tcn timer	0.00
topology change timer	0.00	gc timer	0.00
flags			

hdlc1 (1)

port id	8001	state	disabled
designated root	8000.00080f22cafe	path cost	300
designated bridge	8000.00080f22cafe	message age timer	0.00
designated port	8001	forward delay timer	0.00
designated cost	0	hold timer	0.00
flags			

hdlc2 (2)

port id	8002	state	disabled
designated root	8000.00080f22cafe	path cost	100
designated bridge	8000.00080f22cafe	message age timer	0.00
designated port	8002	forward delay timer	0.00
designated cost	0	hold timer	0.00
flags			

6 Виртуальные локальные сети (VLAN)

vconfig add

Назначение:

Команда создает виртуальный интерфейс.

Синтаксис:

```
vconfig add {interface-name} {vlan_id}
```

Параметр	Описание
interface-name	Имя физического интерфейса, для которого будет создан виртуальный интерфейс
vlan_id	Идентификатор VLAN, который будет присвоен виртуальному интерфейсу

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **vconfig add** позволяет создать виртуальный интерфейс на основе физического интерфейса.

Пример:

```
router#vconfig add eth0 20
```

vconfig rem

Назначение:

Команда удаляет виртуальный интерфейс.

Синтаксис:

```
vconfig rem {vlan-name}
```

Параметр	Описание
vlan-name	Имя виртуального интерфейса, который будет удален

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **vconfig rem** позволяет удалить виртуальный интерфейс. Имя виртуального интерфейса вводится в формате, указанным командой **vconfig set_name_type**.

Пример:

```
router#vconfig rem eth0.20
```

vconfig set_flag

Назначение:

Команда задает обработку тега VLAN.

Синтаксис:

vconfig set_flag {vlan-name} {flag-num 0 | 1}

Параметр	Описание
vlan-name	Имя виртуального интерфейса
flag-num	Признак обработки тега VLAN Ethernet-кадра. Всегда должен принимать значение 1
0	Сохранение тегов VLAN Ethernet-кадра
1	Удаление тегов VLAN из Ethernet-кадра

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

0.

Описание:

Команда **vconfig set_flag** позволяет скорректировать работу устройства, касательно обработки тегов VLAN. При установке значения 1 заголовки кадров Ethernet будут корректироваться с целью удаления тегов VLAN. Используемое по умолчанию значение 0 обеспечивает сохранение тегов VLAN в кадрах Ethernet. Такая корректировка может потребоваться некоторыми программами пакетной фильтрации.

Пример:

```
router#vconfig set_flag eth0.20 1 1
```

vconfig set_name_type**Назначение:**

Команда изменяет способ именования интерфейсов VLAN.

Синтаксис:

vconfig set_name_type {name-type}

Параметр	Описание
name-type	Способ отображения интерфейса

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

DEV_PLUS_VID_NO_PAD.

Описание:

Команда **vconfig set_name_type** позволяет изменить способ именования интерфейсов VLAN. Параметр name-type может принимать четыре значения:

- VLAN_PLUS_VID – виртуальный интерфейс именуется с формате <vlan><идентификатор VLAN>;
- VLAN_PLUS_VID_NO_PAD – тоже что и предыдущий параметр, но без отображения незначащих нулей;
- DEV_PLUS_VID – виртуальный интерфейс именуется с формате <физический интерфейс>.<идентификатор VLAN>;

- DEV_PLUS_VID_NO_PAD - тоже что и предыдущий параметр, но без отображения незначащих нулей.

7 Режим объединения каналов (bond)

bondctl add

Назначение:

Создание логического интерфейса bond.

Синтаксис:

bondctl add {bond}

Параметр	Описание
bond	Имя интерфейса bond.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **bondctl add** создает логический интерфейс объединения каналов – интерфейс bond. Данный интерфейс позволяет объединить несколько физических каналов в один логический. Объединение физических каналов позволяет увеличить общую пропускную способность и повысить отказоустойчивость сети в случае выхода из строя одного из каналов. Имя интерфейса рекомендуется задавать в форме bond с последующим цифровым индексом. Такая форма позволит однозначно идентифицировать интерфейс.

Пример:

Создание интерфейса bond с именем bond0:

```
router#bondctl add bond0
```

bondctl del

Назначение:

Удаление логического интерфейса bond.

Синтаксис:

bondctl del {bond}

Параметр	Описание
bond	Имя интерфейса bond.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **bondctl del** удаляет интерфейс bond.

Пример:

Удаление логического интерфейса bond с именем bond0:

```
router#bondctl del bond0
```

bondctl attach

Назначение:

Включение интерфейса в группу объединения.

Синтаксис:

bondctl attach {bond} {device}

Параметр	Описание
bond	Имя интерфейса bond.
device	Интерфейс, который будет включен в группу.

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **bondctl attach** включает интерфейс в группу объединения. В качестве интерфейсов могут быть указаны ethernet, hdlc.

Пример:

Включение интерфейса hdlc1 в группу объединения bond0:

```
router#bondctl attach bond0 hdlc1
```

bondctl dettach

Назначение:

Исключение интерфейса из группы объединения.

Синтаксис:

bondctl dettach {bond} {device}

Параметр	Описание
bond	Имя интерфейса bond.
device	Интерфейс, который будет включен в группу.

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **bondctl dettach** исключает интерфейс из группы объединения. В качестве интерфейсов могут быть указаны ethernet, hdlc.

Пример:

Исключение интерфейса hdlc1 из группы объединения bond0:

```
router#bondctl detach bond0 hd1c1
```

bondctl show

Назначение:

Отображает список созданных интерфейсов bond.

Синтаксис:

bondctl show

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **bndctl show** отображает список созданных интерфейсов bond.

Пример:

```
router#bondctl show
bond0
bond1
bond2
bond3
```

bondctl detail

Назначение:

Отображает информацию о созданном интерфейсе bond.

Синтаксис:

bondctl detail {bond}

Параметр	Описание
bond	Имя интерфейса bond.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **bondctl detail** отображает информацию о созданном интерфейсе bond и выводит список включенных в данную группу интерфейсов.

8 Пример настройки объединения каналов (bond)

Объединение четырех каналов SHDSL в bond для передачи Ethernet с максимальной пропускной способностью. Схема подключения указана на рисунке 1.

Ошибка! Объект не может быть создан из кодов полей редактирования.

Рисунок 1. Схема локальной сети

Настройка устройства MM-222-1

netconfig.sh

```
#!/bin/sh

## device name
hostname router-master

## interface Ethernet 0 configuration
ifconfig eth0 up mtu 1600

## interface HDLC 0 configuration
sethdlc hdlc0 hdlc-eth
sethdlc hdlc1 hdlc-eth
sethdlc hdlc2 hdlc-eth
sethdlc hdlc3 hdlc-eth
ifconfig hdlc0 down
ifconfig hdlc1 down
ifconfig hdlc2 down
ifconfig hdlc3 down

## Bond configuration
bondctl add bond0
bondctl attach bond0 hdlc0 hdlc1 hdlc2 hdlc3

ifconfig hdlc0 up
ifconfig hdlc1 up
ifconfig hdlc2 up
ifconfig hdlc3 up
ifconfig bond0 up

## bridge configuration
brctl addbr br1
brctl addif br1 eth0
brctl addif br1 bond0

## interface br1 configuration
ifconfig br1 up 192.168.1.1
```

muxd.conf

```
controller SHDSL 1/0
    clocking-scheme synchronous
    modulation TC-PAM 64
    no framing
    line-rate 12672
    no line-probing
!
controller SHDSL 1/1
    clocking-scheme synchronous
    modulation TC-PAM 64
    no framing
    line-rate 12672
    no line-probing
```

```

!
controller SHDSL 2/0
    clocking-scheme synchronous
    modulation TC-PAM 64
    no framing
    line-rate 12672
    no line-probing
!
controller SHDSL 2/1
    clocking-scheme synchronous
    modulation TC-PAM 64
    no framing
    line-rate 12672
    no line-probing
!
interface HDLC 0 SHDSL 1/0
interface HDLC 1 SHDSL 1/1
interface HDLC 2 SHDSL 2/0
interface HDLC 3 SHDSL 2/1

```

Настройка устройства MM-222-2

netconfig.sh

```

#!/bin/sh

## device name
hostname router-slave

## interface Ethernet 0 configuration
ifconfig eth0 up mtu 1600

## interface HDLC 0 configuration
sethdlc hdlc0 hdlc-eth
sethdlc hdlc1 hdlc-eth
sethdlc hdlc2 hdlc-eth
sethdlc hdlc3 hdlc-eth
ifconfig hdlc0 down
ifconfig hdlc1 down
ifconfig hdlc2 down
ifconfig hdlc3 down

## Bond configuration
bondctl add bond0
bondctl attach bond0 hdlc0 hdlc1 hdlc2 hdlc3

ifconfig hdlc0 up
ifconfig hdlc1 up
ifconfig hdlc2 up
ifconfig hdlc3 up
ifconfig bond0 up

## bridge configuration
brctl addbr br1
brctl addif br1 eth0
brctl addif br1 bond0

## interface br1 configuration
ifconfig br1 up 192.168.1.2

```

muxd.conf

```

controller SHDSL 1/0
    core-function NTU slave
    clocking-scheme synchronous

```

```
        no framing
        no line-probing
!
controller SHDSL 1/1
    core-function NTU slave
    clocking-scheme synchronous
    no framing
    no line-probing
!
controller SHDSL 2/0
    core-function NTU slave
    clocking-scheme synchronous
    no framing
    no line-probing
!
controller SHDSL 2/1
    core-function NTU slave
    clocking-scheme synchronous
    no framing
    no line-probing
!
network-clock-select 1 SHDSL 2/0
network-clock-select 1 SHDSL 2/1
network-clock-select 2 SHDSL 1/0
network-clock-select 2 SHDSL 1/1
!
interface HDLC 0 SHDSL 1/0
interface HDLC 1 SHDSL 1/1
interface HDLC 2 SHDSL 2/0
interface HDLC 3 SHDSL 2/1
```

9 Режим Multilink Point-to-Point (MLP)

pppctl add

Назначение:

Создание логического интерфейса ppp.

Синтаксис:

```
pppctl add {ppp} {ipcp | bcp}
```

Параметр	Описание
ppp	Имя интерфейса ppp.
ipcp	Тип создаваемого интерфейса ppp – интерфейс ipv4 (уровень 3 модели OSI)
bcp	Тип создаваемого интерфейса ppp – интерфейс Ethernet (уровень 2 модели OSI)

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **pppctl add** создает логический point-to-point интерфейс. Имя интерфейса рекомендуется задавать в форме ppp с последующим цифровым индексом. Такая форма позволит однозначно идентифицировать интерфейс.

Пример:

Создание Ethernet point-to-point интерфейса с именем ppp0:

```
router#pppctl add ppp0 bcp
```

pppctl del

Назначение:

Удаление логического интерфейса ppp.

Синтаксис:

```
pppctl del {ppp}
```

Параметр	Описание
ppp	Имя интерфейса ppp.

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **pppctl del** удаляет интерфейс ppp.

Пример:

Удаление логического интерфейса ppp с именем ppp0:
router#pppctl del ppp0

pppctl set link multilink

Назначение:

Устанавливает тип интерфейса ppp multilink.

Синтаксис:

pppctl set {ppp} link multilink

Параметр	Описание
ppp	Имя интерфейса ppp.

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **pppctl set link multilink** задает тип интерфейса ppp multilink. Данный интерфейс позволяет объединить несколько физических каналов в один логический. Объединение физических каналов позволяет увеличить общую пропускную способность и повысить отказоустойчивость сети в случае выхода из строя одного из каналов. **Объединять в группу multilink можно только интерфейсы hdlc с типом инкапсуляции ppp-link.**

Пример:

Задание типа интерфейса ppp multilink для интерфейса ppp0:

```
router#pppctl set ppp0 link multilink
```

pppctl link add

Назначение:

Добавляет интерфейс в группу multilink.

Синтаксис:

pppctl link {ppp} add {number} {type} {device}

Параметр	Описание
ppp	Имя интерфейса ppp.
number	Порядковый номер добавляемого интерфейса
type	Тип интерфейса. В текущей версии ПО доступно добавление только интерфейсов типа serial .
device	Добавляемый интерфейс

Режим конфигурации:

Linux shell:

router#

Установка по умолчанию:

Отсутствует.

Описание:

Команда **pppctl link add** добавляет интерфейс в группу multilink. Объединять в группу multilink можно только интерфейсы hdlc с типом инкапсуляции ppp-link.

Пример:

Добавление в группу multilink ppp0 интерфейса hdlc0:

```
router#pppctl link ppp0 add 1 serial hdlc0
```

pppctl link del

Назначение:

Удаляет интерфейс из группы multilink.

Синтаксис:

pppctl link {ppp} del {number}

Параметр	Описание
ppp	Имя интерфейса ppp.
number	Номер удаляемого интерфейса

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **pppctl link del** удаляет интерфейс из группы multilink.

Пример:

Удаление из группы multilink ppp0 интерфейса с номером 1:

```
router#pppctl link ppp0 del 1
```

pppctl show

Назначение:

Отображает информацию о созданных интерфейсах ppp.

Синтаксис:

pppctl show [{ppp}]

Параметр	Описание
ppp	Имя интерфейса ppp.

Режим конфигурации:

Linux shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **pppctl show** отображает информацию о созданных интерфейсах ppp.

Пример:

```
router#pppctl show
```

Name	Type	State	Link
ppp0	bcp	up	multilink

```
router#pppctl show ppp0
```

```
ppp0 type: bcp, state: up
```

```
Main link:
```

```
    Type: multilink
```

```
    Link: yes, opened: yes, stopped: no
```

```
    Fragmentation: off, minimum fragment size: 64
```

```
    Keepalive: off, retries: 5
```

```
    Timeout: 1s, debug: off
```

```
    Packets tx: 8, rx: 8, no proto: 0
```

```
    Tx stopped: 0times
```

```
    Rx fragments lost: 0, dropped: 0
```

```
    Protocols: BCP[opened], BDP, LCP
```

```
Multilink link 1:
```

```
    Type: serial hdlc0
```

```
    Link: yes, opened: yes, stopped: no
```

```
    Keepalive: 10, retries: 5
```

```
    Timeout: 1s, debug: off
```

```
    Packets tx: 94, rx: 89, no proto: 0
```

```
    Tx stopped: 0times
```

```
    Protocols: LCP[opened], Multilink
```

```
Multilink link 2:
```

```
    Type: serial hdlc1
```

```
    Link: yes, opened: yes, stopped: no
```

```
    Keepalive: 10, retries: 5
```

```
    Timeout: 1s, debug: off
```

```
    Packets tx: 96, rx: 91, no proto: 0
```

```
    Tx stopped: 0times
```

```
    Protocols: LCP[opened], Multilink
```

10 Пример настройки Multilink Point-to-Point

Объединение четырех каналов SHDSL в multilink для передачи Ethernet с максимальной пропускной способностью. Схема подключения указана на рисунке 2.

Ошибка! Объект не может быть создан из кодов полей редактирования.

Рисунок 2. Схема локальной сети

Настройка устройства MM-222-1

netconfig.sh

```
#!/bin/sh

## device name
hostname router-m

## interface Ethernet 0 configuration
ifconfig eth0 up mtu 1600

## interface HDLC 0 configuration
sethdlc hdlc0 ppp-link
sethdlc hdlc1 ppp-link
ifconfig hdlc0 down
ifconfig hdlc1 down

sethdlc hdlc2 ppp-link
sethdlc hdlc3 ppp-link
ifconfig hdlc2 down
ifconfig hdlc3 down

## Multilink configuration
pppctl add ppp0 bcp
pppctl set ppp0 link multilink
pppctl link ppp0 add 1 serial hdlc0
pppctl link ppp0 add 2 serial hdlc1
pppctl link ppp0 add 3 serial hdlc2
pppctl link ppp0 add 4 serial hdlc3
ifconfig hdlc0 up
ifconfig hdlc1 up
ifconfig hdlc2 up
ifconfig hdlc3 up

ifconfig ppp0 up mtu 1600

## bridge configuration
brctl addbr br1
brctl addif br1 eth0
brctl addif br1 ppp0

## interface br1 configuration
ifconfig br1 up 192.168.1.1
```

muxd.conf

```
controller SHDSL 1/0
    clocking-scheme synchronous
    modulation TC-PAM 64
    no framing
    line-rate 12672
    no line-probing
!
controller SHDSL 1/1
```

```

        clocking-scheme synchronous
        modulation TC-PAM 64
        no framing
        line-rate 12672
        no line-probing
!
controller SHDSL 2/0
        clocking-scheme synchronous
        modulation TC-PAM 64
        no framing
        line-rate 12672
        no line-probing
!
controller SHDSL 2/1
        clocking-scheme synchronous
        modulation TC-PAM 64
        no framing
        line-rate 12672
        no line-probing
!
interface HDLC 0 SHDSL 1/0
interface HDLC 1 SHDSL 1/1
interface HDLC 2 SHDSL 2/0
interface HDLC 3 SHDSL 2/1

```

Настройка устройства MM-222-2

netconfig.sh

```

#!/bin/sh

## device name
hostname router-m

## interface Ethernet 0 configuration
ifconfig eth0 up mtu 1600

## interface HDLC 0 configuration
sethdlc hdlc0 ppp-link
sethdlc hdlc1 ppp-link
ifconfig hdlc0 down
ifconfig hdlc1 down

sethdlc hdlc2 ppp-link
sethdlc hdlc3 ppp-link
ifconfig hdlc2 down
ifconfig hdlc3 down

## Multilink configuration
pppctl add ppp0 bcp
pppctl set ppp0 link multilink
pppctl link ppp0 add 1 serial hdlc0
pppctl link ppp0 add 2 serial hdlc1
pppctl link ppp0 add 3 serial hdlc2
pppctl link ppp0 add 4 serial hdlc3
ifconfig hdlc0 up
ifconfig hdlc1 up
ifconfig hdlc2 up
ifconfig hdlc3 up

ifconfig ppp0 up mtu 1600

## bridge configuration
brctl addbr br1

```

```
brctl addif br1 eth0
brctl addif br1 ppp0

## interface br1 configuration
ifconfig br1 up 192.168.1.2
```

muxd.conf

```
controller SHDSL 1/0
    core-function NTU slave
    clocking-scheme synchronous
    no framing
    no line-probing
!
controller SHDSL 1/1
    core-function NTU slave
    clocking-scheme synchronous
    no framing
    no line-probing
!
controller SHDSL 2/0
    core-function NTU slave
    clocking-scheme synchronous
    no framing
    no line-probing
!
controller SHDSL 2/1
    core-function NTU slave
    clocking-scheme synchronous
    no framing
    no line-probing
!
network-clock-select 1 SHDSL 2/0
network-clock-select 1 SHDSL 2/1
network-clock-select 2 SHDSL 1/0
network-clock-select 2 SHDSL 1/1
!
interface HDLC 0 SHDSL 1/0
interface HDLC 1 SHDSL 1/1
interface HDLC 2 SHDSL 2/0
interface HDLC 3 SHDSL 2/1
```

11 Статическая маршрутизация

route

Назначение:

Редактирование таблицы статической маршрутизации.

Синтаксис:

```
route {add | del} {-net | -host} {IP} {netmask NM} [gw GW] [metric N] [dev IF]
```

Параметр	Описание
add	Добавление маршрута
del	Удаление маршрута
-net	Указание маршрута к сети
-host	Указание маршрута к определенному узлу
IP	IP-адрес
netmask	Маска подсети
gw	Шлюз
metric	Метрика маршрута, 0 – непосредственно соединенная сеть или узел
dev	Интерфейс отправки пакетов в сеть назначения

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **route** позволяет добавить или удалить маршрут.

Пример:

Добавление маршрута в сеть 172.16.3.0/24.

```
router#route add -net 172.16.3.0 netmask 255.255.255.0 gw 172.16.1.1
```

12 Настройка синхронизации (NTP)

ntp

Назначение:

Установка расписания синхронизации времени с NTP-сервером.

Синтаксис:

```
ntp "{minute} {hour} {day} {month} {week_day}" [-t <sec>] {server} [server2, ...]
```

Параметр	Описание
minute	Минута (0 – 59)
hour	Час (0 - 23)
day	День (1 - 31)
month	Месяц (1 - 12)
week_day	День недели (0 - 7) (воскресенье =0 или =7)
-t	Таймаут для NTP-запроса
server, server2	Имя NTP-сервера

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ntp** позволяет задать расписание синхронизации времени с NTP-сервером. Символ * указывает любое значение.

Пример:

Выполнять синхронизацию с NTP-сервером каждый день в 7 часов 30 минут:

```
router#ntp "30 7 * * *" 192.168.0.105
 2 Jan 22:34:43 ntpdate[292]: step time server 192.168.0.105 offset 313683914.865582
sec
Ntpdate successfully scheduled
```

ntpdate

Назначение:

Установка даты и времени посредством запроса к NTP-серверу.

Синтаксис:

```
ntpdate [-46bBdqsv] [-a key#] [-e delay] [-k file] [-p samples] [-o version#] [-r rate] [-t timeo] [-U
username] [server]
```

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ntpdate** позволяет установить дату и время, отправив запрос NTP-серверу.

13 Протоколирование событий (Syslog)

syslogd

Назначение:

Настройка протоколирования событий.

Синтаксис:

syslogd [-n] [-O FILE] [-I n] [-S] [-y] [-N] [-s SIZE] [-b NUM] [-R HOST[:PORT]] [-L] [-C[size(KiB)]]

Параметр	Описание
-n	Отмена автоматического перехода в фоновый режим
-O FILE	Протоколирование в указанный файл (по-умолчанию используется файл /var/log/messages)
-l	Настройка уровня сообщений, подлежащих протоколированию (0 - 8)
-S	Протоколирование сообщений в краткой форме
-y	Вывод даты с указанием года в журнал событий
-N	Нумерация сообщений
-s SIZE	Размер файла журнала событий, перед тем, как будет создан новый файл (по-умолчанию 200 Кбайт)
-b NUM	Количество файлов журнала событий (по-умолчанию 1, максимально 99)
-R HOST[:PORT]	IP-адрес и порт удаленного Syslog-сервера (по-умолчанию используется UDP-порт 514)
-L	Протоколирование событий локально и на удаленный Syslog-сервер
-C	Протоколирование в буфер

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

-l 7.

Описание:

Настройку протоколирования событий необходимо проводить в соответствии с вышеуказанным синтаксисом в файле syslog.conf. Изменения, внесенные в файл, вступают в силу после перезагрузки устройства.

В syslog.conf параметр -l указывает уровень, начиная с которого события не протоколируются. Таким образом, для того чтобы протоколировать события уровня debug (числовое значение равно 7), необходимо указать -l 8.

Редактирования файла осуществляется с помощью редактора nano.

Пример:

Настройка протоколирования событий, начиная с уровня informational локально и на удаленный Syslog-сервер:

```
router#cat syslog.conf
syslogd -l 7 -R 192.168.0.105 -L
```

13.1 Фильтрация данных

13.2 Фильтр кадров Ethernet (ebtables)

ebtables

Назначение:

Инструмент администрирования фильтра кадров Ethernet.

Синтаксис:

ebtables [-A | D | I] {chain} {rule-specification} [options]

ebtables {-P} {chain} {target}

ebtables [-L| F | Z] [chain]

ebtables [-N | X] [chain]

ebtables {-E} {old-chain-name} {new-chain-name}

Параметр	Описание
-A --append	Добавление цепочки.
-D --delete	Удаление соответствующего правила из цепочки / Удаление правила с определённым номером из цепочки.
-I --insert	Добавление в цепочку правила с определённым номером (по умолчанию 1) в цепочке.
-L --list	Список правил в цепочке или во всех цепочках
-F --flush	Удаление всех правил в цепочке или во всех цепочках
-Z --zero	Обнуление счётчиков в цепочке или во всех цепочках
-N --new	Создание новой пользовательской цепочки
-X --delete-chain	Удаление пользовательской цепочки
-P --policy	Изменение политики цепочки
-E --rename-chain	Изменение имени цепочки
Опции	Описание
-p --proto	Протокол: номер или название
-s --src	MAC-адрес источника
-d --dst	MAC-адрес назначения
-i --in-if	Имя входящего сетевого интерфейса
-o --out-if	Имя исходящего сетевого интерфейса
--logical-in	Имя логического входящего интерфейса bridge
--logical-out	Имя логического исходящего интерфейса bridge
-M --modprobe	Вставка модулей, используя данную команду
-c --set-counters	Установка счетчиков
-v --version	Вывод версии

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **ebtables** используется для установки, настройки и просмотра таблиц правил фильтрации кадров Ethernet.

Каждая таблица содержит несколько предопределённых цепочек и может содержать цепочки, определённые пользователем. Каждая цепочка - это список правил, которые могут воздействовать на множество кадров. Каждое правило определяет, какие действие произвести с кадрами, на которые оно действует. Эти действия называются целью.

Пример:

```
router#ebtables -L
Bridge table: filter
```

```
Bridge chain: INPUT, entries: 0, policy: ACCEPT
```

```
Bridge chain: FORWARD, entries: 0, policy: ACCEPT
```

```
Bridge chain: OUTPUT, entries: 0, policy: ACCEPT
```

13.3 Фильтр пакетов IPv4 и NAT (iptables)

iptables

Назначение:

Инструмент администрирования фильтра пакетов IPv4 и NAT.

Синтаксис:

iptables [-A | -D] {chain} {rule-specification} [options]

iptables {-I} {chain} [rule-num] {rule-specification} [options]

iptables {-R} {chain} {rule-num} {rule-specification} [options]

iptables {-D} {chain} {rule-num} [options]

iptables [-L | S] [chain [rule-num]] [options]

iptables [-F | Z] [chain] [options]

iptables [-N] | [-X] {chain}

iptables {-E} {old-chain-name} {new-chain-name}

iptables {-P} {chain} {target} [options]

Параметр	Описание
-A --append	Добавление цепочки.
-D --delete	Удаление соответствующего правила из цепочки / Удаление правила с определённым номером из цепочки.
-I --insert	Добавление в цепочку правила с определённым номером (по умолчанию 1) в цепочке.
-R --replace	Заменить правило с определённым номером в цепочке.
-L --list	Список правил в цепочке или во всех цепочках
-S --list-rules	Печать правил в цепочке или во всех цепочках
-F --flush	Удаление всех правил в цепочке или во всех цепочках
-Z --zero	Обнуление счётчиков в цепочке или во всех цепочках
-N --new	Создание новой пользовательской цепочки
-X --delete-chain	Удаление пользовательской цепочки
-P --policy	Изменение политики цепочки
-E --rename-chain	Изменение имени цепочки
Опции	Описание
-p --proto	Протокол: номер или название, например, 'tcp'
-s --source	Источник спецификации
-d --destination	Назначение спецификации
-i --in-interface	Имя сетевого интерфейса
-j --jump	Задание правила
-g --goto	Перейти к цепочке без возвращения
-m --match	Расширенный поиск совпадений
-n --numeric	Цифровой выход адресов и портов
-o --out-interface	Имя выходного интерфейса
-t --table	Таблица для изменения (по умолчанию: 'filter')
-v --verbose	Расширенный режим
-x --exact	Расширенные номера (отображаются точные значения)
-f --fragment	Совпадение только второго и последующих фрагментов
--modprobe	Вставка модулей, используя команды
--set-counters	Установка счетчика во время вставки/добавления
-v --version	Вывод версии

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **iptables** используется для установки, настройки и просмотра таблиц правил фильтрации IP-пакетов.

Каждая таблица содержит несколько предопределённых цепочек и может содержать цепочки, определённые пользователем. Каждая цепочка - это список правил, которые могут воздействовать на множество пакетов. Каждое правило определяет, какие действие произвести с пакетами, на которые оно действует. Эти действия называются целью, целью может быть и переход на другую (определённую пользователем) цепочку в этой же таблице.

Пример:

Добавление новой цепочки с именем ZELAX:

```
router#iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy DROP)
target     prot opt source                destination
LAN_router#iptables -N ZELAX
LAN_router#iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy DROP)
target     prot opt source                destination

Chain ZELAX (0 references)
target     prot opt source                destination
```

iptables-restore

Назначение:

Восстановление таблицы.

Синтаксис:

iptables-restore [options] < input_file

iptables-restore [-c] [-n] [-v] [-T] [-t tablename] [-m command] [-h]

Параметр	Описание
-c	Восстановление значений всех счетчиков байтов и пакетов
-n	Не удаляет предыдущее содержимое таблицы. Если этот флаг не указан, iptables-restore сбрасывает всё предыдущее содержимое соответствующих таблиц IP
-v	Расширенный режим
-T	Не совершает ввода, происходит только тестирование
-t	Восстанавливает только одну таблицу. Если данный параметр не задан, то восстанавливает все таблицы.
-m	Позволяет добавлять необходимые модули (цели, совпадения и т.п.) во время добавления/просмотра правил в цепочке
tablename	Имя таблицы
command	Имя добавляемого модуля

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **iptables-restore** используется для восстановления IP таблиц на основе данных из стандартного ввода. Чтобы считать информацию из файла, требуется использовать предоставляемое командной оболочкой перенаправление ввода/вывода.

Пример:

Вывод справки по команде iptables-restore:

```
router#iptables-restore -h
iptables-restore 1.4.6
Restore IP Tables from data specified on STDIN.
Use I/O redirection to restore from a file:
    iptables-restore [options] < input_file

Options: iptables-restore [-c] [-n] [-v] [-T] [-t tablename] [-m command] [-h]
    -c, --counters
Restore the values of all packet and byte counters
    -n, --noflush
Don't flush the previous contents of the table. If not specified,
iptables-restore flushes (deletes) all previous contents
of the respective IP Table.
    -v, --verbose
Verbose output.
    -T, --test
Do not commit input, just testing.
    -t, --table tablename
Restrict restore to only one table. If not specified, restore
includes all available tables.
    -m, --modprobe command
When adding or inserting rules into a chain, use command
to load any necessary modules (targets, match extensions, etc).
    -h, --help
Give this help.
See also
    iptables-save, iptables
The iptables-HOWTO, which details more iptables usage, the
NAT-HOWTO, which details NAT, and the netfilter-hacking-HOWTO which
details the internals.
```

Iptables-save

Назначение:

Сохранение таблицы IP.

Синтаксис:

```
iptables-save [options] > output_file
```

```
iptables-save [-c] [-t tablename] [-m command] [-h]
```

Параметр	Описание
-c	Восстановление значений всех счетчиков байтов и пакетов
-t	Сохранение только одной таблицы. Если данный параметр не задан, то сохраняются все таблицы

-m	Позволяет добавлять необходимые модули (цели, совпадения и т.п.) во время добавления/просмотра правил в цепочке
tablename	Имя таблицы
command	Имя добавляемого модуля
-h	Помощь

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **iptables-save** используется для вывода содержимого IP таблиц (в легко обрабатываемом формате) на стандартный вывод. Также, с помощью данной команды, можно сохранить таблицу в файл.

Пример:

Восстановление значений счётчиков и их вывод с помощью команды iptables-save:

```
router#iptables-save -c

# Generated by iptables-save v1.4.6 on Sat Jan  8 02:20:55 2000
*raw
:PREROUTING ACCEPT [9021:4380086]
:OUTPUT ACCEPT [7890:254042]
COMMIT
# Completed on Sat Jan  8 02:20:55 2000
# Generated by iptables-save v1.4.6 on Sat Jan  8 02:20:55 2000
*nat
:PREROUTING ACCEPT [226:19511]
:POSTROUTING ACCEPT [7:560]
:OUTPUT ACCEPT [7:560]
COMMIT
# Completed on Sat Jan  8 02:20:55 2000
# Generated by iptables-save v1.4.6 on Sat Jan  8 02:20:55 2000
*mangle
:PREROUTING ACCEPT [9021:4380086]
:INPUT ACCEPT [8986:4376871]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [7890:254042]
:POSTROUTING ACCEPT [7890:254042]
COMMIT
# Completed on Sat Jan  8 02:20:55 2000
# Generated by iptables-save v1.4.6 on Sat Jan  8 02:20:55 2000
*filter
:INPUT ACCEPT [8986:4376871]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [7890:254042]
COMMIT
# Completed on Sat Jan  8 02:20:55 2000
```

14 Настройка протокола DHCP

udhcpс

Назначение:

DHCP клиент.

Синтаксис:

udhcpс [-C] | [-f] | [-b] [-n] | [-q] | [-t] | [-v] [-c CID] [-V VCLS] [-H HOSTNAME] [-i interface] [-p pidfile] [-r IP] [-s script]

Параметр	Описание
-C	Скрытие идентификатора по умолчанию у клиента
-f	Запуск в приоритетном режиме
-b	Запуск в фоновом режиме, если сразу не получены параметры
-n	Выход с сообщением об ошибке, если сразу не получены параметры
-q	Выход после получения параметров
-t	Посыл N пакетов на запрос
-v	Версия
-c	Установка идентификатора клиента
-V	Установка идентификатора класса поставщика
-H -h	Имя хоста клиента
-i	Используемый интерфейс (по умолчанию eth0)
-p	Сохранение ID процесса демона в файл
-r	Запрос IP адреса
-s	Запуск файла DHCP событий (по умолчанию: /etc/udhcpс/simple.script)
-S	Логирование в syslog
-R	Освободить IP адрес после выхода

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **udhcpс** настраивает параметры DHCP клиента на устройстве.

Пример:

Запуск DHCP клиента на интерфейсе eth0:

```
router#udhcpс -i eth0
```

udhcpd

Назначение:

DHCP сервер.

Синтаксис:

udhcpd [-f] [-S] [configfile]

Параметр	Описание
-f	Запуск в приоритетном режиме
-S	Логирование в syslog
Configfile	Файл конфигурации DHCP-сервера.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **udhcpd** запускает DHCP сервер на устройстве. Параметры DHCP сервера хранятся в файле `/etc/config/udhcpd.conf`

Пример:

Запуск DHCP сервера в приоритетном режиме:

```
router#udhcpd -f /etc/config/udhcpd.conf
```

Параметры DHCP сервера:

Параметры DHCP сервера хранятся в файле `udhcpd.conf`. Для изменения параметров необходимо открыть файл с помощью тестового редактора `nano` командой:

`nano udhcpd.conf`

Для задания значения параметра, отличным от значения по умолчанию, необходимо удалить символ комментария (`#`) в начале строки и установить необходимо значение. Список доступных параметров приведен в таблице:

Параметр	Описание
<code>start</code>	Первый ip-адрес DHCP пула. Значение по умолчанию 192.168.0.20.
<code>end</code>	Последний ip-адрес DHCP пула. Значение по умолчанию 192.168.0.254.
<code>interface</code>	Интерфейс, на котором запускается DHCP сервер. Значение по умолчанию <code>eth0</code> .
<code>max_leases</code>	Максимальное количество ip-адресов для выдачи. Значение по умолчанию 254.
<code>remaining</code>	Параметр позволяющий хранить в файле <code>udhcpd.leases</code> относительное время до окончания аренды (значение «yes»). Если установлено значение «no», то в файле <code>udhcpd.leases</code> хранится абсолютное время окончания аренды. Значение по умолчанию «yes».
<code>auto_time</code>	Период времени в секундах, через который в файл <code>udhcpd.leases</code> заносится информация о выданных ip-адресах. Значение по умолчанию 7200 (2 часа).
<code>decline_time</code>	Период времени в секундах, через который освобождается выданный ip-адрес, если маршрутизатор получил сообщение DHCP decline. Значение по умолчанию 3600 (1 час).
<code>conflict_time</code>	Период времени в секундах, через который освобождается выданный ip-адрес, если маршрутизатор зафиксировал ARP конфликт. Значение по умолчанию 3600 (1 час).
<code>offer_time</code>	Период времени в секундах, на который резервируется ip-адрес, после отправки DHCP offer и до получения DHCP request от клиента. Значение по умолчанию 60 (1 минута).
<code>lease_file</code>	Файл, в котором хранится информация о выданных ip-адресах. Необходимо установить значение: <code>/etc/config/udhcpd.leases</code>

dumpleases

Назначение:

Вывод списка арендованных IP-адресов.

Синтаксис:

`dumpleases [-r | -a] [-f configfile]`

Параметр	Описание
<code>-r</code>	Интерпретировать время аренды, как оставшееся время
<code>-a</code>	Интерпретировать время аренды, как прошедшее время
<code>configfile</code>	Файл, содержащий список арендованных адресов

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует.

Описание:

Команда **dumpleases** выводит список арендованных IP-адресов, выданных udhcpd. Информация о выданных ip-адресах хранится в файле /etc/config/udhcpd.leases.

Пример:

```
router-m#dumpleases -f udhcpd.leases
Mac Address      IP-Address      Expires in
00:d0:69:12:34:56 192.168.1.208  -1179695041 seconds
```

15 Настройка протокола SNMP

Для мониторинга устройства по протоколу SNMP необходимо настроить и запустить процесс `snmpd`. Настройки данного процесса содержатся в двух файлах:

1. `/etc/snmpd.conf` — системные настройки необходимые для работы процесса `snmpd`, файл не доступен для редактирования.
2. `/etc/config/snmpd.local.conf` — пользовательские настройки.

Для запуска процесса `snmpd` необходимо выполнить команду `snmpd -c /etc/snmpd.conf`. Файл `/etc/snmpd.conf` содержит ссылку на файл `/etc/config/snmpd.local.conf`, после запуска процесса сначала применяются системные настройки, а затем пользовательские.

Настройки в файл `snmpd.local.conf` записываются в виде отдельный строк. Возможные варианты параметров:

Параметр	Описание
<code>rocommunity COMMUNITY</code>	Имя <code>community</code> для чтения SNMP параметров
<code>trapcommunity COMMUNITY</code>	Имя <code>community</code> для отправки SNMP-trap сообщений
<code>trapsink HOST</code>	IP-адрес сервера для отправки SNMP-trap сообщений
<code>linkUpDownNotifications yes</code>	Включение отправки SNMP-trap об изменении состояний интерфейсов
<code>sysLocation STRING</code>	Текущее местонахождение устройство
<code>sysContact STRING</code>	Контактная информация администратора данного устройства

В текущей версии ПО поддерживаются следующие функции SNMP:

- чтение статистики и состояний интерфейсов
- чтение статистики и состояний контроллеров
- чтение статистики по протоколам IP, TCP, UDP, SNMP
- отправка SNMP-trap сообщений о состоянии интерфейсов и контроллеров

Пример:

Пример настройки SNMP параметров:

```
router#cat snmpd.local.conf
rocommunity public

trapcommunity public
trapsink 192.168.0.111

linkUpDownNotifications yes

sysLocation techsupport
sysContact +74957487187
```

16 Настройка качества обслуживания (QoS)

Настройка качества обслуживания (QoS) проводится в режиме Linux Shell. Команды для настройки качества обслуживания необходимо записать в файл `netconfig.sh`. Редактирование файла `netconfig.sh` доступно в текстовом редакторе `nano`.

Качество обслуживания (управление трафиком) представляет собой комплекс систем обработки очередей и функций приема/передачи пакетов. Данный комплекс включает в себя механизмы принятия решений:

- какие пакеты принимать;
- какие пакеты передавать;
- в каком порядке передавать пакеты;
- с какой скоростью передавать пакеты.

16.1.1 Элементы управления исходящим трафиком.

Ограничение исходящего трафика (`shaping`) – механизм, с помощью которого пакеты задерживаются перед передачей для того, чтобы скорость передачи соответствовала желаемой. Дополнительно данный механизм позволяет сглаживать взрывообразный трафик.

Планирование – алгоритм обработки пакетов в очереди, который позволяет упорядочить и переупорядочить пакеты. Примерами планировщиков могут быть алгоритмы FIFO, SFQ, TBF и другие.

Классификация – механизм, разделяющий пакеты разных типов для различной обработки. Классификация может выполняться на каждом промежуточном узле либо на границе сети с единым администрированием. В последнем случае в пакет устанавливается определенный маркер, на основе которого работают планировщики на всем пути прохождения пакета. Обычно в качестве маркера используются поля IP Precedence и DSCP (ToS). Общей рекомендацией является в сети с единым администрированием классифицировать пакеты как можно ближе к их источнику.

16.1.2 Механизмы управления исходящим трафиком.

Ограничение исходящего трафика реализуется с помощью классов (`class`). Класс представляет собой группу пакетов с одинаковым способом обработки.

Планирование обработки пакетов осуществляют дисциплины (`qdisc`). Планирование может быть простым без деления пакетов на классы (бесклассовые дисциплины) и сложным с делением пакетов на классы (классовые дисциплины).

Бесклассовые дисциплины обработки очереди используются, если необходимо ко всем пакетам проходящим через интерфейс применить одинаковый алгоритм планирования. Бесклассовые дисциплины не могут содержать классы и к ним нельзя подключать фильтры. При использовании бесклассовых дисциплин классификация не имеет смысла.

В случае, когда тот или иной вид трафика необходимо обрабатывать по-разному, необходимо использовать классовые дисциплины обработки очереди. Когда пакет передается на обработку классовой дисциплине обработки очереди, он должен быть отнесен к одному из классов (классифицирован).

Классификация выполняется фильтрами (`filter`) с помощью классификаторов. Примерами классификаторов могут служить `u32`, `fw` и другие. Фильтры, присоединенные к дисциплине, возвращают результат классификации (класс пакета), после чего пакет передается в очередь, соответствующую заданному классу. Каждый из классов, в свою очередь, может состоять из подклассов и иметь свой набор фильтров, для выполнения более точной классификации своей доли трафика.

Для классификации того или иного пакета, всякий раз вызывается так называемая "цепочка классификации". Эта цепочка состоит из всех фильтров, присоединенных к полноклассовой дисциплине. Когда пакет необходимо поставить в очередь, проверяется каждая ветвь в цепочке фильтров.

Классы существуют только внутри классовых дисциплин обработки очереди. Класс может содержать несколько подклассов или дисциплину обработки очереди. К каждому классу может быть подключено произвольное количество фильтров, которые позволяют распределять по подклассам, переквалифицировать и уничтожать трафик, проходящий через данный класс.

Любой класс, содержащий подклассы, называется внутренним (или корневым, если нет родительского класса). Краевой класс – завершающий класс дисциплины. Краевой класс не может содержать подклассы, только бесклассовую дисциплину обработки очереди.

Каждый класс и классовая дисциплина требуют уникального идентификатора (handle) в конструкциях управления трафиком. Этот идентификатор называется дескриптором и состоит из двух чисел: старшего и младшего номера. Формат дескриптора: <старший> : <младший>. В качестве номеров могут использоваться произвольные числа, заданные в соответствии со следующими правилами:

- Старший номер – совершенно произвольное число, соответствующее планируемой схеме нумерации. Все объекты структуры управления трафиком, у которых есть общий родитель, должны иметь общий старший номер дескриптора.
- Младший номер равен нулю для дисциплин обработки очереди. Значения отличные от нуля обозначают классы. Все классы с общим родителем должны иметь уникальный младший номер дескриптора.

Дескриптор используется в качестве аргумента ключей classid и flowid при создании фильтров.

Схема управления исходящим трафиком в случае использования классовых дисциплин обработки представлена на рисунке 1.

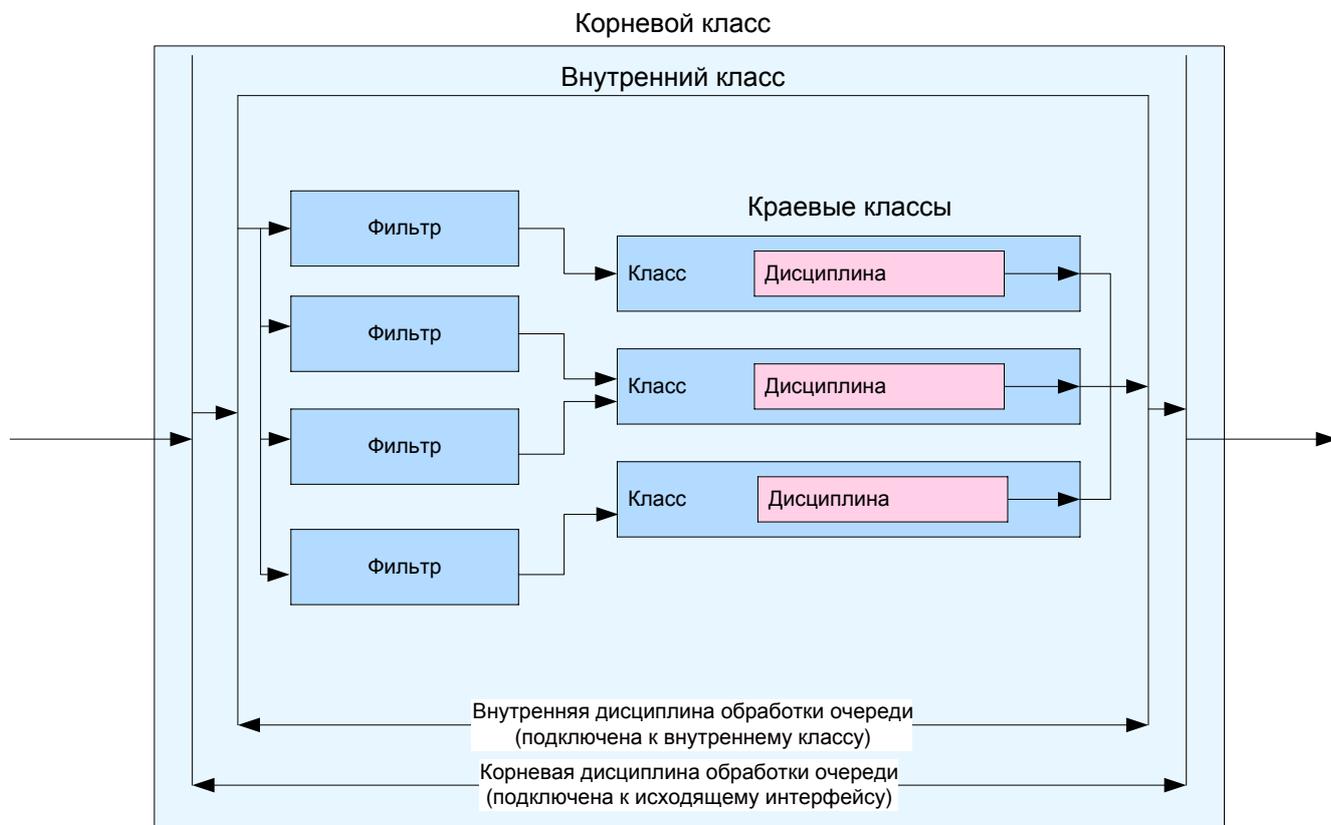


Рисунок 1. Схема управления исходящим трафиком.

tc

Назначение:

Изменение/мониторинг параметров системы управления трафиком.

Синтаксис:

tc [qdisc | class | filter] [commands] – изменение/настройка параметров системы управления трафиком

tc [-s] [-d] [qdisc | class | filter] show [commands] – мониторинг текущего состояния системы управления трафиком

Параметр	Описание
qdisc	Управление дисциплинами, мониторинг состояния.
class	Управление классами, мониторинг состояния.
filter	Управление фильтрами, мониторинг состояния.
-s	Отображать текущую статистику.
-d	Отображать детали состояния.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc** выполняет все необходимые действия при работе с системой управления трафиком. В качестве первого аргумента (не ключа) утилиты ожидается один из трех компонентов системы управления трафиком: дисциплину (**qdisc**), класс (**class**) или фильтр (**filter**). Для каждого типа объекта существует свой набор опций. Возможные опции подробно рассмотрены в соответствующих разделах далее.

16.2 Бесклассовые дисциплины обработки очереди

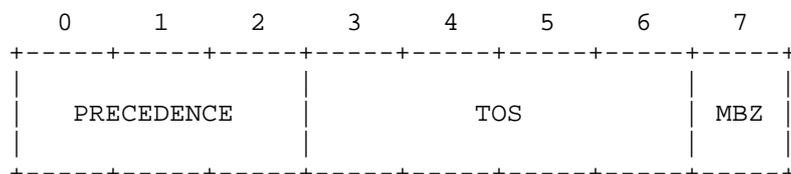
16.2.1 Дисциплина pfifo_fast

Дисциплина **pfifo_fast** является наиболее простой и включена на интерфейсах процессора по умолчанию. Параметры обработки пакетов в очереди заданы в ядре операционной системы. Изменения параметров процедуры обработки пакетов в дисциплине **pfifo_fast** невозможно.

Исходящая очередь разделяется на три полосы. В каждой полосе пакеты обрабатываются по принципу «первый пришел, первый вышел» («first in, first out» - FIFO). Но полоса 1 не будет обслуживаться до тех пор, пока есть пакеты в полосе 0. Аналогично, пока есть пакеты в полосе 1, не обрабатывается полоса 2. Распределение пакетов в ту или иную очередь происходит на основе поля ToS.

Дисциплина **pfifo_fast** является наиболее простым решением **для передачи трафика с различными метками ToS**, т.к. не требует настройки. Для правильной приоритезации пакетов необходимо их классифицировать на входящем интерфейсе с помощью метки ToS. Однако не рекомендуется использовать эту дисциплину обработки очереди для передачи большого объема данных, т.к. данная дисциплина не имеет возможности ограничивать скорость исходящего трафика.

Поле ToS имеет следующий формат:



Четыре бита TOS определяются так:

Двоичное значение	Десятичное значение	Описание
1000	8	Минимальная задержка (md)
0100	4	Максимальная пропускная способность (mt)
0010	2	Максимальная надежность (mr)
0001	1	Минимальная стоимость (mmc)
0000	0	Обычное обслуживание

Так как в поле ToS справа имеется бит MBZ, реальное значение поля ToS в два раза больше значения TOS. Ядро распределяет пакет с установленной меткой ToS в соответствии с приоритетом обслуживания.

Ниже в таблице приведено соответствие реального значения поля ToS, класса обслуживания Linux и полосы очереди pfifo_fast.

ToS	Описание	Приоритет Linux	Полоса pfifo_fast
0x0	Обычное обслуживание	0 Best Effort	1
0x2	Минимальная стоимость	1 Filler	2
0x4	Максимальная надежность	0 Best Effort	1
0x6	nmc+mr	0 Best Effort	1
0x8	Максимальная пропускная способность	2 Bulk	2
0xa	nmc+mt	2 Bulk	2
0xc	mr+mt	2 Bulk	2
0xe	nmc+mr+mt	2 Bulk	2
0x10	Минимальная задержка	6 Interactive	0
0x12	nmc+md	6 Interactive	0
0x14	mr+md	6 Interactive	0
0x16	nmc+mr+md	6 Interactive	0
0x18	mt+md	4 Int. Bulk	1
0x1a	nmc+mt+md	4 Int. Bulk	1
0x1c	mr+mt+md	4 Int. Bulk	1
0x1e	nmc+mr+mt+md	4 Int. Bulk	1

ifconfig txqueuelen

Назначение:

Регулирует длину исходящей очереди на интерфейсе

Синтаксис:

ifconfig [interface] **txqueuelen** NN

Параметр	Описание
interface	Интерфейс, для которого настраивается размер очереди (eth0, hdlc0 и т.п.)
NN	Размер исходящей очереди в пакетах.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Для интерфейса eth0 250 пакетов, для интерфейсов hdlc – 256 пакетов.

Описание:

Команда **txqueuelen** позволяет изменять размер исходящей очереди на интерфейсе

Пример:

```
router#ifconfig eth0 txqueuelen 10
```

```

router#ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:1A:81:00:35:E1
          inet addr:172.16.1.156  Bcast:172.16.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:259537 errors:0 dropped:0 overruns:0 frame:0
          TX packets:245585 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:16869869 (16.0 MiB)  TX bytes:13273534 (12.6 MiB)
          Base address:0xe00

```

16.2.2 Дисциплина Token Bucket Filter (TBF)

Token Bucket Filter (TBF) простая дисциплина обработки очереди, которая передает поступающие пакеты со скоростью, не превышающей административно заданный порог.

TBF является наиболее простым решением **для ограничения исходящей скорости на порту**.

Реализована TBF в виде буфера, постоянно заполняющегося токенами с заданой скоростью. Каждый прибывающий токен соответствует байту. При передаче пакета определенного размера из буфера удаляется соответствующее количество токенов. В зависимости от скорости поступления токенов и данных возможна одна из следующих ситуаций:

- Данные поступают со скоростью равной скорости поступления токенов. В этом случае каждый пакет имеет соответствующий токен и проходит очередь без задержки.
- Данные поступают со скоростью меньшей, чем скорость поступления токенов. В этом случае лишь часть существующих токенов уничтожается, оставшиеся токены накапливаются до размера буфера. Далее накопленные токены могут использоваться при всплесках для передачи данных со скоростью, превышающей скорость пребывающих токенов.
- Данные поступают быстрее, чем токены. Это означает, что в буфере скоро не останется токенов, что заставит дисциплину приостановить передачу данных. Эта ситуация называется "превышением". Если пакеты продолжают поступать, они начинают уничтожаться.

tc qdisc tbf

Назначение:

Задаёт дисциплину обработки очереди tbf на интерфейсе.

Синтаксис:

```

tc qdisc [add | del | change | show ] dev [interface] root tbf rate [rate][kbit] limit [bytes] burst [bytes]
[mtu [bytes]] [peakrate [kbps]] [latency [time][ms]]

```

Параметр	Описание
add	Добавление дисциплины обработки исходящей очереди на интерфейс. Если на интерфейсе уже задана дисциплина обработки исходящей очереди, отличная от дисциплины по умолчанию, то для изменения типа дисциплины необходимо сначала установить тип дисциплины <code>pfifo_fast</code> , удалив текущую дисциплину, а затем создать новую дисциплину обработки исходящей очереди..
del	Удаление текущей дисциплины обработки очереди, установка дисциплины по умолчанию – <code>pfifo_fast</code>
change	Изменение текущих параметров дисциплины обработки очереди.
show	Вывод текущих параметров дисциплины обработки очереди.
dev	Интерфейс, на котором настраивается дисциплина обработки исходящей очереди.
rate	Ограничение скорости на порту. Числовое значение в битах/с (по умолчанию) или килобитах/с (ключевое слово kbit).
limit	Количество байт, которые могут быть помещены в очередь ожидания токенов.

	Числовое значение.
burst	Размер буфера в байтах. Числовое значение.
mtu	Максимальное значение в байтах для передачи через очередь без фрагментации.
peakrate	Скорость прохождения очереди. Максимальное значение 1 Мбит/с. Для установки более высокой скорости используйте параметр mtu.
latency	Максимальное время нахождения пакета в очереди ожидания токенов. Числовое значение в микросекундах (по умолчанию) и миллисекундах (ключевое слово ms). Аналогично параметру limit. При настройке необходимо задать один из параметров: limit или latency. Расчет второго параметра будет произведен автоматически.

Ядро автоматически рассчитывает ближайшие к настроенным параметры для указанной скорости передачи.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc qdisc tbf** позволяет изменять дисциплину обработки исходящей очереди на интерфейсе.

Пример:

Настройка ограничения исходящей скорости на порту Ethernet до 2Мбит/с :

```
router#tc qdisc add dev eth0 root tbf rate 2000kbit latency 50ms burst 2000
router#tc qdisc show dev eth0
qdisc tbf 800a: root rate 2000Kbit burst 1999b lat 50ms
```

16.2.3 Дисциплина Stochastic Fairness Queueing (SFQ)

Дисциплина SFQ работает с потоками данных. Поток данных представляет собой пакеты с одинаковыми параметрами адреса источника и назначения, типа протокола и номера порта TCP/UDP. Для упрощения можно считать, что поток приблизительно соответствует сеансу TCP или потоку UDP. Трафик делится на достаточное количество очередей типа FIFO, по одной на каждый поток. После этого, все очереди обрабатываются в циклическом порядке, тем самым обеспечивая каждому сеансу равные шансы на передачу данных.

Дисциплина SFQ является наиболее простым решением **для выделения примерно одинаковой полосы пропускания на каждый сеанс TCP или поток UDP**. Дисциплина SFQ **эффективна, только если исходящий интерфейс полностью загружен**. В противном случае очередь будет отсутствовать и, следовательно, никакого положительного эффекта наблюдаться не будет.

Распределение потоков в очереди происходит на основе хеш-алгоритма. Из-за использования хеша, несколько сессий могут попасть в одну и ту же очередь, что уменьшает шансы на передачу каждого сеанса. Для того, чтобы эта проблема не ощущалась, SFQ часто меняет алгоритм хеширования, поэтому, если сессии и попадут в одну очередь, длиться это будет лишь несколько секунд.

tc qdisc sfq

Назначение:

Задаёт дисциплину обработки очереди sfq на интерфейсе.

Синтаксис:

tc qdisc [add | del | change | show] dev [interface] root sfq [limit [number]] [perturb [seconds]] [quantum [bytes]]

Параметр	Описание
add	Добавление дисциплины обработки исходящей очереди на интерфейс. Если на интерфейсе уже задана дисциплина обработки исходящей очереди, отличная от дисциплины по умолчанию, то для изменения типа дисциплины необходимо сначала установить тип дисциплины <code>pfifo_fast</code> , удалив текущую дисциплину, а затем создать новую дисциплину обработки исходящей очереди..
del	Удаление текущей дисциплины обработки очереди, установка дисциплины по умолчанию – <code>pfifo_fast</code>
change	Изменение текущих параметров дисциплины обработки очереди.
show	Вывод текущих параметров дисциплины обработки очереди.
dev	Интерфейс, на котором настраивается дисциплина обработки исходящей очереди.
limit	Общее количество пакетов, которые могут быть помещены в очередь. При превышении заданного количества пакетов все поступающие на интерфейс пакеты удаляются.
perturb	Интервал изменения алгоритма хеширования. Если данный параметр не задан, алгоритм хеширования не меняется. Крайне нежелательно оставлять этот параметр незадаанным.
quantum	Количество байт, выводящихся из очереди за один раз. По умолчанию данный параметр равен MTU на интерфейсе. Нежелательно устанавливать данный параметр меньше размера MTU.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc qdisc sfq** позволяет изменять дисциплину обработки исходящей очереди на интерфейсе.

Пример:

Настройка разделения полосы пропускания между всеми пользователями:

```
router#tc qdisc add dev eth0 root sfq perturb 10
router#tc qdisc show dev eth0
qdisc sfq 800a: root limit 127p quantum 1500b perturb 10sec
```

16.2.4 Дисциплина Traffic Equalizer (TEQL)

Дисциплина `teql` используется для распределения нагрузки между несколькими интерфейсами. При этом для `teql` не является обязательным совпадения типа интерфейса и скорости. Для распределения нагрузки между интерфейсами необходимо добавить данную дисциплину на каждый интерфейс. После этого появится виртуальный интерфейс `teql`. Данный виртуальный интерфейс представляет собой объединение всех интерфейсов, на которые была добавлена дисциплина `teql`. Работа с виртуальным интерфейсом не отличается от работы с обычными интерфейсами маршрутизатора. Распределение пакетов по интерфейсам происходит по алгоритму `round-robin`, т.е. пакеты отправляются по очереди в каждый интерфейс.

tc qdisc teql

Назначение:

Задаёт дисциплину обработки очереди `teql` на интерфейсе.

Синтаксис:

tc qdisc [add | del | change | show] dev [interface] root teql[number]

Параметр	Описание
add	Добавление дисциплины обработки исходящей очереди на интерфейс. Если на интерфейсе уже задана дисциплина обработки исходящей очереди, отличная от дисциплины по умолчанию, то для изменения типа дисциплины необходимо сначала установить тип дисциплины <code>pfifo_fast</code> , удалив текущую дисциплину, а затем создать новую дисциплину обработки исходящей очереди..
del	Удаление текущей дисциплины обработки очереди, установка дисциплины по умолчанию – <code>pfifo_fast</code>
change	Изменение текущих параметров дисциплины обработки очереди.
show	Вывод текущих параметров дисциплины обработки очереди.
dev	Интерфейс, на котором настраивается дисциплина обработки исходящей очереди.
teql	Виртуальный интерфейс <code>teql</code> . Имя виртуального интерфейса записывается в виде <code>teql[номер-интерфейса]</code> , например <code>teql0</code> .

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc qdisc teql** позволяет создавать виртуальный интерфейс `teql` и включать в него физические интерфейсы для распределения нагрузки.

Пример:

Распределение нагрузки между двумя потоками E1:

`netconfig.sh`

```
## interface HDLC 0 configuration
sethdlc hdlc0 cisco
sethdlc hdlc1 cisco
ifconfig hdlc0 10.0.0.2 netmask 255.255.255.0 up
ifconfig hdlc1 10.0.1.2 netmask 255.255.255.0 up
```

```
## interface TEQL 0 configuration
tc qdisc add dev hdlc0 root teql0
tc qdisc add dev hdlc1 root teql0
ifconfig teql0 up
```

`muxd.conf`

```
controller E1 2/0
    framing pcm31
!
controller E1 2/1
    framing pcm31
!
interface HDLC 0 E1 2/0
interface HDLC 1 E1 2/1
```

`router#ifconfig`

```
hdlc0    Link encap:cisco-hdlc  (keepalive 10)
         inet addr:10.0.0.2  P-t-P:10.0.0.2  Mask:255.255.255.0
         UP POINTOPOINT NOARP  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:256
```

```
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
hdlc1 Link encap:cisco-hdlc (keepalive 10)
inet addr:10.0.1.2 P-t-P:10.0.1.2 Mask:255.255.255.0
UP POINTOPOINT NOARP MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:256
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
teq10 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
UP POINTOPOINT RUNNING NOARP MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

16.3 Классовые дисциплины обработки очереди

16.3.1 Создание классов

tc class

Назначение:

Создает класс или редактирует параметры класса.

Синтаксис:

```
tc class [add | del | change | replace | show ] dev [interface] classid [classid] [ parent [ parent ] | root ]
[qdisk] [параметры qdisk]
```

Параметр	Описание
add	Добавление класса.
del	Удаление класса
change	Изменение текущих параметров класса.
replace	Замена текущего класса.
show	Вывод текущих параметров класса.
dev	Интерфейс, к которому присоединяется класс.
classid	Идентификатор класса в формате <старший> : <младший>. Младший идентификатор должен быть отличным от нуля.
parent	Идентификатор родительского класса. Если родительский класс является корневым, указывается ключевое слово root .
qdisk	Дисциплина обработки очереди внутри класса. Данный параметр необходимо настраивать только для корневого и внутренних классов. Используются классовые дисциплины очереди. Параметры классовых дисциплин рассмотрены отдельно в разделе 12.3.2. Для краевых классов используются бесклассовые дисциплины обработки, которые задаются с помощью команды tc qdisc. Исключение составляет дисциплина prio. Параметры настройки дисциплины prio рассмотрены в разделе 12.3.2.1.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc class** позволяет создавать классы для классификации пакетов. Изменять параметры классов и классовых дисциплин обработки очереди.

tc qdisc parent

Назначение:

Задаёт параметры дисциплины обработки очереди для краевого класса.

Синтаксис:

tc qdisc [add | del | change | replace | show] dev [interface] [parent [parent]] [handle [handle]] [qdisc] [параметры qdisc]

Параметр	Описание
add	Добавление класса.
del	Удаление класса
change	Изменение текущих параметров класса.
replace	Замена текущего класса.
show	Вывод текущих параметров класса.
dev	Интерфейс, к которому присоединяется класс.
parent	Идентификатор родительского класса. В формате <старший>:<младший>.
handle	Идентификатор дисциплины в формате <число>:
qdisc	Дисциплина обработки очереди внутри краевого класса. Бесклассовая дисциплина обработки очереди внутри краевого класса. Параметры для каждой дисциплины рассмотрены в разделе 12.2.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

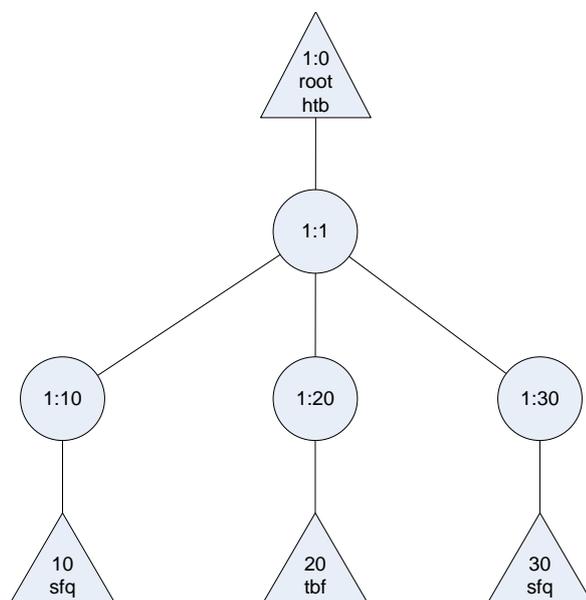
Отсутствует

Описание:

Команда **tc qdisc parent** задаёт дисциплину обработки очереди для краевого класса. В случае разбиения трафика на классы в команде появляется обязательный параметр для обозначения родительского класса **parent**.

Пример:

Создание иерархии классов и дисциплин. На диаграмме треугольниками показаны дисциплины, кругами – классы. Ниже приведены команды для реализации данной схемы:



```

router#tc qdisc add dev eth0 root handle 1: htb default 30
router#tc class add dev eth0 parent 1: classid 1:1 htb rate 20mbit

router#tc class add dev eth0 parent 1:1 classid 1:10 htb rate 3mbit ceil 5mbit
router#tc class add dev eth0 parent 1:1 classid 1:20 htb rate 10mbit ceil
15mbit
router#tc class add dev eth0 parent 1:1 classid 1:30 htb rate 3mbit ceil 5mbit

router#tc qdisc add dev eth0 parent 1:10 handle 10: sfq perturb 10
router#tc qdisc add dev eth0 parent 1:20 handle 20: tbf rate 10000kbit burst
1604 latency 50ms
router#tc qdisc add dev eth0 parent 1:30 handle 30: sfq perturb 10

```

16.3.2 Задание классовых дисциплин обработки очереди

16.3.2.1 Дисциплина PRIO

Дисциплина PRIO никак не ограничивает трафик, она лишь выполняет его классификацию на основе присоединенных к ней фильтров. Данная дисциплина может быть рассмотрена как более функциональная версия дисциплины `pfifo_fast`, в которой каждая из полос является отдельным классом, а не простой очередью FIFO.

По умолчанию дисциплина PRIO создает три класса с дисциплиной FIFO. Дисциплина FIFO может быть заменена дисциплиной любого доступного типа.

При извлечении пакета из очереди сначала проверяется класс :1, после класс :2 и в последнюю очередь класс :3. При этом класс :2 не будет обслуживаться, пока есть пакеты в классе :1. Аналогично, пока есть пакеты в классе :2, не обрабатывается класс :3.

Дисциплина PRIO применяется, **если необходимо разделить трафик по приоритетам, основываясь не только на значении поля ToS**. Не рекомендуется использовать дисциплину PRIO, когда интерфейс полностью загружен, так как данная дисциплина не имеет возможности ограничивать скорость исходящего трафика.

tc qdisc prio

Назначение:

Задаёт классовую дисциплину обработки очереди `prio` на интерфейсе.

Синтаксис:

```
tc qdisc [add | del | change | show ] dev [interface] [ parent [parentid] | root ] [ handle [handle] ] prio [
bands [bands] ] [ priomap [priomap] ]
```

Параметр	Описание
add	Добавление дисциплины обработки исходящей очереди на интерфейс.
del	Удаление текущей дисциплины обработки очереди.
change	Изменение текущих параметров дисциплины обработки очереди.
show	Вывод текущих параметров дисциплины обработки очереди.
dev	Интерфейс, на котором настраивается исходящая очередь.
parent	Идентификатор родительского класса. Если создается корневая дисциплина используется ключевое слово root . Обязательный параметр.
handle	Идентификатор дисциплины в формате <число>:
burst	Размер буфера в байтах. Числовое значение.
bands	Количество создаваемых подклассов. По умолчанию создается три класса.
priomap	Карта приоритетов трафика в соответствии с меткой ToS. По умолчанию карта приоритетов аналогична карте приоритетов для дисциплины <code>pfifo_fast</code> .

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc qdisc prio** позволяет задавать классовую дисциплину обработки исходящей очереди на интерфейсе.

Пример:

Задание очереди prio:

```
router#tc qdisc add dev eth0 root handle 1: prio
##По умолчанию создается 3 класса: 1:1,1:2 и 1:3
router#tc -s class show dev eth0
class prio 1:1 parent 1:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
class prio 1:2 parent 1:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
class prio 1:3 parent 1:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
router#tc qdisc add dev eth0 parent 1:1 handle 10: prio
router#tc -s class show dev eth0
class prio 1:1 parent 1: leaf 10:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
class prio 1:2 parent 1:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
class prio 1:3 parent 1:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
class prio 10:1 parent 10:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
class prio 10:2 parent 10:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
class prio 10:3 parent 10:
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
```

16.3.2.2 Дисциплина Class Based Queueing (CBQ)

CBQ - одна из самых сложных дисциплин обработки очереди. Данная дисциплина является классовой и имеет возможность **ограничения трафика(shaping)**. Ограничение трафика осуществляется на основе времени простоя канала, которое вычисляется на основании среднего размера пакета и полосы пропускания.

Если ограничение трафика настраивается с помощью дисциплины CBQ, необходимо учитывать, что CBQ не определяет объем трафика, а измеряет среднее время простоя канала (интервал в микросекундах между запросами на передачу) и на основе полученного времени рассчитывает среднюю загрузку канала. Таким образом, если с помощью дисциплины CBQ ограничить канал пропускной способностью 10 Мбит/с величиной в 1 Мбит/с, время простоя канала увеличится до 90% общего времени работы.

Для определения среднего времени простоя (**avgidle**) рассчитывается эффективное время простоя (как экспоненциальное взвешенное по скользящему окну) и расчетное время простоя (на основе заданной общей пропускной способности канала). Расчетное время простоя вычитается из эффективного времени

простоя, в результате получается величина **avgidle**. Полностью загруженный канал имеет величину **avgidle** равную нулю, т.е. промежуток времени между передачей пакетов точно совпадает с расчетным. В случае превышения заданного ограничения пропускной способности канала, величина **avgidle** становится отрицательной. Если превышение достигает некоторого порога, CBQ приостанавливает передачу на время, пока **avgidle** не увеличится достаточно для передачи следующего пакета. Во избежание длительных периодов ожидания, величина **avgidle** ограничена снизу параметром **minidle**.

С другой стороны, после нескольких часов простоя, величина **avgidle** может стать слишком большой, что приведет к попыткам передачи слишком большого объема трафика в дальнейшем. Чтобы этого не произошло, величина **avgidle** ограничивается сверху параметром **maxidle**. Параметры **maxidle** и **minidle** нельзя задать напрямую. Только косвенно с помощью параметров **maxburst** и **minburst**.

tc qdisc cbq

Назначение:

Задаёт классовую корневую дисциплину обработки очереди cbq на интерфейсе.

Синтаксис:

```
tc qdisc [add | del | change | show ] dev [interface] root [ handle [handle] ] cbq [ bandwidth [bandwidth_bps] ] [ avpkt [avpkt-bytes] ] [ mpu [mpu] ] [ cell [cell] ]
```

Параметр	Описание
add	Добавление дисциплины обработки исходящей очереди на интерфейс.
del	Удаление текущей дисциплины обработки очереди.
change	Изменение текущих параметров дисциплины обработки очереди.
show	Вывод текущих параметров дисциплины обработки очереди.
dev	Интерфейс, на котором настраивается исходящая очередь.
root	Ключевое слово, указывающее на то, что данная дисциплина является корневой.
handle	Идентификатор дисциплины в формате <число>:
bandwidth	Физическая пропускная способность интерфейса. Данный параметр используется для вычисления расчетного времени простоя. Обязательный параметр.
avpkt	Усредненный размер одного пакета. Используется для вычисления параметра maxidle .
mpu	Минимальный размер пакета в байтах. Данный параметр необходимо указывать для правильного вычисления времени простоя, поскольку передача даже пустого пакета требует определенной затраты времени.
cell	Размер шага, с которым увеличивается расчетное время передачи. Например, если указано значение 8, то расчетное время передачи для пакетов размером 800 и 806 байт будет одинаковым, а для пакета 810 байт несколько большим. Рекомендованное значение 8. Значение cell должно быть степенью числа 2.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc qdisc cbq** позволяет задавать классовую корневую дисциплину обработки исходящей очереди на интерфейсе.

tc class cbq

Назначение:

Создает класс cbq или редактирует параметры класса cbq.

Синтаксис:

tc class [**add** | **del** | **change** | **replace** | **show**] **dev** [interface] **classid** [classid] [**parent** [parentid] | **root**] **cbq** [**bandwidth** [bandwidth-bps]] [**rate** [rate-bps]] [**maxburst** [maxburst-pkts]] [**minburst** [minburst-pkts]] [**bounded**] [**isolated**] [**allot** [allot-bytes]] [**mpu** [mpu-bytes]] [**weight** [weight]] [**prio** [prio-number]] [**cell** [cell]] [**ewma** [ewma-number]] [**split** [classid]] [**defmap** [defmap-hex]]

Параметр	Описание
add	Добавление класса.
del	Удаление класса
change	Изменение текущих параметров класса.
replace	Замена текущего класса.
show	Вывод текущих параметров класса.
dev	Интерфейс, к которому присоединяется класс.
classid	Идентификатор класса в формате <старший> : <младший>. Младший идентификатор должен быть отличным от нуля.
parent	Идентификатор родительского класса. Если родительский класс является корневым, указывается ключевое слово root .
bandwidth	Физическая пропускная способность интерфейса. Данный параметр используется для вычисления расчетного времени простоя. Обязательный параметр.
rate	Максимальная скорость для этого класса и всех его дочерних классов. Обязательный параметр.
maxburst	Параметр используется для вычисления значения maxidle . Задаёт количество усредненных пакетов, которые будут обработаны до того, как значение avgidle уменьшится до нуля. Параметр maxidle нельзя задать напрямую, только косвенно, с помощью этого параметра.
minburst	Параметр используется для вычисления значения minidle . Задаёт количество пакетов, которые будут простаивать в очереди до того, как значение avgidle возрастет до нуля. Параметр minidle нельзя задать напрямую, только косвенно, с помощью этого параметра.
bounded	Параметр, запрещающий классу занимать пропускную способность у других классов. Если данный параметр не задан, используется значение borrow , позволяющее классу занимать пропускную способность у других классов.
isolated	Параметр, указывающий, что у данного класса нельзя занимать пропускную способность, даже если она свободна. Если данный параметр не задан, используется значение sharing , позволяющее занимать пропускную способность у данного класса, если она свободна.
allot	Количество байт, которые могут быть переданы из класса за один опрос очередью.
mpu	Минимальный размер пакета в байтах. Данный параметр необходимо указывать для правильного вычисления времени простоя, поскольку передача даже пустого пакета требует определенной затраты времени.
weight	Вес. Дисциплина СВQ вычисляет сумму весов всех классов и затем нормирует их полученной величиной, поэтому в качестве весов можно использовать любые значения, важно соблюдать отношение этих значений. Рекомендованное значения веса определяется как (полоса пропускания класса)*0,1. Для определения количества данных, посылаемых классом за раз, нормированное значение умножается на величину allot .
prio	Приоритет класса. В первую очередь обрабатываются данные из классов с более высоким приоритетом. Чем выше значение, тем меньше приоритет. Обязательный параметр.
cell	Размер шага, с которым увеличивается расчетное время передачи. Например, если указано значение 8, то расчетное время передачи для пакетов размером 800 и 806 байт будет одинаковым, а для пакета 810 байт несколько большим. Рекомендованное значение 8. Значение cell должно быть степенью числа 2.
ewma	Степень сглаживания экспоненциального скользящего среднего. Число от 0 до 32. Значение по умолчанию 5. Чем меньше степень сглаживания, тем алгоритм более чувствителен к изменению времени простоя.
split	Если фильтры класса не вынесли решение, классификация может быть проведена

defmap	на основе метки ToS. Доступно 16 значений ToS (от 0 до 15). Параметр split определяет, из какого класса или дисциплины передается трафик с заданными приоритетами в класс указанный в параметре classid . Параметр defmap определяет какой класс получает пакеты каких приоритетов. Соответствие приоритетов определяется битовой таблицей. Значение defmap задается в виде двузначного шестнадцатиричного числа. Например, параметры classid 10:1 split 10:0 defmap c0 определяют передачу пакетов с метками ToS 6 и 7 классу 10:1, параметры classid 10:2 split 10:0 defmap 3f определяют передачу пакетов с метками ToS 0,1,2,3,4,5 в класс 10:2.
--------	--

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc class cbq** позволяет создавать классы с дисциплиной обработки очереди cbq. Изменять параметры классов и дисциплины cbq.

Пример:

Реализация следующей схемы:

Используется 2 класса трафика. Полоса пропускания трафика из класса 1:3 ограничена 5 Мбит/с, полоса пропускания трафика из класса 1:4 ограничена 3 Мбит/с, суммарная полоса пропускание не превышает 6 Мбит/с. Классы могут занимать пропускную полосу друг у друга. Используется интерфейс Ethernet 10 Мбит/с.

```
##создание корневой дисциплины
router#tc qdisc add dev eth0 root handle 1:0 cbq bandwidth 10Mbit \
> avpkt 1000 cell 8
```

##создание родительского класса с общей пропускной способностью 6 Мбит/с и запретом занимать пропускную способность.

```
router#tc class add dev eth0 parent 1:0 classid 1:1 cbq bandwidth 10Mbit \
> rate 6Mbit weight 0.6Mbit prio 8 allot 1514 cell 8 maxburst 20 \
> avpkt 1000 bounded
```

```
##создание дочернего класса с ограничением пропускной способности 5 Мбит/с
router#tc class add dev eth0 parent 1:1 classid 1:3 cbq bandwidth 10Mbit \
> rate 5Mbit weight 0.5Mbit prio 5 allot 1514 cell 8 maxburst 20 \
> avpkt 1000
```

```
##создание дочернего класса с ограничением пропускной способности 3 Мбит/с
router#tc class add dev eth0 parent 1:1 classid 1:4 cbq bandwidth 10Mbit \
> rate 3Mbit weight 0.3Mbit prio 5 allot 1514 cell 8 maxburst 20 \
> avpkt 1000
```

```
router#tc -s class show dev eth0
class cbq 1: root rate 10000Kbit (bounded,isolated) prio no-transmit
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  rate 0bit 0pps backlog 0b 0p requeues 0
  borrowed 0 overactions 0 avgidle 78 undertime 0
class cbq 1:1 parent 1: rate 6000Kbit (bounded) prio no-transmit
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  rate 0bit 0pps backlog 0b 0p requeues 0
  borrowed 0 overactions 0 avgidle 14782 undertime 0
class cbq 1:3 parent 1:1 rate 5000Kbit prio 5
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  rate 0bit 0pps backlog 0b 0p requeues 0
```

```

borrowed 0 overactions 0 avgidle 42130 undertime 0
class cbq 1:4 parent 1:1 rate 3000Kbit prio 5
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
rate 0bit 0pps backlog 0b 0p requeues 0
borrowed 0 overactions 0 avgidle 71696 undertime

```

16.3.2.3 Дисциплина Hierarchical Token Bucket (HTB)

Дисциплина HTB реализуют мощную иерархию классов. Важной составляющей дисциплины HTB является механизм распределения пропускной способности. HTB **позволяет установить гарантированную пропускную способность и максимальную пропускную способность для класса**. Гарантированная пропускная способность определяется параметром **rate**. При максимальной загрузке канала данные класса будут передаваться со скоростью, указанной в параметре **rate**. Если канал загружен неполностью и объем данных класса превышает значение **rate**, подклассы могут использовать часть пропускной способности своих родительских классов. Максимальное значение пропускной способности подкласса определяется параметром **ceil**.

Объекты ограничения HTB базируются на TBF. Также в HTB **существует возможность устанавливать приоритеты классов**. Наиболее простое применение данной дисциплины – **ограничение скорости для разных видов трафика**.

Реализована HTB в виде набора буферов, постоянно заполняющихся токенами с заданной скоростью. Каждый прибывающий токен соответствует байту. При передаче пакета определенного размера из буфера удаляется соответствующее количество токенов.

Процесс ограничения полосы пропускания выполняется в краевых классах. Внутренние классы предназначены для реализации механизма распределения пропускной способности между классами.

Дисциплина HTB позволяет создавать классы двух типов: внутренние и краевые. В зависимости от задействованной пропускной способности и типа класса возможны следующие ситуации:

Тип класса	Состояние класса	Внутреннее состояние HTB	Поведение
Краевой	< rate	HTB_CAN_SEND	Краевой класс будет передавать данные, пока есть доступные токены
Краевой	> rate, < ceil	HTB_MAY_BORROW	Краевой класс попытается занять токены для передачи данных у родительского класса.
Краевой	> ceil	HTB_CANT_SEND	Пакеты не будут передаваться до тех пор, пока не будет достаточного количества токенов.
Внутренний, корневой	< rate	HTB_CAN_SEND	Внутренний класс будет передавать токены своим подклассам.
Внутренний, корневой	> rate, < ceil	HTB_MAY_BORROW	Внутренний класс попытается занять токены у родительского класса и передать их своим подклассам.
Внутренний, корневой	> ceil	HTB_CANT_SEND	Внутренний класс не будет пытаться занять токены у родительского класса и не будет передавать токены своим подклассам.

tc qdisc htb

Назначение:

Задаёт классовую корневую дисциплину обработки очереди htb на интерфейсе.

Синтаксис:

tc qdisc [add | del | change | show] dev [interface] root [handle [handle]] htb [default [default_handle]]

Параметр	Описание
add	Добавление дисциплины обработки исходящей очереди на интерфейс.
del	Удаление текущей дисциплины обработки очереди.
change	Изменение текущих параметров дисциплины обработки очереди.
show	Вывод текущих параметров дисциплины обработки очереди.
dev	Интерфейс, на котором настраивается исходящая очередь.
root	Ключевое слово, указывающее на то, что данная дисциплина является корневой.
handle	Идентификатор дисциплины в формате <число>:
default_handle	Идентификатор дисциплины, которой будет передан пакет в случае отсутствия совпадения в фильтрах класса, в формате <число>

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc qdisc htb** позволяет задавать классовую корневую дисциплину обработки исходящей очереди на интерфейсе.

tc class htb

Назначение:

Создает класс htb или редактирует параметры класса htb.

Синтаксис:

tc class [add | del | change | replace | show] dev [interface] classid [classid] [parent [parentid] | root] htb [prio [prio-number]] [rate [rate-bps]] [ceil [ceil-bps]] [burst [burst-bytes]] [cburst [cburst-bytes]]

Параметр	Описание
add	Добавление класса.
del	Удаление класса
change	Изменение текущих параметров класса.
replace	Замена текущего класса.
show	Вывод текущих параметров класса.
dev	Интерфейс, к которому присоединяется класс.
classid	Идентификатор класса в формате <старший> : <младший>. Младший идентификатор должен быть отличным от нуля.
parent	Идентификатор родительского класса. Если родительский класс является корневым, указывается ключевое слово root .
prio	Приоритет класса. Задается в виде целого положительного числа. Чем меньше значение, тем выше приоритет. Классы с меньшим приоритетом не обрабатываются, пока есть данные в классах с большим приоритетом. Обязательный параметр.
rate	Гарантированная пропускная способность для данного класса. Обязательный параметр. Поскольку HTB ограничивает скорость только в краевых классах, сумма значений параметров rate краевых классов не должна превышать значения ceil родительского класса. Лучше всего, когда эта сумма равна параметру rate родительского класса.
ceil	Максимально возможная пропускная способность для данного класса. По умолчанию устанавливается равной параметру rate , т.е. по умолчанию классу запрещено использовать пропускную способность родительского класса.
burst	Размер буфера токенов в байтах. Данный параметр определяет максимальное количество байт, для которого могут быть доступны токены в один момент времени при передаче на скорости, определенной параметром ceil . Для родительского класса данный параметр должен быть не меньше, чем для дочерних классов. Не

	устанавливайте данный параметр меньше значения MTU.
cburst	Размер буфера токенов в байтах. Данный параметр определяет максимальное количество байт, для которого могут быть доступны токены в один момент времени при передаче на скорости, определенной скоростью интерфейса. Для родительского класса данный параметр должен быть не меньше, чем для дочерних классов. Не устанавливайте данный параметр меньше значения MTU.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc class htb** позволяет создавать классы с дисциплиной обработки очереди htb. Изменять параметры классов и дисциплины htb.

Пример:

Реализация следующей схемы:

Используется 3 класса трафика. Гарантированная пропускная способность класса 1:10 2 Мбит/с. Гарантированная пропускная способность класса 1:20 3 Мбит/с. Гарантированная пропускная способность класса 1:30 1 Мбит/с. Классы могут занимать пропускную способность друг у друга. Суммарная пропускная способность классов не может превышать 6 Мбит/с.

```
##создание корневой дисциплины
router# tc qdisc add dev eth0 root handle 1: htb default 30

##создание родительского класса с общей пропускной способностью 6 Мбит/с
router# tc class add dev eth0 parent 1: classid 1:1 htb rate 6mbit

##создание дочернего класса с ограничением пропускной способности 2 Мбит/с
router# tc class add dev eth0 parent 1:1 classid 1:10 htb rate 2mbit ceil
6mbit

##создание дочернего класса с ограничением пропускной способности 3 Мбит/с
router# tc class add dev eth0 parent 1:1 classid 1:20 htb rate 3mbit ceil
6mbit

##создание дочернего класса с ограничением пропускной способности 1 Мбит/с
router# tc class add dev eth0 parent 1:1 classid 1:30 htb rate 1mbit ceil
6mbit

##задание дисциплины обработки очереди sfq для краевых классов
router# tc qdisc add dev eth0 parent 1:10 handle 10: sfq perturb 10
router# tc qdisc add dev eth0 parent 1:20 handle 20: sfq perturb 10
router# tc qdisc add dev eth0 parent 1:30 handle 30: sfq perturb 10

router#tc -s class show dev eth0
class htb 1:10 parent 1:1 leaf 10: prio 0 rate 2000Kbit ceil 6000Kbit burst
2599b cburst 4599b
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  rate 0bit 0pps backlog 0b 0p requeues 0
  lended: 0 borrowed: 0 giants: 0
  tokens: 10156 ctokens: 5989

class htb 1:1 root rate 6000Kbit ceil 6000Kbit burst 4599b cburst 4599b
  Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  rate 0bit 0pps backlog 0b 0p requeues 0
  lended: 0 borrowed: 0 giants: 0
```

```
tokens: 5989 ctokens: 5989
```

```
class htb 1:20 parent 1:1 leaf 20: prio 0 rate 3000Kbit ceil 6000Kbit burst
3099b cburst 4599b
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
rate 0bit 0pps backlog 0b 0p requeues 0
lended: 0 borrowed: 0 giants: 0
tokens: 8072 ctokens: 5989
```

```
class htb 1:30 parent 1:1 leaf 30: prio 0 rate 1000Kbit ceil 6000Kbit burst
2099b cburst 4599b
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
rate 0bit 0pps backlog 0b 0p requeues 0
lended: 0 borrowed: 0 giants: 0
tokens: 16406 ctokens:
```

16.3.3 Создание фильтров

tc filter

Назначение:

Задаёт или редактирует фильтр.

Синтаксис:

```
tc filter [add | del | change | replace | show ] dev [interface] [ prio [priority] ] [ protocol [protocol] ] [
parent [ parent ] | root ] handle [filter_id] [filter_type] [ параметры filter_type ] flowid flow-id
```

Параметр	Описание
add	Добавление фильтра.
del	Удаление текущей фильтра
change	Изменение текущих параметров фильтра.
replace	Замена текущего фильтра. Если фильтр не существует, он будет создан.
show	Вывод текущих параметров фильтра.
dev	Интерфейс, на котором настраивается фильтр.
prio	Приоритет фильтра. Чем выше число, тем ниже приоритет. Фильтры разных протоколов не могут иметь одинаковый приоритет. Приоритет, указанный в фильтре указывает на то, с каким приоритетом будет использоваться трафик в этом фильтре. Т.е. чем ниже цифра - тем приоритетнее трафик из этого фильтра в очереди.
protocol	Протокол, который будет обрабатывать фильтр. Например, ip, ipv6, arp.
parent	Идентификатор класса, к которому присоединен фильтр. Если необходимо присоединить фильтр к корневому классу (на вход интерфейса) используется ключевое слово root .
handle	Идентификатор фильтра. Если не задан явно, присваивается автоматически начиная с 800.
filter_type	Вид фильтра. Допустимые значения: rsvp, u32, fw, route и т.д. В зависимости от выбранного типа фильтра, используются разные параметры.
flowid	Идентификатор дисциплины, к которой будет отнесен пакет, в случае, если выполняются условия фильтра.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc filter** позволяет создать фильтр для классификации пакетов.

16.3.3.1 Общее описание фильтров

Утилита **tc** позволяет использовать различные типы фильтров для классификации пакетов. Их краткое описание приведено в таблице ниже. Наиболее распространенным является фильтр **u32**, т.к. он позволяет проводить сравнение по любому полю данных в пакете.

Фильтр	Описание
fw	Фильтр, принимающий решение на основе метки в пакете, установленной iptables .
u32	Фильтр, принимающий решение на основе значения заданного поля данных в заголовке пакета.

16.3.3.2 Фильтр fw

Фильтр **fw** позволяет классифицировать пакеты на основе поля **MARK** (меток **netfilter**). Значение поля **MARK** изменяется командой **iptables -j MARK**.

iptables -j MARK

Назначение:

Используется для установки меток для определенных пакетов. Это действие может выполняться только в пределах таблицы *mangle*. Метка пакета существует только в промежутки времени, пока пакет не покинул маршрутизатор, т.е. метка не передается по сети. Если необходимо пометить пакет так, чтобы использовать маркировку на других устройствах, устанавливается метка **ToS**.

Синтаксис:

```
iptables -A [ PREROUTING | POSTROUTING | INPUT | OUTPUT | FORWARD ] -t mangle -i [interface] -j MARK --set-mark [number]
```

Параметр	Описание
- A --append	Добавление цепочки.
PREROUTING	Цепочка, которая используется для внесения изменений на входе в маршрутизатор, перед принятием решения о маршрутизации.
POSTROUTING	Цепочка, которая используется для внесения изменений на выходе из маршрутизатора, после принятия решения о маршрутизации.
INPUT	Цепочка, которая используется для внесения изменений в пакеты перед тем, как они будут переданы локальному приложению внутри маршрутизатора.
OUTPUT	Цепочка, которая используется для внесения изменений в пакеты, поступающие от приложений внутри маршрутизатора.
FORWARD	Цепочка, которая используется для внесения изменений в транзитные пакеты после первого принятия решения о маршрутизации, но перед последним принятием решения о маршрутизации.
-t	Таблица для изменения.
mangle	Таблица, которая используется для внесения изменений в заголовки пакетов. Примером может служить изменение поля TTL , TOS или MARK . Важно: в действительности поле MARK не изменяется, но в памяти ядра заводится структура, которая сопровождает данный пакет все время его прохождения через маршрутизатор, так что другие правила и приложения на данном маршрутизаторе (и только на данном маршрутизаторе) могут использовать это поле в своих целях.
number	Метка, присваиваемая пакету. Целое число в диапазоне от 1 до 255.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **iptables -j MARK** позволяет присваивать пакетам целочисленные метки. Данные метки могут быть использованы в фильтрах для определения принадлежности пакета к тому или иному классу или дисциплине. Данные метки имеют локальное значение, классификация на основе меток может быть реализована только на том маршрутизаторе, где были установлены метки.

Пример:

Установка метки 6 на все пакеты проходящие через интерфейс eth0:

```
router#iptables -A FORWARD -t mangle -i eth0 -j MARK --set-mark 6
```

tc filter fw

Назначение:

Задаёт или редактирует фильтр u32.

Синтаксис:

```
tc filter [add | del | change | replace | show ] dev [interface] [ prio [priority] ] [ protocol [protocol] ] [ parent [ parent ] ] [handle [number]] fw [ classid [class_id] ]
```

Параметр	Описание
add	Добавление фильтра.
del	Удаление текущей фильтра
change	Изменение текущих параметров фильтра.
replace	Замена текущего фильтра. Если фильтр не существует, он будет создан.
show	Вывод текущих параметров фильтра.
dev	Интерфейс, на котором настраивается фильтр.
pref	Приоритет фильтра. Чем выше число, тем ниже приоритет. Фильтры разных протоколов не могут иметь одинаковый приоритет.
protocol	Протокол, который будет обрабатывать фильтр. Например, ip, ipv6, agr.
parent	Идентификатор класса, к которому присоединен фильтр. Если необходимо присоединить фильтр к корневому классу (на вход интерфейса) используется ключевое слово root .
handle	Метка пакета
classid	Идентификатор класса, к которому будет отнесен пакет, в случае, если выполняются условия фильтра.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc filter fw** позволяет создать фильтр fw для классификации пакетов на основе меток. Необходимо учитывать, что в отличие от других типов фильтров параметр handle задаёт не идентификатор фильтра, а значение метки, по которому происходит классификация.

Пример:

Настройка фильтра для классификации пакетов с меткой 6 в класс 1:10 :

```
router#tc filter add dev eth0 pprotocol ip parent 1:0 prio 1 handle 6 fw classid 1:10
```

16.3.3.3 Фильтр U32

Фильтр U32 является наиболее распространенным фильтром для классификации пакетов. Фильтр u32 позволяет проводить сравнение по любому блоку битов в пакете. Если значение блока совпадает со значением, заданным в фильтре, над пакетом выполняется некоторое действие.

tc filter u32

Назначение:

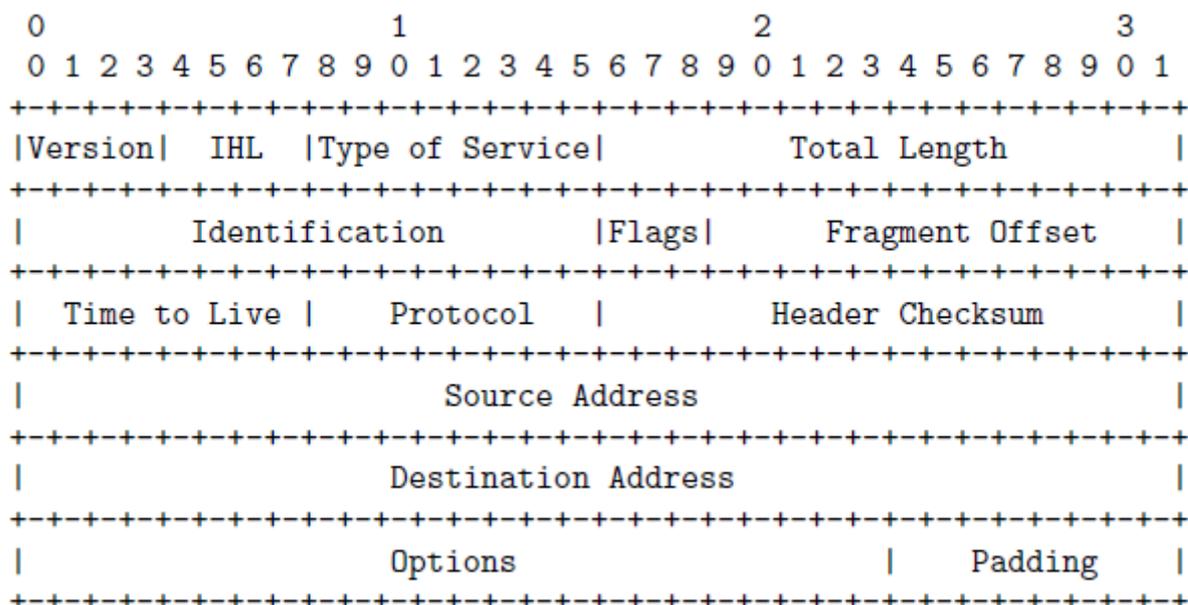
Задаёт или редактирует фильтр u32.

Синтаксис:

```
tc filter [add | del | change | replace | show ] dev [interface] [ prio [priority] ] [ protocol [protocol] ] [ parent [ parent ] | root ] [ handle [number] ] u32 [ classid [class_id] ] match [type] [value] [mask] at [offset] [ flowid [flow-id] ]
```

Параметр	Описание
add	Добавление фильтра.
del	Удаление текущей фильтра
change	Изменение текущих параметров фильтра.
replace	Замена текущего фильтра. Если фильтр не существует, он будет создан.
show	Вывод текущих параметров фильтра.
dev	Интерфейс, на котором настраивается фильтр.
pref	Приоритет фильтра. Чем выше число, тем ниже приоритет. Фильтры разных протоколов не могут иметь одинаковый приоритет.
protocol	Протокол, который будет обрабатывать фильтр. Например, ip, ipv6, arp.
parent	Идентификатор класса, к которому присоединен фильтр. Если необходимо присоединить фильтр к корневому классу (на вход интерфейса) используется ключевое слово root .
handle	Идентификатор дисциплины.
classid	Идентификатор класса, к которому будет отнесен пакет, в случае, если выполнятся условия фильтра.
type	Условие сопоставления. Размер блока данных для сопоставления в битах. Допустимые параметры: u32, u16, u8 – 32, 16, 8 бит соответственно.
value	Значение, с которым будут сравниваться биты из пакета.
mask	Битовая маска. Битовая маска накладывается на биты из пакета. Результат наложения сравнивается с заданным значением. Размер маски должен совпадать с размером блока данных для сравнения.
offset	Смещение от начала пакета до требуемого блока битов в байтах. Если значение не задано, по умолчанию рассматривает начало пакета сетевого уровня (0-вая граница). В случае необходимости сравнения в заголовке пакета канального уровня используется отрицательное смещение.
flowid	Идентификатор дисциплины, к которой будет отнесен пакет, в случае, если выполнятся условия фильтра.

Параметр смещения для ip пакета может быть определен из формата ip заголовка (RFC791):



Заголовки других протоколов доступны в соответствующих RFC.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc filter u32** позволяет создать фильтр u32 для классификации пакетов.

Пример:

Настройка фильтра для классификации пакетов из сети 192.168.8.0/24 в класс 1:8 :

```
router#tc filter add dev eth0 parent 1: protocol ip prio 1 u32 \
>classid 1:8 \
>match u32 0xc0a80800 0xffffffff00 at 12
```

Кроме стандартного формата фильтр u32 позволяет использовать для сопоставления селекторы. В качестве селекторов предлагаются наиболее распространенные поля пакета. Ниже в таблице приведены значения некоторых селекторов.

Формат использования:

match [selector]

Селектор	Поле для сравнения
ip df	Бит DF (Don't fragment / Не фрагментировать).
ip dport [value] [mask]	TCP/UDP порт назначения.
ip dst [value]/ [mask]	IP адрес назначения. В данном селекторе маска используется в формате CIDR (количество значимых бит).
ip protocol [value] [mask]	Тип протокола в IP заголовке. Для сравнения необходимо указать код протокола (например, 6 для TCP, 17 для UDP), указывать символьные имена (TCP, UDP) нельзя.
ip sport [value] [mask]	TCP/UDP порт источника.
ip src [value]/ [mask]	IP адрес источника. В данном селекторе маска используется в формате CIDR (количество значимых бит).

ip tos [value] [mask]	Поле ToS. Указывается в формате hex.
tcp dst [value] [mask]	TCP порт назначения.
tcp src [value] [mask]	TCP порт источника.
udp dst [value] [mask]	UDP порт назначения.
udp src [value] [mask]	UDP порт источника.

Пример:

Настройка фильтра для классификации пакетов из сети 192.168.8.0/24 в класс 1:8 с помощью селектора:

```
router#tc filter add dev eth0 parent 1: protocol ip prio 1 u32 \
>classid 1:8 \
>match ip src 192.168.8.0/24
```

Существует возможность создания более сложных фильтров с комбинированными условиями (операции логического «И», «ИЛИ» и их комбинаций). Логическое «И» позволяет классифицировать пакеты по значению нескольких полей. Логическое «ИЛИ» позволяет классифицировать в один класс пакеты с различным значением одного поля.

Логическое «И»

Для учета значений нескольких полей необходимо создать фильтр с несколькими полями match. При этом сопоставление будет успешно только в том случае, если все условия будут выполняться.

Пример:

Настройка фильтра для классификации пакетов из сети 192.168.8.0/24 и значением поля ToS 0x10 (интерактивный трафик) в класс 1:8 :

```
router#tc filter add dev eth0 parent 1: protocol ip prio 1 u32 \
>classid 1:8 \
>match ip src 192.168.8.0/24 \
>match ip tos 0x10 0x1e
```

Логическое «ИЛИ»

Для классификации пакетов с различными значениями одного и того же поля в один класс необходимо создать несколько фильтров с одинаковым параметром classid (flowid для дисциплины). При этом для одного протокола значения параметра prio может совпадать, для различных протоколов значения prio должны быть различны.

Пример:

Настройка фильтра для классификации пакетов из сети 192.168.8.0/24 и из сети 172.16.1.0/24 в класс 1:8:

```
router#tc filter add dev eth0 parent 1: protocol ip prio 1 u32 \
>classid 1:8 \
>match ip src 192.168.8.0/24
router#tc filter add dev eth0 parent 1: protocol ip prio 1 u32 \
>classid 1:8 \
>match ip src 172.16.1.0/24
```

16.4 Ограничение входящего трафика (policing)

Ограничение входящего трафика (policing) – элемент системы качества обслуживания, основной задачей которого является ограничение скорости входящего трафика. Этот механизм принимает пакеты до определенной скорости, а над частью трафика, превысившей заданный порог, выполняется определенное действие. Чаще всего это действие – уничтожение. Не смотря на то, что в данном случае так же, как и в случае с исходящим трафиком, используется концепция буфера токенов, для входящего трафика не поддерживается возможность планирования порядка обработки пакетов и задержки пакетов.

Входящий трафик проходят через входящую дисциплину (ingress qdisc). Возможности данной дисциплины ограничены: для данной дисциплины нельзя создавать классы, к ней можно подключить только фильтр. Ограничитель скорости входящего трафика существует только как составная часть данного фильтра.

tc qdisc ingress

Назначение:

Задаёт входящую дисциплину обработки входящей очереди на интерфейсе.

Синтаксис:

tc qdisc [add | del | change | show] dev [interface] handle ffff: ingress

Параметр	Описание
add	Добавление дисциплины обработки исходящей очереди на интерфейс.
del	Удаление текущей дисциплины обработки очереди.
change	Изменение текущих параметров дисциплины обработки очереди.
show	Вывод текущих параметров дисциплины обработки очереди.
dev	Интерфейс, на котором настраивается исходящая очередь.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc qdisc ingress** позволяет задавать входящую дисциплину обработки входящей очереди на интерфейсе.

Пример:

```
Задание очереди ingress:
router#tc qdisc add dev eth0 handle ffff: ingress
router#tc -s qdisc show
qdisc ingress ffff: dev eth0 parent ffff:fff1 -----
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
  rate 0bit 0pps backlog 0b 0p requeues 0
```

tc filter u32 police

Назначение:

Задаёт или редактирует ограничитель входящего трафика в составе фильтра u32.

Синтаксис:

tc filter [add | del | change | replace | show] dev [interface] [prio [priority]] [protocol [protocol]] parent ffff: [handle [number]] u32 match [type] [value] [mask] at [offset] action police rate [rate-bps] burst

[burst-bytes] [mtu [mtu-bytes]] [peakrate [peakrate-bps]] conform-exceed [pipe | ok | reclassify |drop]
flowid :1

Параметр	Описание
add	Добавление фильтра.
del	Удаление текущей фильтра
change	Изменение текущих параметров фильтра.
replace	Замена текущего фильтра. Если фильтр не существует, он будет создан.
show	Вывод текущих параметров фильтра.
dev	Интерфейс, на котором настраивается фильтр.
pref	Приоритет фильтра. Чем выше число, тем ниже приоритет. Фильтры разных протоколов не могут иметь одинаковый приоритет.
protocol	Протокол, который будет обрабатывать фильтр. Например, ip, ipv6, arp.
handle	Идентификатор дисциплины.
type	Условие сопоставления. Размер блока данных для сопоставления в битах. Допустимые параметры: u32, u16, u8 – 32, 16, 8 бит соответственно.
value	Значение, с которым будут сравниваться биты из пакета.
mask	Битовая маска. Битовая маска накладывается на биты из пакета. Результат наложения сравнивается с заданным значением. Размер маски должен совпадать с размером блока данных для сравнения.
offset	Смещение от начала пакета до требуемого блока битов в байтах. Если значение не задано, по умолчанию рассматривает начало пакета сетевого уровня (0-вая граница). В случае необходимости сравнения в заголовке пакета канального уровня используется отрицательное смещение.
rate	Ограничение скорости на порту. Числовое значение в битах/с (по умолчанию) или килобитах/с (ключевое слово kbit).
burst	Размер буфера в байтах (по умолчанию) или килобайтах (ключевое слово k).
mtu	Максимальный размер пакета в байтах, который может быть обработан фильтром. Пакеты большего размера уничтожаются. Значение по умолчанию 2000 байт.
peakrate	Ограничение скорости на порту, если используется 2 буфера. Числовое значение в битах/с (по умолчанию) или килобитах/с (ключевое слово kbit). Размер каждого буфера в этом случае берется равным параметру mtu . Параметр mtu должен быть задан в явном виде.
conform-exceed	Действие, выполняемой над пакетами, превышающими заданный порог. Допустимые значения: pipe , ok – пропустить пакет, reclassify - передать пакет для классификации фильтру с меньшим приоритетом, drop - уничтожить пакет.

Режим конфигурации:

Команда доступна в Linux shell.

Установка по умолчанию:

Отсутствует

Описание:

Команда **tc filter u32 police** позволяет создать ограничитель входящего трафика в составе фильтра u32.

Пример:

Настройка фильтра с ограничением скорости входящего tcp-трафика из сети 192.168.8.0/24 в сеть 192.168.9.0/24 до 2 Мбит/с :

```
router#tc filter add          \
dev eth0                     \
parent ffff:                 \
pref 10                       \
protocol ip                   \
handle ::1                    \
```

```
u32 \
match ip protocol 6 0xff \
match ip src 192.168.8.0/24 \
match ip dst 192.168.9.0/24 \
action police \
rate 2Mbit burst 200K conform-exceed drop
```

17 Настройка протокола OSPF

Перед настройкой протокола динамической маршрутизации OSPF необходимо запустить процессы ospfd и zebra с помощью команд:

1. `ospfd -P 0 -d -f /etc/config/ospfd.conf`
2. `zebra -P 0 -d -f /etc/config/zebra.conf`

Данные команды необходимо записать в файл netconfig.sh. Настройки протокола OSPF хранятся в файлах ospfd.conf и zebra.conf.

Команды для настройки протокола OSPF доступны в режиме vty shell (Quagga).

vttysh

Назначение:

Команда переводит устройство в режим конфигурирования vty shell (Quagga).

Синтаксис:

vttysh

Режим конфигурации:

Команда доступна в режиме конфигурирования Linux shell:

```
router#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **vttysh** позволяет перейти в режим командной строки Quagga. В этом режиме доступно конфигурирование протокола динамической маршрутизации OSPF. По умолчанию вход осуществляется в непривилегированный режим. В непривилегированном режиме доступен просмотр некоторых параметров конфигурации.

Для выхода из режима vty shell необходимо выполнить команду **exit**.

Пример:

```
router#vttysh
```

```
Hello, this is Quagga (version 0.99.15).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
router>
```

enable

Назначение:

Команда переводит устройство в привилегированный режим vty shell (Quagga).

Синтаксис:

enable

Режим конфигурации:

Команда доступна в режиме конфигурирования vty shell:

```
router>
```

Установка по умолчанию:

отсутствует

Описание:

Команда **enable** позволяет перейти в привилегированный режим командной строки Quagga.

Для выхода из привилегированного режима vty shell необходимо выполнить команду **disable**.

Пример:

```
router> enable
router#
```

copy running-config startup-config**Назначение:**

Команда сохраняет текущую конфигурацию

Синтаксис:

copy running-config startup-config

Режим конфигурации:

Команда доступна в привилегированном режиме конфигурирования vty shell:

```
router#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **copy running-config startup-config** позволяет сохранить текущую конфигурацию в энергонезависимую память устройства.

Пример:

```
router# copy running-config startup-config
Building Configuration...
Configuration saved to /etc/config/zebra.conf
Configuration saved to /etc/config/ospfd.conf
[OK]
```

configure terminal**Назначение:**

Команда переводит устройство в режим настройки.

Синтаксис:

configure terminal

Режим конфигурации:

Команда доступна в привилегированном режиме vty shell:

```
router#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **configure terminal** позволяет перейти в режим настройки маршрутизатора.

Для выхода из данного режима необходимо выполнить команду **exit**.

Пример:

```
router# configure terminal
router(config)#
```

link-detect

Назначение:

Команда включает отслеживание состояния работы интерфейса на физическом уровне.

Синтаксис:

link-detect

Режим конфигурации:

Команда доступна в режиме настройки интерфейса:

```
router(config-if)#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **link-detect** включает отслеживание состояния работы соответствующего интерфейса на физическом уровне. При включении данной опции в случае отказа интерфейса (например, при обрыве линии связи), устройство мгновенно перестроит таблицу маршрутизации, не дожидаясь истечения интервала `dead-interval`.

Пример:

```
router(config)# interface hdlc0
router(config-if)# link-detect
```

router ospf

Назначение:

Команда включает протокол OSPF и переводит устройство в режим настройки протокола OSPF.

Синтаксис:

router ospf

Режим конфигурации:

Команда доступна в режиме настройки маршрутизатора:

```
router(config)#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **router ospf** включает протокол OSPF на маршрутизаторе и позволяет перейти в режим настройки протокола OSPF. Сетевые устройства на основе Quagga поддерживают один экземпляр протокола OSPF.

Для выхода из данного режима необходимо выполнить команду **exit**. Для выключения протокола OSPF следует выполнить команду **no router ospf**.

Пример:

```
router(config)# router ospf
router(config-router)#
```

network

Назначение:

Включение протокола OSPF на интерфейсе.

Синтаксис:

network [ip-address]/[network-mask] **area** [area-number]

Параметр	Описание
ip-address	Ip адрес сети, в которой находится интерфейс
network-mask	Маска подсети, в которой находится интерфейс
area-number	Номер зоны OSPF. Допустимо представление номера зоны в виде десятичного числа с диапазоне 0-4294967295 и в виде ip адреса в формате A.B.C.D

Режим конфигурации:

Команда доступна в режиме настройки протокола OSPF:

```
router(config-router)#.
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **network** включает протокол OSPF на определенном сетевом интерфейсе. Указание зоны OSPF обязательно. Для выключения протокола OSPF на интерфейсе необходимо использовать команду **no network** [ip-address]/[network-mask] **area** [area-number]

Пример:

```
router(config-router)# network 1.1.1.0/24 area 0
```

router-id

Назначение:

Задает идентификатор маршрутизатора в процессе OSPF.

Синтаксис:

router-id [router-id]

Параметр	Описание
router-id	Идентификатор маршрутизатора в формате ip адреса

Режим конфигурации:

Команда доступна в режиме настройки протокола OSPF:

```
router(config-router)#.
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **router-id** задает идентификатор маршрутизатора в процессе OSPF. Идентификатор маршрутизатора отображается во всех таблицах и пакетах OSPF.

Пример:

```
router(config-router)# router-id 1.1.1.1
```

passive-interface

Назначение:

Запрещает рассылку OSPF пакетов через указанный интерфейс.

Синтаксис:

passive-interface [интерфейс | **default**]

Параметр	Описание
Интерфейс	Название интерфейса
default	Запрет рассылки OSPF пакетов через все интерфейсы, на которых включен протокол OSPF

Режим конфигурации:

Команда доступна в режиме настройки протокола OSPF:

```
router(config-router)#.
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **passive-interface** запрещает рассылку OSPF пакетов через указанный интерфейс. Данная настройка применима для сокращения объема сетевого трафика, если через указанный интерфейс не может быть установлено соседство OSPF. Для отмены режима **passive-interface** следует использовать команду **no passive-interface** [интерфейс | **default**].

Пример:

```
router(config-router)# passive-interface hdlc0
```

default-information originate**Назначение:**

Рассылает в OSPF информацию о маршруте по умолчанию.

Синтаксис:

default-information originate [always]

Параметр	Описание
always	Рассылка маршрута по умолчанию, даже если его нет в таблице маршрутизации

Режим конфигурации:

Команда доступна в режиме настройки протокола OSPF:

```
router(config-router)#.
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **default-information originate** рассылает OSPF соседям маршрут по умолчанию. Маршрутизатор-инициатор рассылки маршрута по умолчанию становится шлюзом по умолчанию для OSPF соседей.

Пример:

```
router(config-router)# default-information originate
```

show ip ospf

Назначение:

Мониторинг работы протокола OSPF.

Синтаксис:

show ip ospf [border-routers | database | interface | neighbor | route]

Параметр	Описание
border-routers	Вывод информации о пограничных маршрутизаторах зоны
database	Вывод базы данных OSPF
interface	Вывод информации об интерфейсах, на которых запущен OSPF
neighbor	Вывод информации об OSPF соседях
route	Вывод маршрутов OSPF

Режим конфигурации:

Команда доступна в привилегированном режиме vty shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **show ip ospf** выводит информацию о текущих настройках и состоянии протокола OSPF. При вводе команды без параметров выводится общая информация о протоколе OSPF.

Пример:

```
router# sh ip ospf
OSPF Routing Process, Router ID: 2.2.2.2
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 200 millisec(s)
Minimum hold time between consecutive SPF's 1000 millisec(s)
Maximum hold time between consecutive SPF's 10000 millisec(s)
Hold time multiplier is currently 1
SPF algorithm last executed 3m20s ago
SPF timer is inactive
Refresh timer 10 secs
Number of external LSA 0. Checksum Sum 0x00000000
Number of opaque AS LSA 0. Checksum Sum 0x00000000
Number of areas attached to this router: 1

Area ID: 0.0.0.0 (Backbone)
Number of interfaces in this area: Total: 2, Active: 2
Number of fully adjacent neighbors in this area: 1
Area has no authentication
SPF algorithm executed 15 times
Number of LSA 4
Number of router LSA 2. Checksum Sum 0x0000f14d
Number of network LSA 2. Checksum Sum 0x0000e660
Number of summary LSA 0. Checksum Sum 0x00000000
Number of ASBR summary LSA 0. Checksum Sum 0x00000000
Number of NSSA LSA 0. Checksum Sum 0x00000000
Number of opaque link LSA 0. Checksum Sum 0x00000000
Number of opaque area LSA 0. Checksum Sum 0x00000000
```

log**Назначение:**

Установка параметров логирования.

Синтаксис:

log {file {filename} | syslog [logging level] | stdout [logging level]}

Параметр	Описание
file	Протоколирование событий будет осуществляться в файл
filename	Имя файла, в который будет осуществляться логирование. Для настройки логирования протокола OSPF нужно задать имя файла /var/log/ospfd.log
syslog	Включение удаленного логирования. Для работы данной команды необходимо также запустить процесс syslogd
logging level	Установка уровня логирования, по умолчанию logging level = debugging
stdout	Включение локального логирования

Режим конфигурации:

Команда доступна в режиме настройки vty shell:

```
router(config)#
```

Установка по умолчанию:

Отсутствует.

Описание:

Для локального логирования протокола OSPF необходимо настроить сохранение лога в файл /var/log/ospfd.log командой `log file /var/log/ospfd.log` и включить логирование командой `log stdout informational`. Уровень логирования `informational` устанавливает отправку информационных сообщений.

Для включения удаленного логирования OSPF необходимо ввести команду `log syslog`.

Пример:

```
router(config)# log file /var/log/ospfd.log
router(config)# log stdout informational
router(config)# log syslog
```

18 Пример настройки протокола OSPF

Параметры сети:

- R1 и R2 соединены по каналу G.703.
- Внутри корпоративной сети используется протокол OSPF с одной зоной 0.
- Сетевые параметры указаны на рисунке 3.

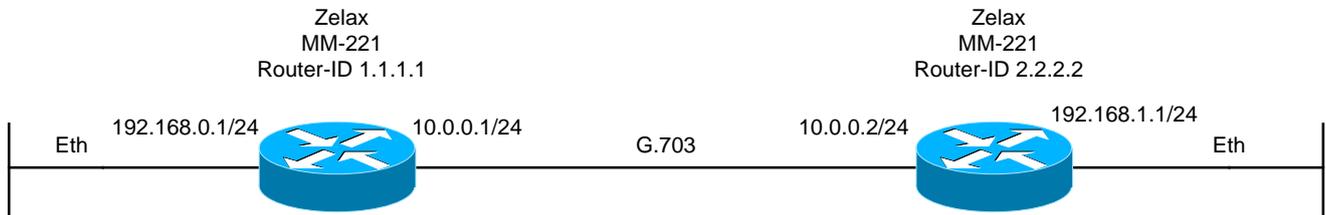


Рисунок 3. Схема локальной сети

Настройка маршрутизатора MM-221 Router-ID 1.1.1.1

netconfig.sh

```
#!/bin/sh
```

```
ospfd -P 0 -d -f /etc/config/ospfd.conf  
zebra -P 0 -d -f /etc/config/zebra.conf
```

```
## device name  
hostname router
```

```
## interface Ethernet 0 configuration  
ifconfig eth0 192.168.0.1 up mtu 1600
```

```
## interface HDLC 0 configuration  
sethdlc hdlc0 hdlc-eth  
ifconfig hdlc0 10.0.0.1 netmask 255.255.255.0 up
```

muxd.conf

```
interface HDLC 0 E1 2/0
```

router#vtysh

```
Hello, this is Quagga (version 0.99.15).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
router> enable  
router# configure terminal  
router(config)# router ospf  
router(config-router)# router-id 1.1.1.1  
router(config-router)# network 10.0.0.0/24 area 0  
router(config-router)# network 192.168.0.0/24 area 0  
router(config-router)# exit  
router(config)# exit  
router# copy running-config startup-config  
Building Configuration...  
Configuration saved to /etc/config/zebra.conf  
Configuration saved to /etc/config/ospfd.conf  
[OK]
```

Настройка маршрутизатора MM-221 Router-ID 2.2.2.2

netconfig.sh

```
#!/bin/sh
```

```
ospfd -P 0 -d -f /etc/config/ospfd.conf
zebra -P 0 -d -f /etc/config/zebra.conf

## device name
hostname router

## interface Ethernet 0 configuration
ifconfig eth0 192.168.1.1 up mtu 1600

## interface HDLC 0 configuration
sethdlc hdlc0 hdlc-eth
ifconfig hdlc0 10.0.0.2 netmask 255.255.255.0 up

muxd.conf
interface HDLC 0 E1 2/0

router#vtysh

Hello, this is Quagga (version 0.99.15).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

router> enable
router# configure terminal
router(config)# router ospf
router(config-router)# router-id 2.2.2.2
router(config-router)# network 10.0.0.0/24 area 0
router(config-router)# network 192.168.1.0/24 area 0
router(config-router)# exit
router(config)# exit
router# copy running-config startup-config
Building Configuration...
Configuration saved to /etc/config/zebra.conf
Configuration saved to /etc/config/ospfd.conf
[OK]
```

19 Настройка протокола RIP

Перед настройкой протокола динамической маршрутизации RIP необходимо запустить процессы `ripd` и `zebra` с помощью команд:

1. `ripd -P 0 -d -f /etc/config/ripd.conf`
2. `zebra -P 0 -d -f /etc/config/zebra.conf`

Данные команды необходимо записать в файл `netconfig.sh`. Настройки протокола RIP хранятся в файлах `ripd.conf` и `zebra.conf`.

Команды для настройки протокола RIP доступны в режиме `vtu shell` (Quagga).

router rip

Назначение:

Команда включает протокол RIP и переводит устройство в режим настройки протокола RIP.

Синтаксис:

router rip

Режим конфигурации:

Команда доступна в режиме настройки маршрутизатора:

```
router(config)#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **router rip** включает протокол RIP на маршрутизаторе и позволяет перейти в режим настройки протокола RIP. Для выхода из данного режима необходимо выполнить команду **exit**. Для выключения протокола RIP следует выполнить команду **no router rip**.

Пример:

```
router(config)# router rip
router(config-router)#
```

version

Назначение:

Команда определяет версию протокола RIP.

Синтаксис:

version {1 | 2}

Параметр	Описание
1	Устанавливает для протокола RIP версию 1
2	Устанавливает для протокола RIP версию 2

Режим конфигурации:

Команда доступна в режиме настройки протокола RIP:

```
router(config-router)#
```

Установка по умолчанию:

Принимаются пакеты RIPv1 и RIPv2, отправляются пакеты RIPv2

Описание:

Команда **version** определяет используемую версию протокола RIP. Рекомендуется отключить RIPv1 и указать использование RIPv2 путем использования команды `version 2`.

Пример:

```
router(config-router)# version 2
```

network**Назначение:**

Включение протокола RIP на интерфейсе.

Синтаксис:

network [ip-address]/[network-mask]

Параметр	Описание
ip-address	IP-адрес сети, в которой находится интерфейс
network-mask	Маска подсети, в которой находится интерфейс

Режим конфигурации:

Команда доступна в режиме настройки протокола RIP:

```
router(config-router)#.
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **network** включает протокол RIP на определенном сетевом интерфейсе. Для выключения протокола RIP на интерфейсе необходимо использовать команду **no network** [ip-address]/[network-mask]

Пример:

```
router(config-router)# network 192.168.0.0/24
```

passive-interface**Назначение:**

Запрещает рассылку RIP пакетов через указанный интерфейс.

Синтаксис:

passive-interface [интерфейс | **default**]

Параметр	Описание
Интерфейс	Название интерфейса
default	Запрет рассылки RIP пакетов через все интерфейсы, на которых включен протокол RIP

Режим конфигурации:

Команда доступна в режиме настройки протокола RIP:

```
router(config-router)#.
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **passive-interface** запрещает рассылку RIP пакетов через указанный интерфейс. Данная настройка применима для сокращения объема сетевого трафика, если через указанный интерфейс не могут быть достигнуты другие маршрутизаторы RIP. Для отмены режима **passive-interface** следует использовать команду **no passive-interface** [интерфейс | **default**].

Пример:

```
router(config-router)# passive-interface hdlc0
```

default-information originate

Назначение:

Рассылает в RIP информацию о маршруте по умолчанию.

Синтаксис:

default-information originate

Режим конфигурации:

Команда доступна в режиме настройки протокола RIP:

```
router(config-router)#.
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **default-information originate** рассылает другим маршрутизаторам RIP маршрут по умолчанию. Маршрутизатор-инициатор рассылки маршрута по умолчанию становится шлюзом по умолчанию для остальных маршрутизаторов RIP.

Пример:

```
router(config-router)# default-information originate
```

show ip rip

Назначение:

Мониторинг работы протокола RIP.

Синтаксис:

show ip rip [status]

Параметр	Описание
status	Вывод информации о текущих настройках и состоянии протокола RIP

Режим конфигурации:

Команда доступна в привилегированном режиме vty shell:

```
router#
```

Установка по умолчанию:

Отсутствует.

Описание:

Команда **show ip rip** выводит информацию о текущих настройках и состоянии протокола RIP. При вводе команды без параметра status выводится общая информация о протоколе RIP.

Пример:

```
router# show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 12 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
```

```
Redistributing:
Default version control: send version 2, receive version 2
  Interface      Send  Recv  Key-chain
  eth0           2    2
Routing for Networks:
  2.2.2.2/32
  192.168.0.0/24
Routing Information Sources:
  Gateway          BadPackets BadRoutes  Distance Last Update
Distance: (default is 120)
```

link-detect

Назначение:

Команда включает отслеживание состояния работы интерфейса на физическом уровне.

Синтаксис:

```
link-detect
```

Режим конфигурации:

Команда доступна в режиме настройки интерфейса:

```
router(config-if)#
```

Установка по умолчанию:

отсутствует

Описание:

Команда **link-detect** включает отслеживание состояния работы соответствующего интерфейса на физическом уровне. При включении данной опции в случае отказа интерфейса (например, при обрыве линии связи), устройство мгновенно перестроит таблицу маршрутизации, не дожидаясь момента рассылки следующего планового обновления..

Пример:

```
router(config)# router ospf
router(config-router)#
```

20 Настройка протокола VRRP

Перед настройкой протокола VRRP необходимо запустить процесс `keepalived` с помощью команды:
service start keepalived -P -l -f /etc/config/keepalived.conf

Настройки протокола VRRP хранятся в файле **keepalived.conf**.

Команды для настройки протокола VRRP необходимо записать в файл **keepalived.conf**. Редактирование файла **keepalived.conf** доступно в текстовом редакторе **nano**.

vrrp_instance

Назначение:

Команда для настройки экземпляра протокола VRRP.

Синтаксис:

vrrp_instance [имя] { команды-настройки-экземпляра-VRRP }

Параметр	Описание
имя	Название экземпляра протокола VRRP. Каждый экземпляр протокола VRRP должен иметь уникальное название.

Режим конфигурации:

Команда вводится в файле `keepalived.conf`.

Установка по умолчанию:

отсутствует

Описание:

Команда **vrrp_instance** создает экземпляр протокола VRRP на маршрутизаторе. Экземпляр протокола VRRP представляется собой группу логически объединенных виртуальных ip адресов. Маршрутизаторы поддерживают неограниченное количество экземпляров протокола VRRP.

Команды для настройки виртуального объединения маршрутизаторов необходимо вводить после команды **vrrp_instance** в фигурных скобках.

Пример:

```
vrrp_instance VI_GATEWAY {
    state MASTER
    interface eth0
    virtual_router_id 52
    priority 150
    advert_int 1
    virtual_ipaddress {
        10.20.40.1/24
    }
}
```

state

Назначение:

Команда для настройки роли маршрутизатора в протоколе VRRP.

Синтаксис:**state** [MASTER | BACKUP]

Параметр	Описание
MASTER	Состояние Master (основной маршрутизатор)
BACKUP	Состояние Backup (резервный маршрутизатор)

Режим конфигурации:

Команда вводится в файле `keepalive.conf` после команды `vrrp_instance` в фигурных скобках.

Установка по умолчанию:

отсутствует

Описание:

Команда в явном виде задает состояние маршрутизатора. Возможные состояния: `master` – основной маршрутизатор, `backup` – резервный маршрутизатор. В состоянии `master` маршрутизатор обрабатывает пакеты, отправленные на виртуальный IP-адрес, и отвечает на ARP запросы. В состоянии `backup` маршрутизатор не обрабатывает пакеты, отправленные на виртуальный IP-адрес, и ARP запросы. Основная задача маршрутизатора в состоянии `backup` – мониторинг состояния маршрутизатора `master`.

Основным при выборе состояния маршрутизатора является приоритет. Маршрутизатор с более высоким приоритетом принимает роль `master`, даже если в команде `state` указана роль `backup`.

Пример:

```
vrrp_instance VI_GATEWAY {
    state MASTER
}
```

priority**Назначение:**

Команда для настройки приоритета маршрутизатора в протоколе VRRP.

Синтаксис:**priority** [приоритет]

Параметр	Описание
приоритет	Приоритет маршрутизатора. Числовое значение. Допустимый диапазон: 0 - 255

Режим конфигурации:

Команда вводится в файле `keepalive.conf` после команды `vrrp_instance` в фигурных скобках.

Установка по умолчанию:

отсутствует

Описание:

Приоритет маршрутизатора используется при определении состояния маршрутизатора и является основным критерием выбора. Маршрутизатор с более высоким приоритетом принимает роль `master`, даже если в команде `state` указана роль `backup`.

Пример:

```
vrrp_instance VI_GATEWAY {
    priority 150
}
```

interface

Назначение:

Команда для настройки интерфейса для обмена VRRP сообщениями.

Синтаксис:

interface [название-интерфейса]

Параметр	Описание
название-интерфейса	Интерфейс маршрутизатора, через который будут отправляться служебные сообщения VRRP. За этим интерфейсом должен находиться маршрутизатор с запущенным протоколом VRRP.

Режим конфигурации:

Команда вводится в файле `keepalive.conf` после команды `vrrp_instance` в фигурных скобках.

Установка по умолчанию:

отсутствует

Описание:

Команда **interface** определяет интерфейс маршрутизатора, через который отправляются сообщения VRRP. За этим интерфейсом должен находиться маршрутизатор с запущенным протоколом VRRP. Если для установления отношений VRRP используется несколько интерфейсов, необходимо ввести команду несколько раз с указанием всех используемых интерфейсов.

Пример:

```
vrrp_instance VI_GATEWAY {  
    interface eth0  
    interface hdlc0  
}
```

virtual_router_id

Назначение:

Команда для настройки идентификатора маршрутизатора в виртуальной группе VRRP.

Синтаксис:

virtual_router_id [идентификатор]

Параметр	Описание
идентификатор	Идентификатор маршрутизатора в виртуальной группе. Числовое значение.

Режим конфигурации:

Команда вводится в файле `keepalive.conf` после команды `vrrp_instance` в фигурных скобках.

Установка по умолчанию:

отсутствует

Описание:

Команда **virtual_router_id** определяет идентификатор маршрутизатора в группе VRRP. Каждый маршрутизатор должен иметь уникальный идентификатор в рамках одного экземпляра протокола VRRP.

Пример:

```
vrrp_instance VI_GATEWAY {  
    virtual_router_id 52  
}
```

advert_int

Назначение:

Команда для настройки идентификатора маршрутизатора в виртуальной группе VRRP.

Синтаксис:

advert_int [время]

Параметр	Описание
время	Временной интервал рассылки служебных сообщений VRRP. Числовое значение, указывается в секундах.

Режим конфигурации:

Команда вводится в файле `keepalive.conf` после команды `vrrp_instance` в фигурных скобках.

Установка по умолчанию:

отсутствует

Описание:

Команда **advert_int** определяет интервал рассылки служебных сообщений VRRP. Данные сообщения используются для обмена информацией о параметрах протокола VRRP и для обнаружения изменений топологии внутри виртуальной группы маршрутизаторов. Чем меньше интервал между сообщениями, тем быстрее маршрутизаторы реагируют на изменение топологии, но тем больший объем служебного трафика генерируется в процессе работы протокола VRRP. Данный параметр должен совпадать на всех маршрутизаторах внутри виртуальной группы.

Пример:

```
vrrp_instance VI_GATEWAY {  
    advert_int 1  
}
```

virtual_ipaddress

Назначение:

Команда для настройки виртуального IP адреса.

Синтаксис:

virtual_ipaddress { [ip-address_1] [ip-address_2] ... }

Режим конфигурации:

Параметр	Описание
ip_address	Виртуальный IP адрес группы VRRP.

Команда вводится в файле `keepalive.conf` после команды `vrrp_instance` в фигурных скобках.

Установка по умолчанию:

отсутствует

Описание:

Команда **virtual_ipaddress** определяет виртуальный IP адрес для группы маршрутизаторов, объединенных по протоколу VRRP. В зависимости от задачи может быть указано несколько виртуальных IP адресов. Виртуальный IP адрес используется другими устройствами сети для обращения к маршрутизаторам, объединенным в группу VRRP.

Пример:

```
vrrp_instance VI_GATEWAY {  
    virtual_ipaddress {  
        10.20.40.1  
    }  
}
```

21 Пример настройки протокола VRRP

R1 и R2 объединены в группу VRRP с виртуальным ip-адресом 172.16.1.153. R1 является основным маршрутизатором, R2 – резервным. К интерфейсу Ethernet 0 обоих маршрутизаторов подключен коммутатор. Схема подключения приведена на рисунке 4.

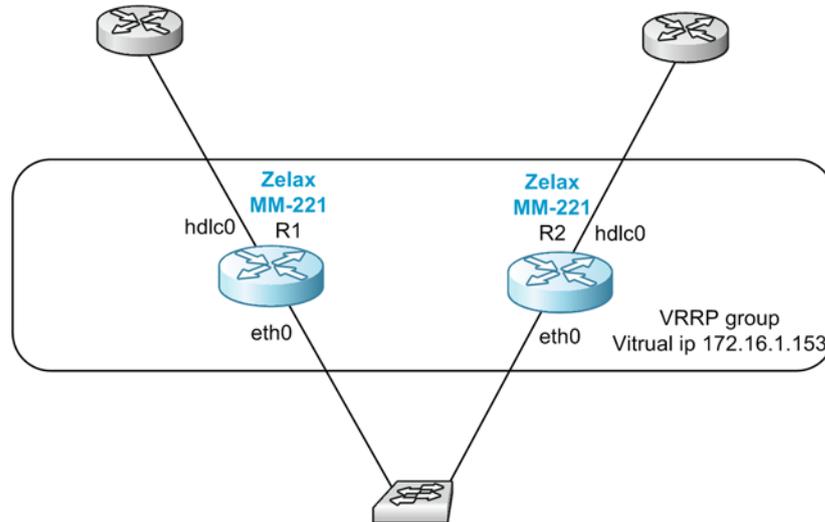


Рисунок 4. Схема подключения

Настройка маршрутизатора MM-221 R1

netconfig.sh

```
service start keepalived -P -l -f /etc/config/keepalived.conf
ifconfig eth0 172.16.1.156 netmask 255.255.255.0 up
```

keepalived.conf

```
vrpp_instance VI_1 {
    state master
    interface eth0
    track_interface {
        hdlc0
    }
    virtual_router_id 1
    priority 150
    advert_int 1
    virtual_ipaddress {
        172.16.1.153/24
    }
}
```

Настройка маршрутизатора MM-221 R2

netconfig.sh

```
service start keepalived -P -l -f /etc/config/keepalived.conf
ifconfig eth0 172.16.1.157 netmask 255.255.255.0 up
```

keepalived.conf

```
vrpp_instance VI_1 {
    state slave
    interface eth0
    track_interface {
        hdlc0
    }
}
```

```
virtual_router_id 1
priority 50
advert_int 1
virtual_ipaddress {
  172.16.1.153/24
}
}
```